Service Catalog Administration Guide

PDF generated using the open source mwlib toolkit. See http://code.pediapress.com/ for more information.
PDF generated at: Sat, 15 Mar 2014 17:44:55 PST

Using the Service Catalog

Using the Service Catalog

Overview

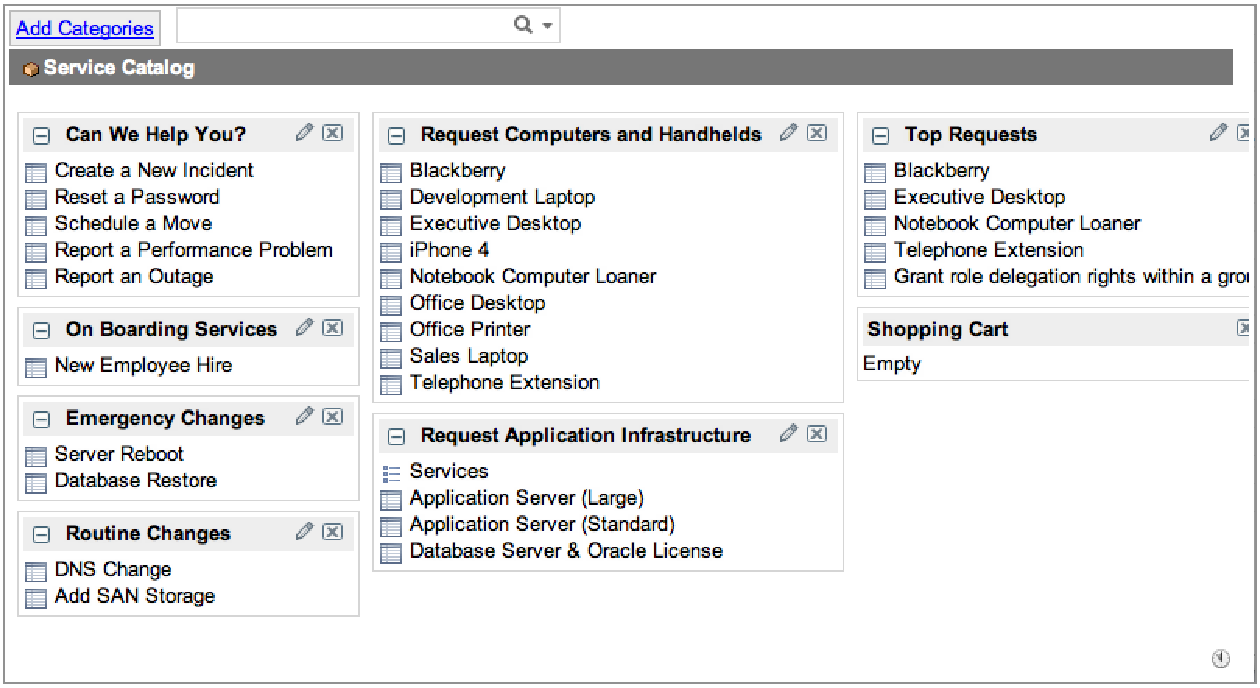
The service catalog enables customers to view and order items from departments (such as IT) within your organization. These items can include goods and services (*catalog items*), or information (*content items*).

Anything that can be ordered individually can be ordered as a catalog item. Anything that only exists as a part of a larger whole cannot be a catalog item. For example, a laptop can be a catalog item, but a high-resolution display for a laptop cannot.

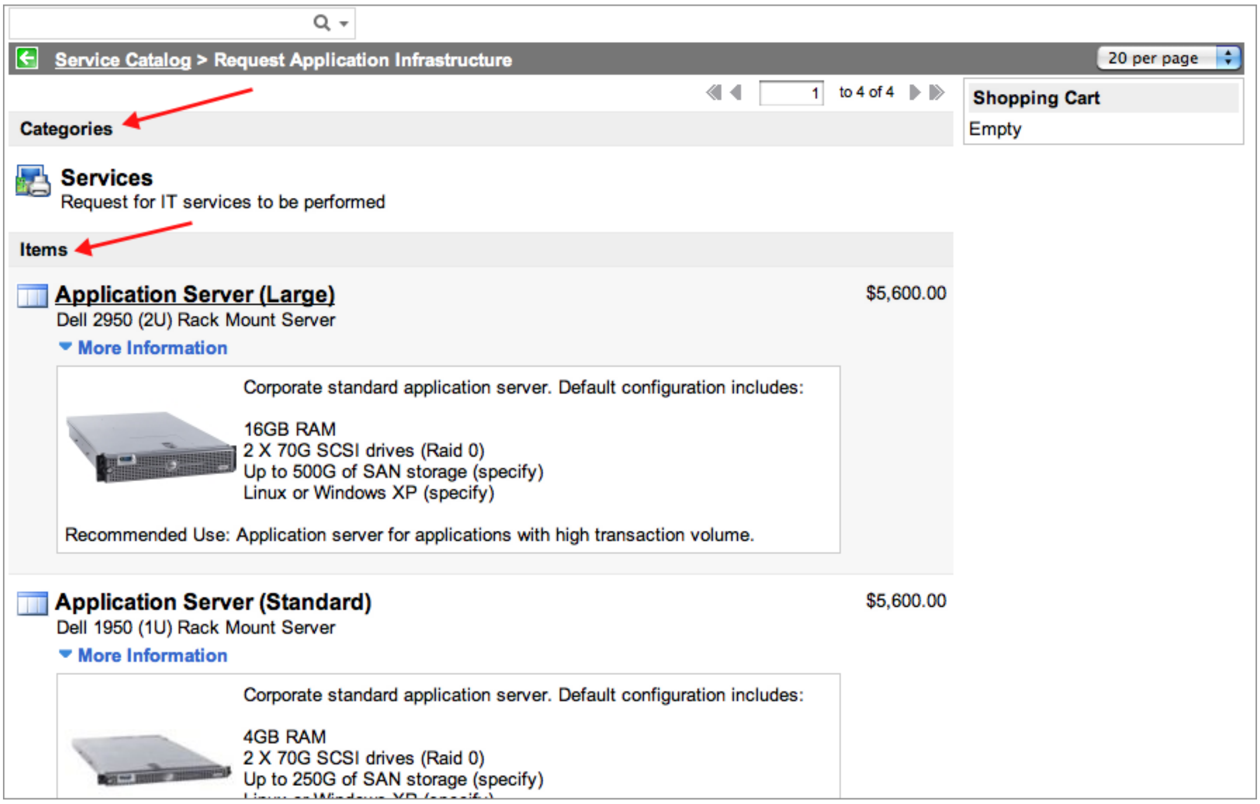
After placing an order, the customer can track its progress. Predefined groups follow a series of tasks to deliver the item, based on a fulfillment process, including any approvals required.

Viewing and Navigating the Service Catalog

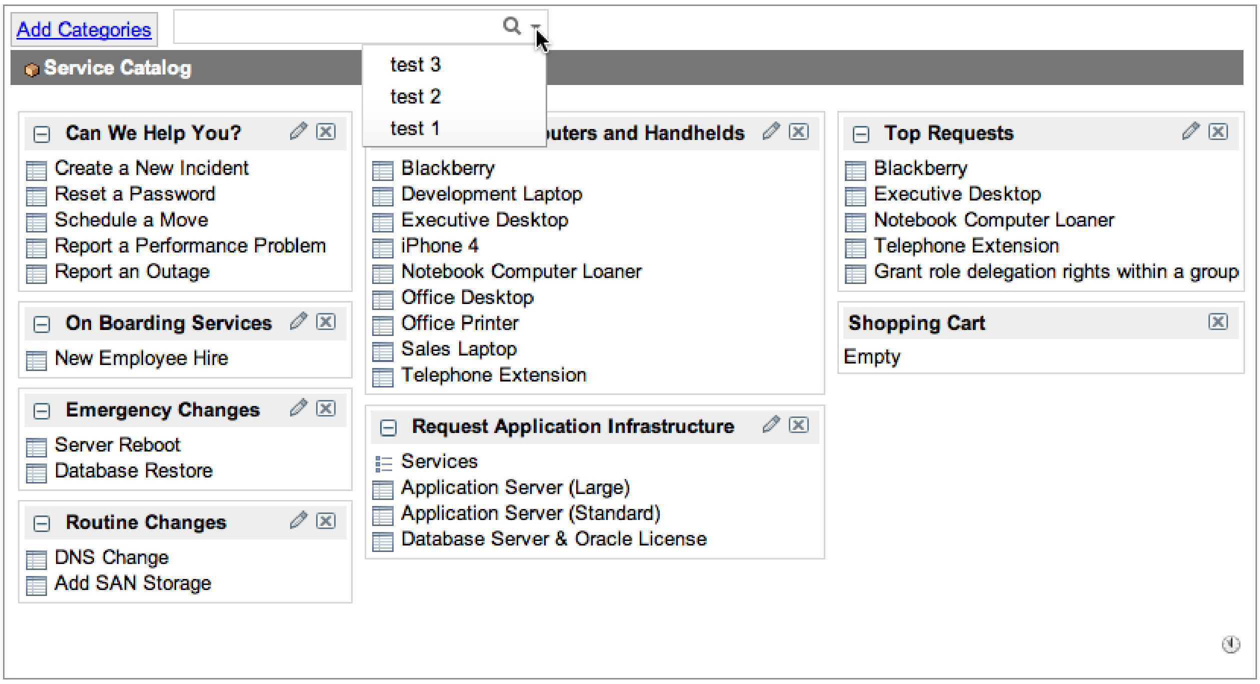
The service catalog homepage lists the goods and services available to order. Catalog items are grouped into categories, which may also contain one or more subcategories. By default, the first ten items in a category appear under the category name on the service catalog homepage.



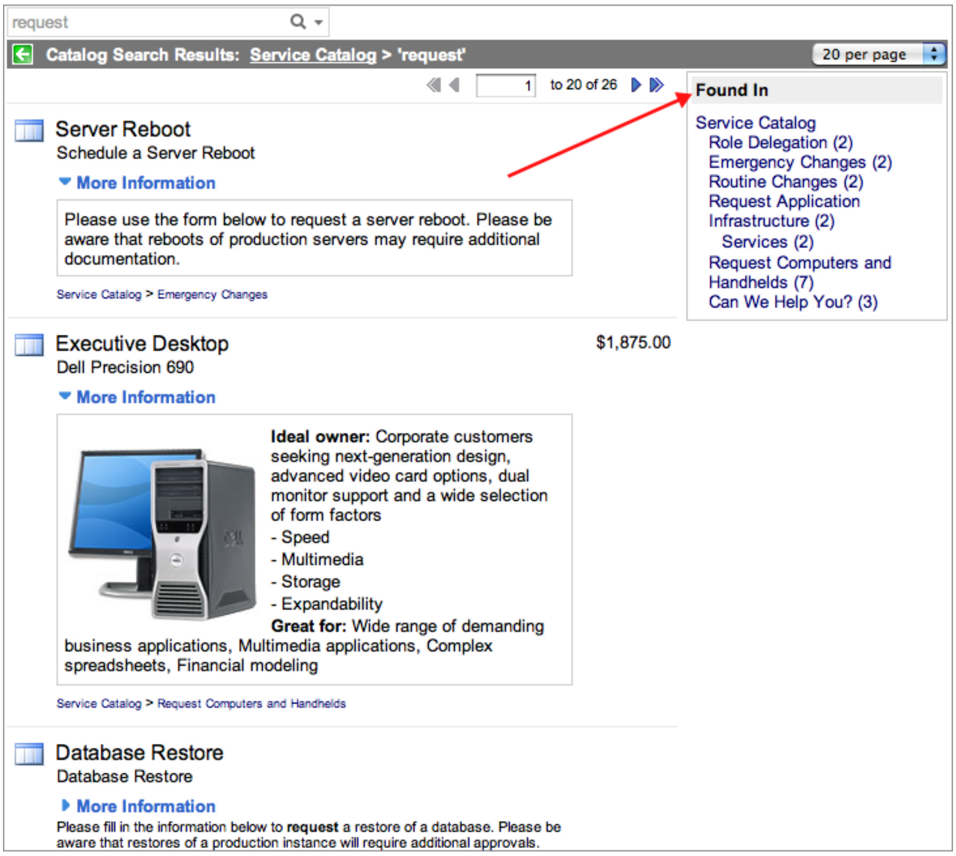
Click a category name to see a list of all subcategories and items in the category.



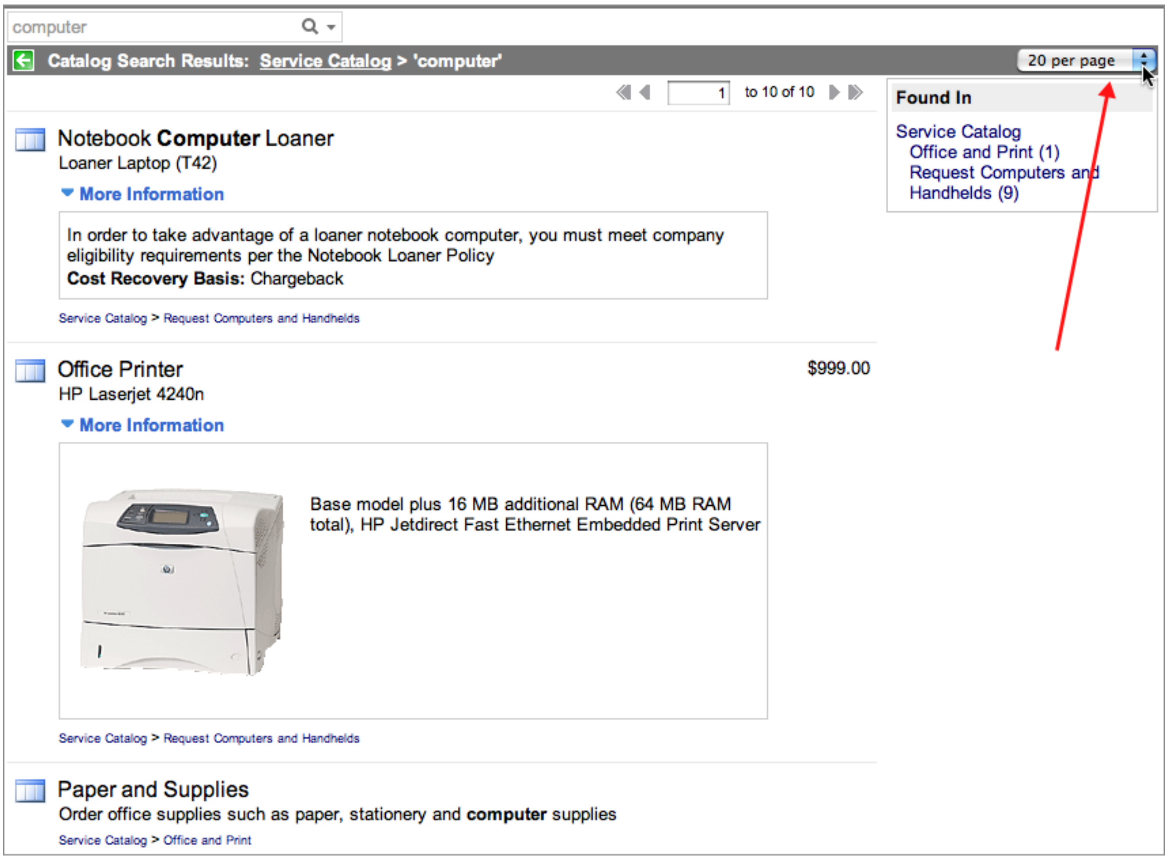
To find a specific item, use the **Search** field while browsing the service catalog. Click the down arrow to see a list of previous searches. Catalog search is available on catalog and category pages.



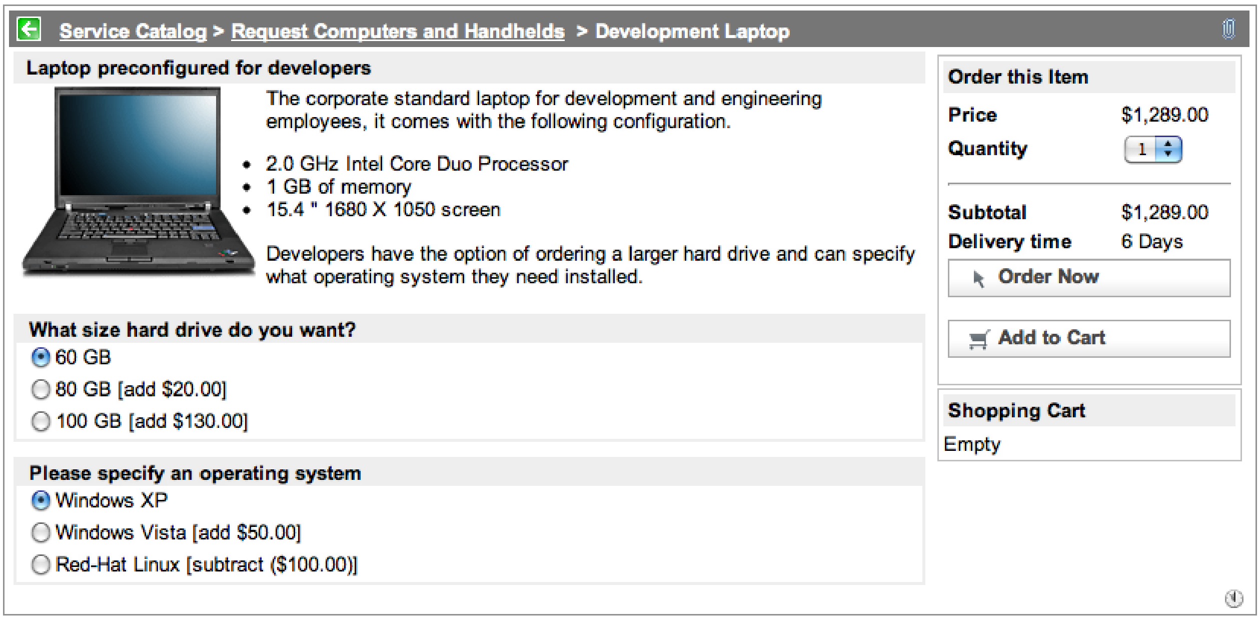
Search results list items in active categories along with their short description. Use the breadcrumbs in the header bar or below the short description to quickly navigate to a different page. The **Found In** list shows the categories related to the search results in hierarchical format.



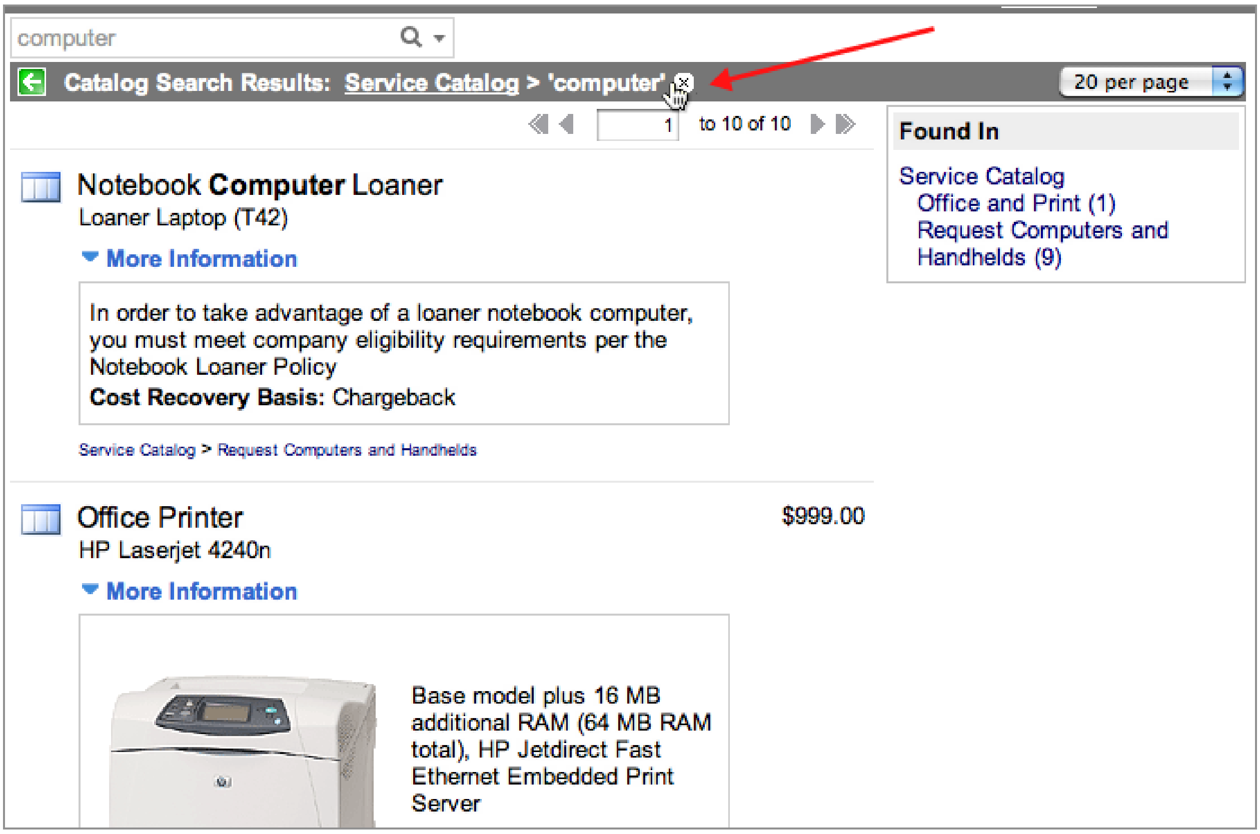
Control the number of results shown on one page with the search results selection list.



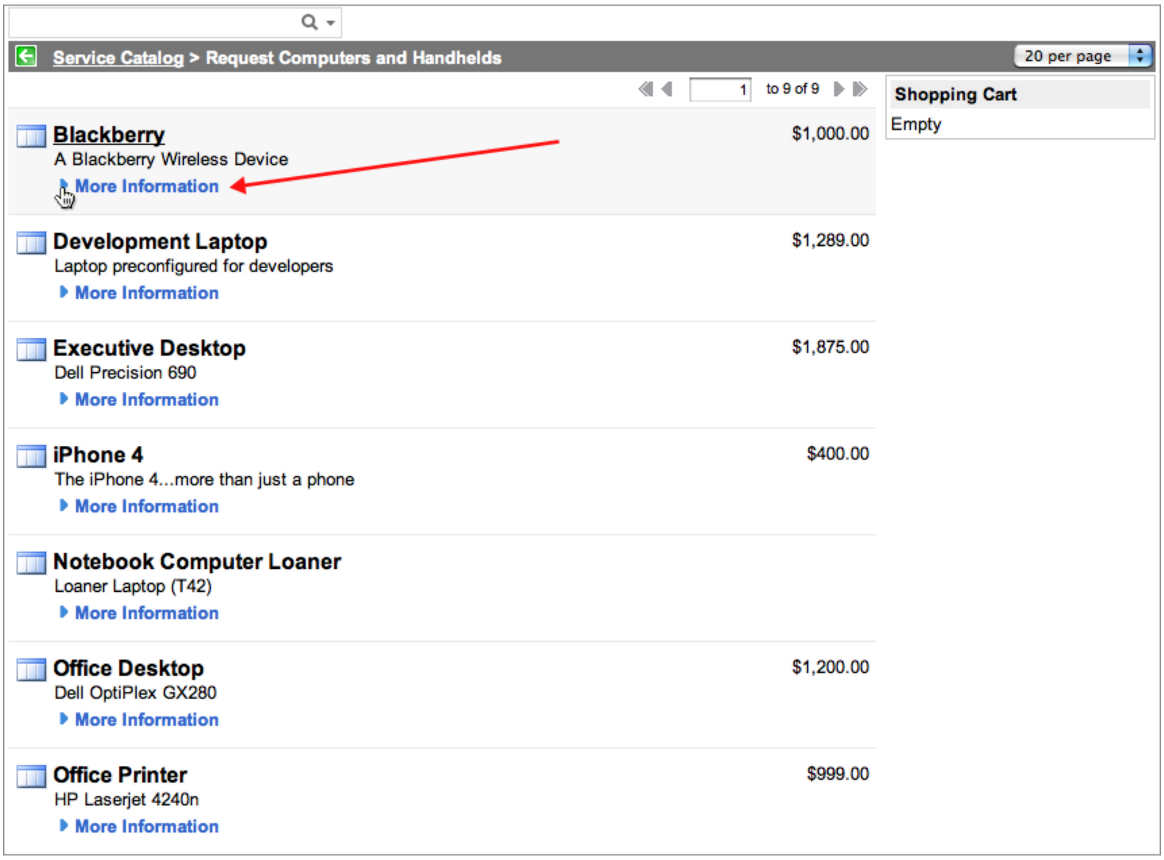
In the catalog listing, click a link in the breadcrumbs at the top of the screen to quickly navigate back one or more levels in the hierarchy.



To remove the search term from the breadcrumbs, point to then click the *x* next to the term.



If a search returns multiple results, obtain details about a specific catalog item by clicking **More Information**.

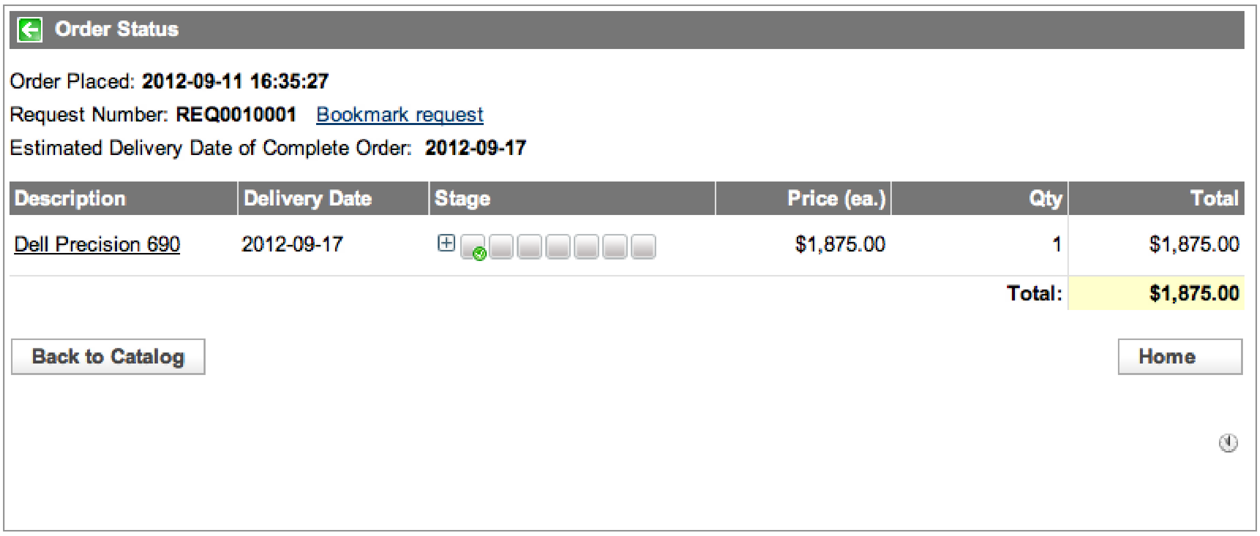


Requesting a Service Catalog Item

To place a request, navigate to **Self-Service > Service Catalog** and select the item to order.

See Placing a Request for more information about making an order.

When an item is ordered, ServiceNow generates a request to track the order and displays a summary that includes the order status in the **Stage** column:



Each individual catalog item in a request creates a discrete request item. For example, a request for 2 PCs, 1 chair, and 1 desk would produce four request items on a single request.

Placing a Request

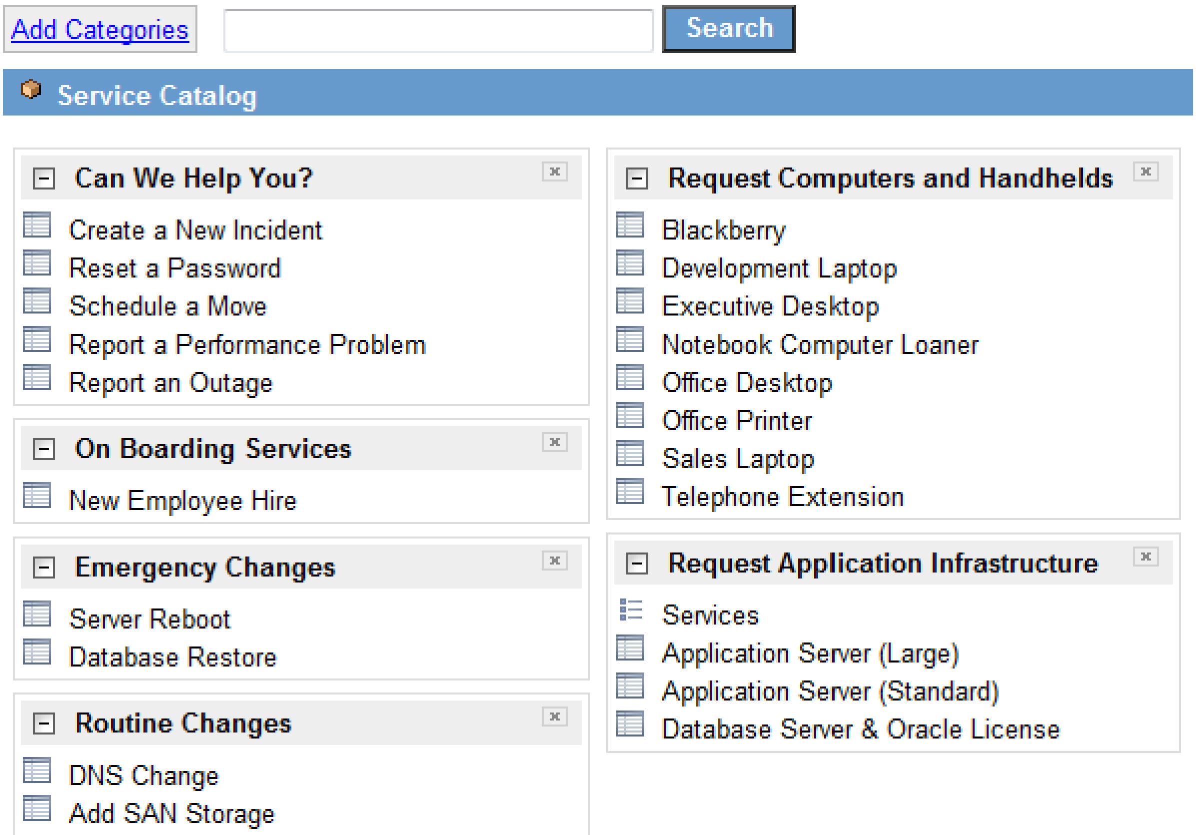
Overview

The ServiceNow service catalog offers an intuitive, web-based interface for ordering predefined goods and services.

Placing a Request

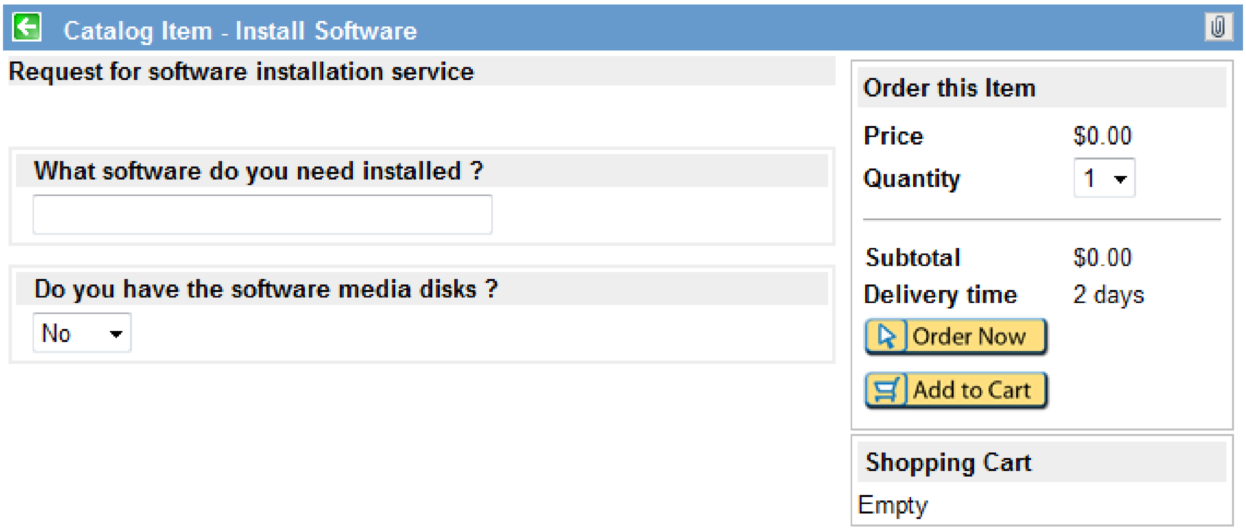
1. Navigate to **Self-Service > Service Catalog**.

   The default catalog view organizes items in categories and subcategories.



2. Select an item to order.

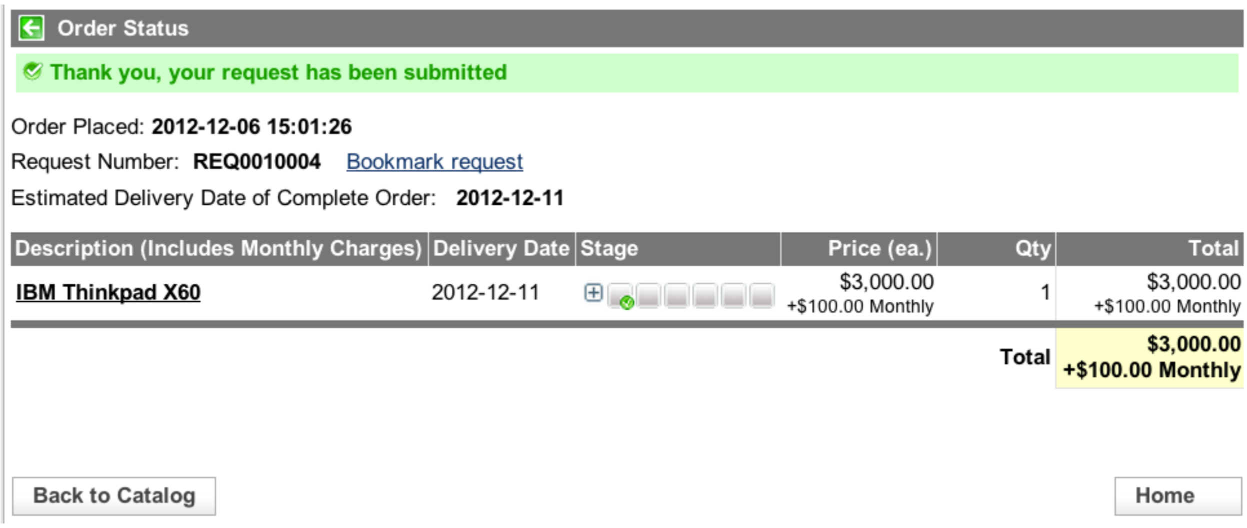
   The order screen requests any additional information that's needed, such as the requester's name or location and the date the item is needed. It also displays the item's delivery time after approval.



Some items are *order guides*, which combine related items for easier ordering. For example, an order guide called *New Employee Hire* presents a list of items that new employees typically receive (desk, phone, computer, email account). After you select the items you want to order, the order guide requests the information required and shows a separate time to delivery for each item ordered.

3. Enter complete and accurate information to expedite your order.
4. Order any other items required.
5. Click **Order Now** when the order is complete.

   ServiceNow displays a confirmation screen for your order:



   You can view the order's progress in the **Stage** column.

   It may be useful to note the order number for future reference.

ServiceNow automatically notifies the approvers if approval is required and creates work orders to fulfill the order according to your organization's process. For example, if you order a new notebook computer, the purchasing group might receive a work order to order the computer, and the desktop services group might receive a work order to configure the new computer and deliver it to your office after it arrives.

Viewing Request Status

1. Navigate to **Self-Service > Requested Items**.
2. Click your order number to view details.
3. View the order's progress in the **Stage** field.

Modifying a Request

After creating a service catalog request, you can delete items from it. Only an administrator or a user with the catalog_admin role can add items to a request.

Modifying a request does not resubmit it to the approval process.

Adding New Items

Only an administrator or a user with the catalog_admin role can add a new item to a submitted request.

1. Navigate to **Service Catalog > Open Records > Requests** and open an existing request.
2. Click **Add New Item**.
3. In the dialog box that appears, choose an item and a quantity to add.
4. Click **OK**.

Deleting Items

All users can delete items from their own requests any time prior to delivery.

1. Navigate to **Service Catalog > Open Records > Requests** and open an existing request.
2. In the **Requested Items** related list, select the check box beside the item to delete.
3. In the **Actions** choice list, select **Delete**.

   ServiceNow recalculates the price and delivery date for the order.

Service Catalog Management

Service Catalog Management

Overview

A service catalog provides a customer-facing view of available service and product offerings provided by departments within the organization. This allows organizations to promote these offerings in a structured and easily navigable way, and encourages customers to access the catalog to serve their own needs.

A well-defined and managed service catalog provides:

- A self-service opportunity for customers.
- A single portal to present all service and product offerings.
- A standardized approach to request fulfillment.
- Management of customer expectations.

The aim of service catalog management is to ensure the accuracy and availability of the items provided within the catalog.

The Service Catalog application allows you to define and manage all aspects of your service catalog, including the ability to define the available catalog items, define specific request fulfillment workflows, and create a customizable catalog portal page.

Setting Up a Service Catalog

Administrators and catalog administrators, users with the catalog_admin role, can use the Service Catalog application to define service catalog content and layout.

A typical service catalog setup process involves the following steps:

1. Assign roles to users who are working with the service catalog.
2. Customize the service catalog homepage to meet your requirements.
3. Define content to provide in the service catalog:
    - Catalog items: the goods and services available within the catalog.
    - Categories: the groups of items displayed on the catalog home page.
    - Variables: the options available for tailoring a catalog item to meet specific needs.
4. Define request fulfillment processes, the procedures your organization uses to deliver ordered catalog items.

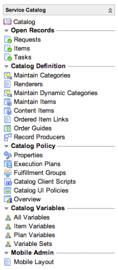
Administrators and catalog administrators can extend the service catalog with specialized catalog items, configuration options, and scripting functions, to provide more powerful features. For example, administrators can customize the checkout process used when ordering catalog items.

Roles

Role Title [Name]	Description
Administrator [admin]	Can manage all aspects of the Service Catalog application, including scripting.
Catalog administrator [catalog admin]	Can manage most aspects of the Service Catalog application, including categories and items. Cannot use scripting functions, such as creating UI macros or business rules.

Menus and Modules

The Service Catalog application contains the following modules.



- **Catalog:** Access the service catalog portal page to view and request items.
- **Open Records:** View open **Requests**, **Items**, or **Tasks**.
- **Catalog Definition**
  - **Maintain Categories:** Create and edit categories and subcategories within a catalog.
  - **Renderers:** Define how categories are displayed.
  - **Maintain Dynamic Categories:** Create and edit dynamic categories to display commonly-requested items.
  - **Maintain Items:** Create and edit catalog items, the actual goods or services available to order from a catalog.
  - **Content Items:** Create and edit information-based catalog items such as knowledge articles.
  - **Ordered Item Links:** Create and edit link on the ordered item screen, such as a link to a standard terms and conditions page.
  - **Order Guides:** Create and edit standard groups of related items, allowing end users to easily order these items in one request. For example, define a **New Hire Equipment** order guide.
  - **Record Producers:** Enable records to be created directly from the service catalog. For example, allow customers to raise incidents from the service catalog.
- **Catalog Policy**
  - **Properties:** Define general behavior and usage for a service catalog. For example, configure search behavior and results.
  - **Execution Plans:** Define how a catalog item is procured, configured, and installed (a less-used alternative to workflows).
  - **Fulfillment Groups:** Define groups to perform the work involved in request fulfillment.
  - **Catalog Client Scripts:** Define dynamic effects and validation to items or variable sets.
  - **Catalog UI Policies:** Define the appearance and behavior of fields. For example, make it mandatory for anyone with the title **IT Technician** to add a mobile phone number when ordering business cards.
  - **Overview:** View a homepage showing reports about open requests for catalog items.
- **Catalog Variables**
  - **All Variables:** View and edit all service catalog variables, including those associated to catalog items.
  - **Item Variables:** View and edit only the variables that are associated to catalog items.
  - **Plan Variables:** View and edit variables used with execution plans.
  - **Variable Sets:** View and edit groups of variables that can be shared as sets between catalog items.
- **Mobile Admin**
  - **Mobile Layout:** Configure categories displayed in the mobile interface.

Enhancements

Dublin

- Administrators can create service catalog record producers directly from the table record.
- Users can access the service catalog from mobile devices.
- On catalog item records, the **Model** field is now visible by default. The field is automatically populated for items created by publishing models.

Calgary

- Searching and navigation enhancements.
- New reference qualifier variables.
- Support for catalog UI policy and catalog client scripts in service catalog wizard screens.
- Catalog variable data lookup support.
- Support for HTML variables.
- Setting recurring prices on catalog items.
- Use of renderers to customize category look-and-feel.
- Coordination with Cloud Provisioning.

In addition, new catalog properties are available, to configure:

- Behavior for classes of catalog items. For example, use `glide.sc.item.cannot_add_to_request` to specify a list of class names for catalog items that cannot be added to an existing request.
- Display and view of categories in the service catalog. For example, use `glide.sc.use_sub_cat_section` to display subcategories in a panel.
- Search behavior. For example, use `glide.sc.search.disabled_cats` to search inactive categories.
- Catalog item displays. For example, use `glide.sc.max_items` to configure the number of catalog items or categories to preview in a section.
- CMS behavior. For example, use `glide.sc.search.cms_page` to specify a specific service catalog search page for CMS.

Setting Up the Service Catalog

Managing the Service Catalog Homepage

Overview

The service catalog homepage provides the primary front end for ordering catalog items.

Administrators and users with the catalog_admin role can design the service catalog homepage by:

- Customizing the service catalog homepage.
- Adding, removing, and arranging categories.
- Enabling content types.
- Using service catalog properties to provide additional control over behavior and appearance.
- Using service catalog renderers to define the appearance of service catalog categories.

Note: *End users can access the service catalog via the customizable user homepage, or with content management pages that use content blocks for categories.*

Enabling Content Types for the Service Catalog

To select content types, also called (widgets), to use for the service catalog:

1. Navigate to **Service Catalog > Catalog Policy > Properties**.
2. In the property **List of content types (comma-separated) to allow on the catalog homepage**, enter the content types to use in the service catalog.

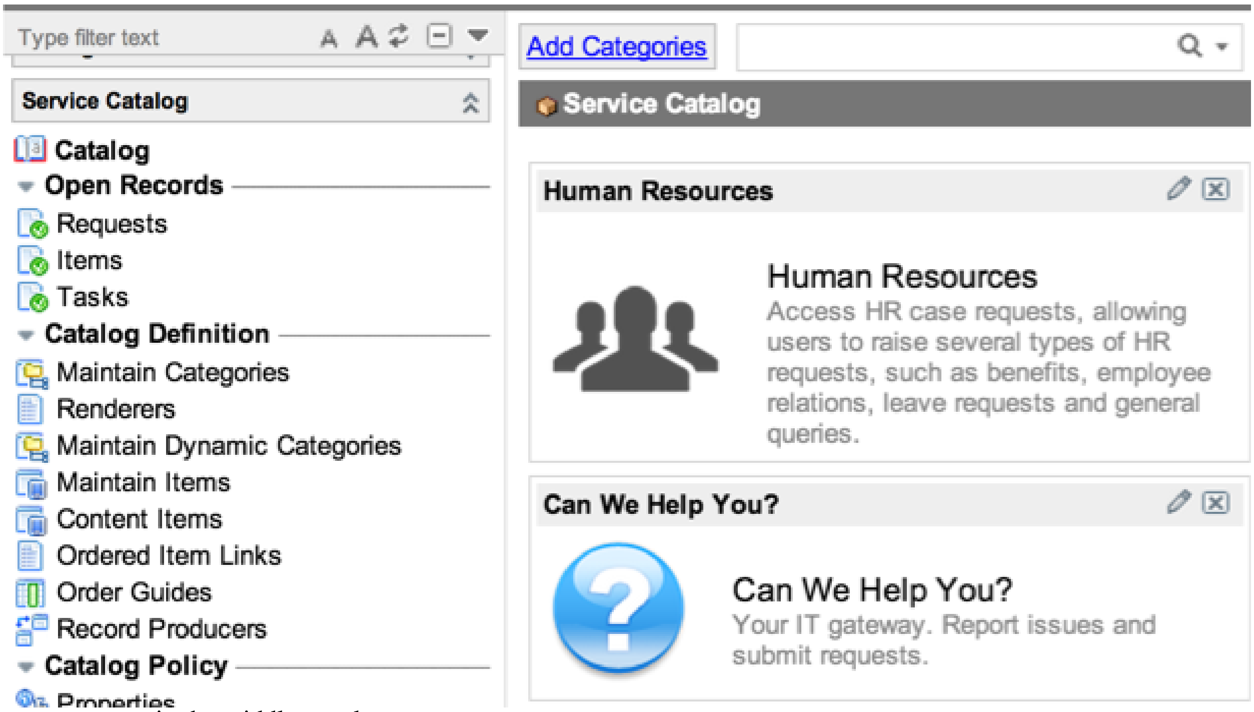
    To see a list of content types available, navigate to **System UI > Widgets**.

    The default value is **Service Catalog**, which includes only service catalog categories. For example, to include service catalog categories and gauges, set the property to: **Service Catalog, Gauges**.

Customizing the Service Catalog Homepage

Administrators and catalog administrators can customize the service catalog homepage in a similar way to customizing any homepage.

1. Navigate to **Service Catalog > Catalog**.
2. Click **Add Categories** at the top of the page.



3. Select a category in the middle panel.
4. Click **Add here** in the location where the category should appear on the homepage.
5. [Optional] Repeat steps 2-4 to add more categories.
6. Close the pop-up window.
7. [Optional] To change a category's location, drag it to the appropriate place.
8. [Optional] To remove a category, click the X on the top right of the category header.
9. [Optional] Add a shopping cart on the page. For information, see Creating a Custom Shopping Cart.

Limiting Search Access

The service catalog homepage provides a search bar to help locate items not displayed on the homepage. Customers can search only for items they have access to.

To limit access to the search function by role:

1. Navigate to **Service Catalog > Catalog Policy > Properties**.
2. In the property **List of roles (comma-separated) that can search the service catalog**, select the role, as appropriate.

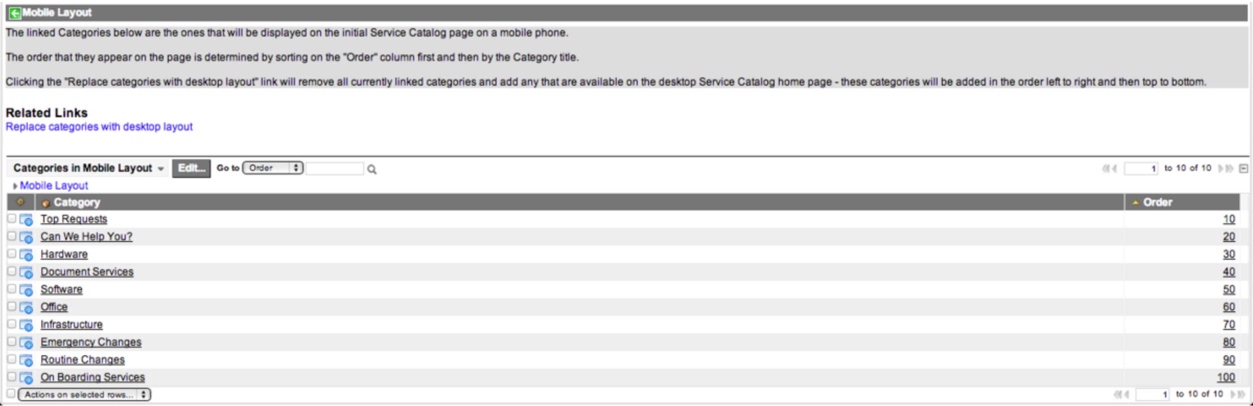
   To remove the search function for all users, select **None**.

Defining Mobile Layout for Categories

By default, categories are displayed on mobile devices in the same order as on desktop devices.

Navigate to **Service Catalog > Mobile Layout** to configure:

- Which categories appear on mobile devices.
- The order of categories that appear on mobile devices.



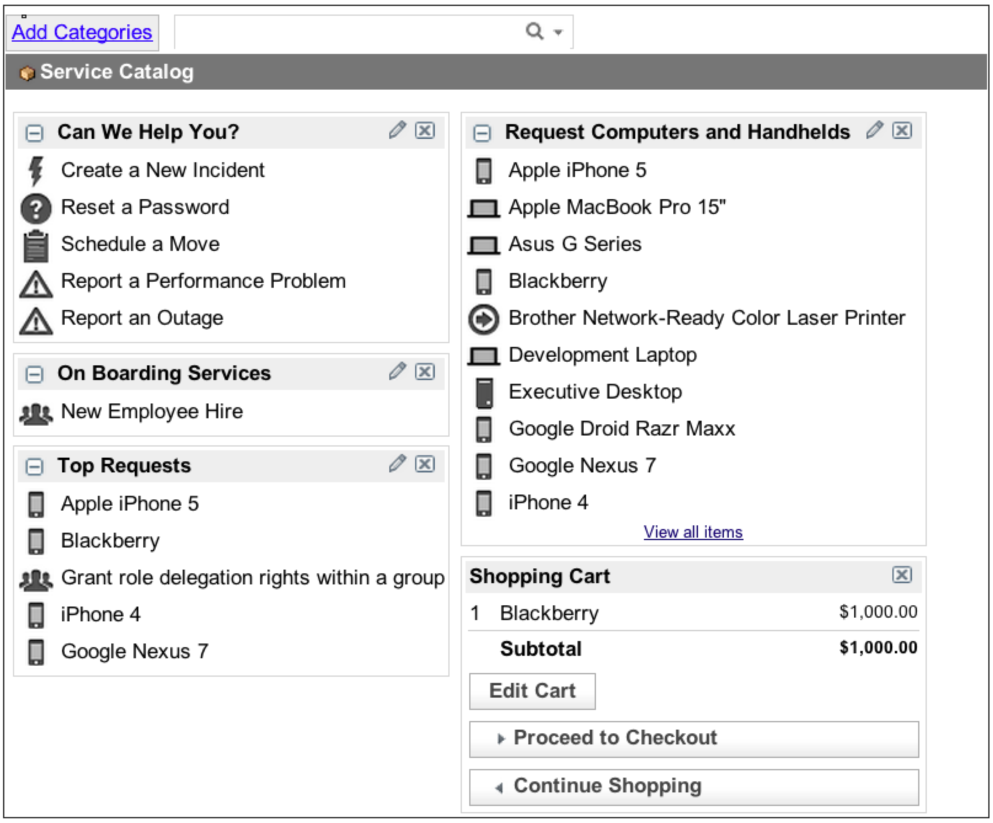
Select the **Replace categories with desktop layout** link to return the display to the default setting, showing the same order and values as for the desktop.

Administrators can configure the category display on mobile devices starting with the Dublin release.

Service Catalog Categories

Overview

Categories are used to organize service catalog items into logical groups. Categories can have a parent-child relationship, for example, *IT* and *Laptops*. A child category is a *subcategory* of its parent category. Each catalog item, order guide, record producer, content item, and subcategory appears as a single item within the category.



Administrators and catalog administrators can create and configure categories, defining their characteristics and adding content such as catalog items to them.

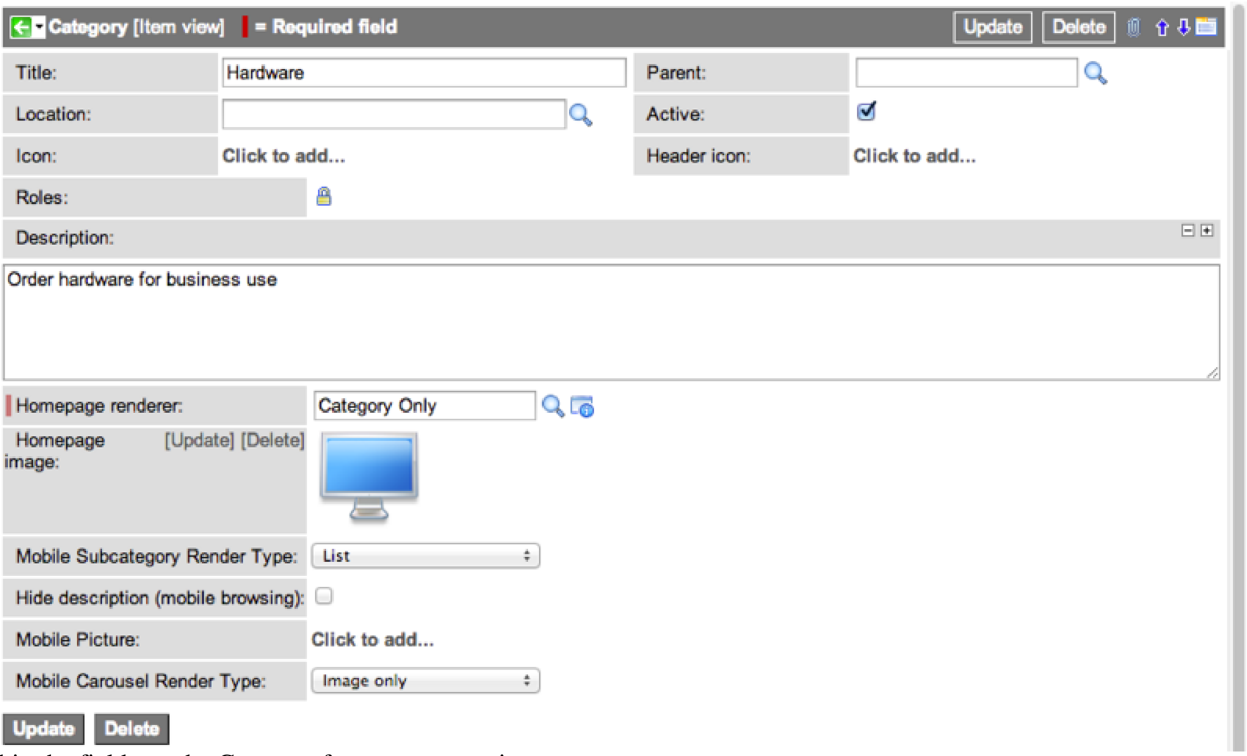
Note: *If there are no active items in a category's hierarchy, that category does not appear in (and cannot be added to) the catalog. Users with the admin or catalog_admin roles can see all categories, regardless of the number of active items. Configure the* `glide.sc.category.canview.override` *property to change this behavior.*

Creating and Configuring Categories

To create and configure a category:

1. Navigate to **Service Catalog > Catalog Definition > Maintain Categories**.

   A list of existing categories appears. If the **Parent** field is blank, the category does not have a parent category.
2. Click **New** or select the category to edit.



3. Fill in the fields on the Category form, as appropriate.

Field	Description
Title	The descriptive name for the category.
Location	Any location information relevant for the category.
Icon	The small icon displayed beside the category name, when the category is listed as a subcategory.
Roles	Roles permitted to order items in this category. Leave the field blank, or use the role public, to allow access for all users. Category-level restrictions may be overridden by restrictions on the individual items within that category.
Parent	Any parent category for which this category is a subcategory.
Active	A check box for indicating whether the category is available to add to the service catalog homepage.
Header icon	The icon displayed beside the category header, when the category is a top-level category.
Description	Information about the category. The description appears when the category is collapsed on the service catalog homepage. The description also appears in the **Related Categories** list when a user clicks the parent category title.
Homepage renderer	A Renderer to use for displaying this category on the service catalog homepage.
Homepage image	A larger image to display with the category on the service catalog homepage. This is visible only if the relevant renderer defines it to be used (the default renderer does not display this).
Mobile Subcategory Render Type	The rendering option to control how subcategories are displayed. Select **List** for a single-column list, or **Cards** for a multi-column display. Available starting with the Dublin release.
Hide description (mobile browsing)	Option for displaying the description at the top of the list on mobile devices. Available starting with the Dublin release.

Mobile Picture	A picture to use on mobile devices for this category. If a mobile image is not selected, the **Homepage image** appears if the **Mobile Subcategory Render Type** is set to **Cards**. Available starting with the Dublin release.

Configuring Dynamic Categories

Dynamic categories display commonly requested items on the right side of the service catalog homepage, by default. Use dynamic categories to provide users with an access option that automatically adjusts to changing request trends.

To configure dynamic catalog categories:

1. Navigate to **Service Catalog > Catalog Definition > Maintain Dynamic Categories**.
2. Select **New** or open an existing dynamic category.
3. Fill in the fields on the Dynamic category form, as appropriate.

The Dynamic category form contains many of the same fields as the Category form. Additional fields are:

Field	Description
Type	Kind of items to display (requested items or knowledge base articles).
Search how long?	For catalog items only, the period during which to search for the most common requests. For example, enter 7 00:00:00 to display the most commonly requested items in the previous seven days.
Number of Entries	The number of items to display in the dynamic category.

Defining Catalog Items

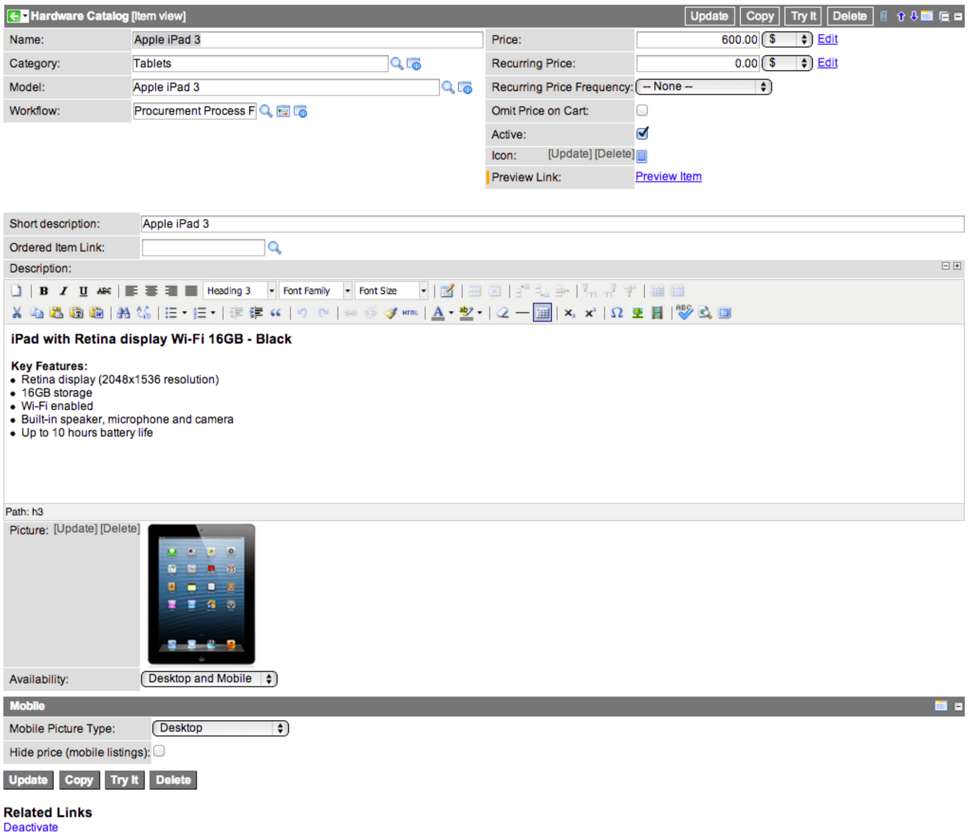
Overview

Catalog items are goods or services available to order from the service catalog. Administrators and users with the catalog_admin role can define catalog items, including formatted descriptions, photos, and prices.

Defining Catalog Items

To define a catalog item:

1. Navigate to **Service Catalog > Catalog Definition > Maintain Items**.
2. Click **New**.
3. Enter the catalog item details (see table).
4. Click **Submit**.



5. Define variables for the item, if applicable.

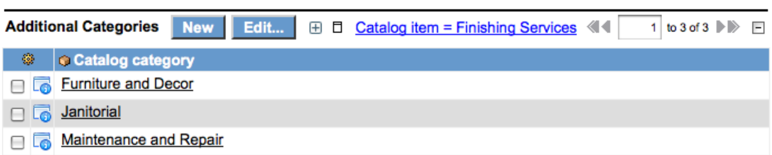
Field	Description
Name	Enter the item name to appear in the catalog.
Category	Select a category for the item. Catalog searches find only items that are assigned to a category.
Model	[Read-only] Click the reference icon ( ) to view the product model to which the item is linked. This field is visible by default only for items created by publishing models (starting with the Dublin release).
Workflow or Execution Plan	Select either a workflow or an execution plan (formerly named delivery plan) to define how the item request is fulfilled. If you select a workflow, the **Execution Plan** field is hidden. Clear the **Workflow** field to select an execution plan.
Price	Set a price for the item and select the currency from the choice list.
Recurring Price	Set a price that occurs repeatedly at a regular interval. For example, a printer maintenance service could have a $100.00 monthly recurring price. For details, see Setting Recurring Price.
Recurring Price Frequency	Select the time frame for recurrence only if the **Recurring Price** field has an entry. For example, **Monthly** or **Annually**.
Omit Price on Cart	Select this check box to hide the item price in the cart and the catalog listing. See Hiding Prices in the Service Catalog.
Active	Select this check box to make this item active (available to be ordered).
Icon	Upload a 16x16 pixel image to appear as an icon beside the item name in the catalog. If no image is uploaded, the default icon appears beside this item.
Preview Link	Click **Preview Item** to preview (in a new window) how the current item definition would look in the catalog.
Short Description	Enter text that appears on the service catalog homepage, search results, and the title bar of the order form.
Ordered Item Link	Specify the record defining a link to more information, as shown on the ordered item screen. See Adding Ordered Item Links for more details.
Description	Enter a full description of the item. This description appears on the item page and when a user points to the item's icon in the catalog.
Picture	[Optional] Upload an image of the item.

Availability	Define which devices the item should be displayed on: **Desktop and Mobile**, **Desktop Only**, or **Mobile Only**. **Note:** Unsupported catalog item types are not displayed on mobile devices, even if the **Availability** is set to show an item of this type. Available starting with the Dublin release.
Mobile picture type	Define the type of picture to display for the item on mobile devices. Set to **Desktop** to use the standard desktop picture, **Mobile** to use a specific image for the mobile device using the **Mobile picture** field, or **None** to not use a picture. Available starting with the Dublin release.
Mobile picture	If **Mobile picture type** is set to **Mobile**, this appears, to choose the image file to upload for the mobile picture. Available starting with the Dublin release.
Hide price (mobile listings)	Select this to hide the item price on mobile devices. Clear the check box to display the price. Available starting with the Dublin release.
Fields that can be added by personalizing the form:	
Template	[Available for Record Producer items only] Select a template to populate the generated record with predefined values.
No quantity	Select this check box to hide the quantity selector for the item. See Hiding the Shopping Cart and Catalog Item Quantity.
No cart	Select this check box to hide the shopping cart for the item. See Hiding the Shopping Cart and Catalog Item Quantity.
Order	Control the ordering of items in category lists.
Related Lists	
Variables	Define variables for the item to provide options for ordering the item. See Using Service Catalog Variables.
Variable Sets	Link an existing variable set to the item to provide multiple options for ordering the item. See Service Catalog Variable Sets.
Approved By Group	Add the groups that have to approve requests for this item. See Approval Rules.
Approved By	Add the users who have to approve requests for this item. See Approval Rules.
Additional Categories	Add any aditional categories associated with this item, besides the category selected on the form. See Assigning Multiple Categories.

Assigning Additional Categories

A catalog item may be available from more than one category. For example, a laptop carrying case may be available from both *Laptops* and *Cases and Accessories*.

To assign an item to more than one category, enter one category in the **Category** field on the Catalog Item form and the other categories in the **Additional Categories** related list.

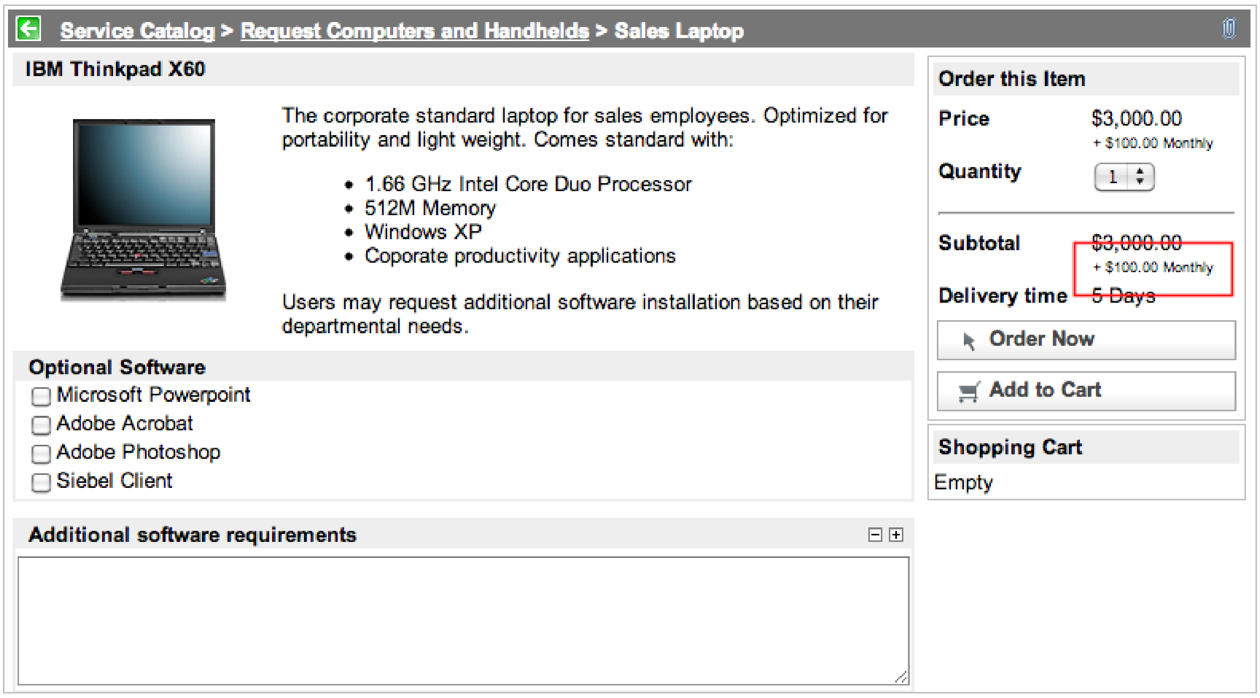


Additional Categories

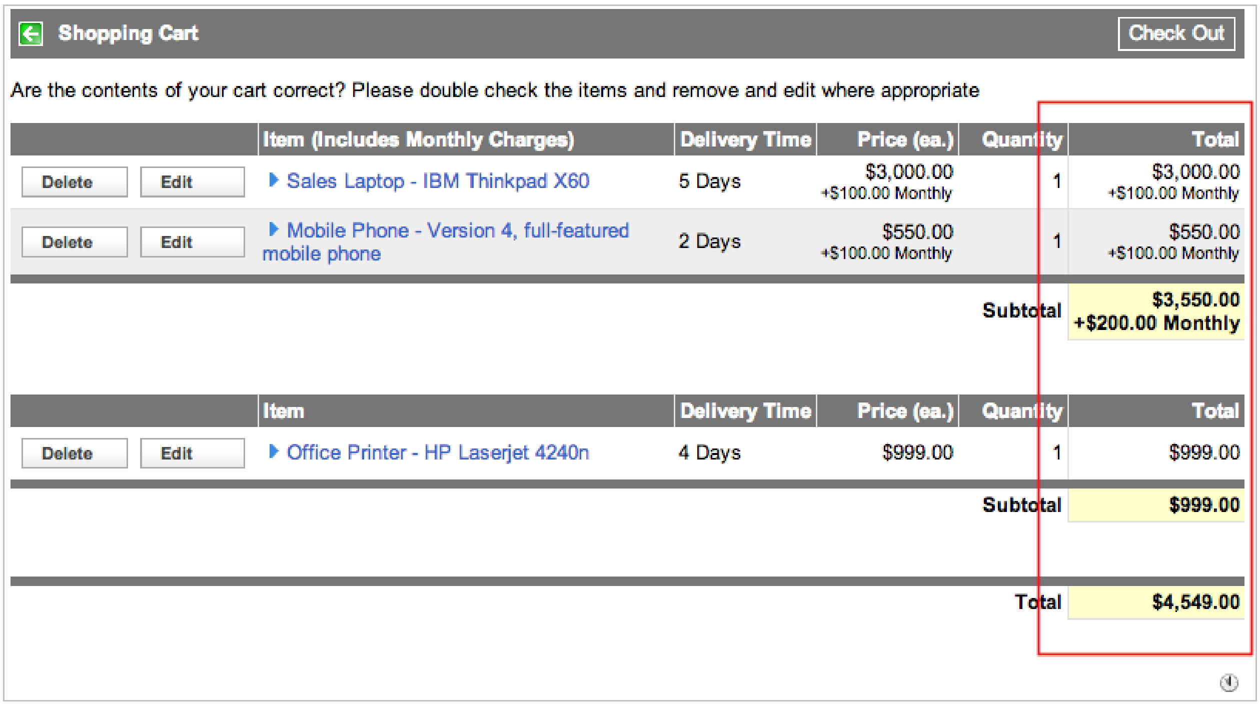
Setting Recurring Price

A catalog item can have a recurring price, as well as an initial price. For example, a subscription to a mobile phone contract may cost $500.00, with an $30.00 monthly recurring price. The recurring price feature is available starting with the Calgary release.

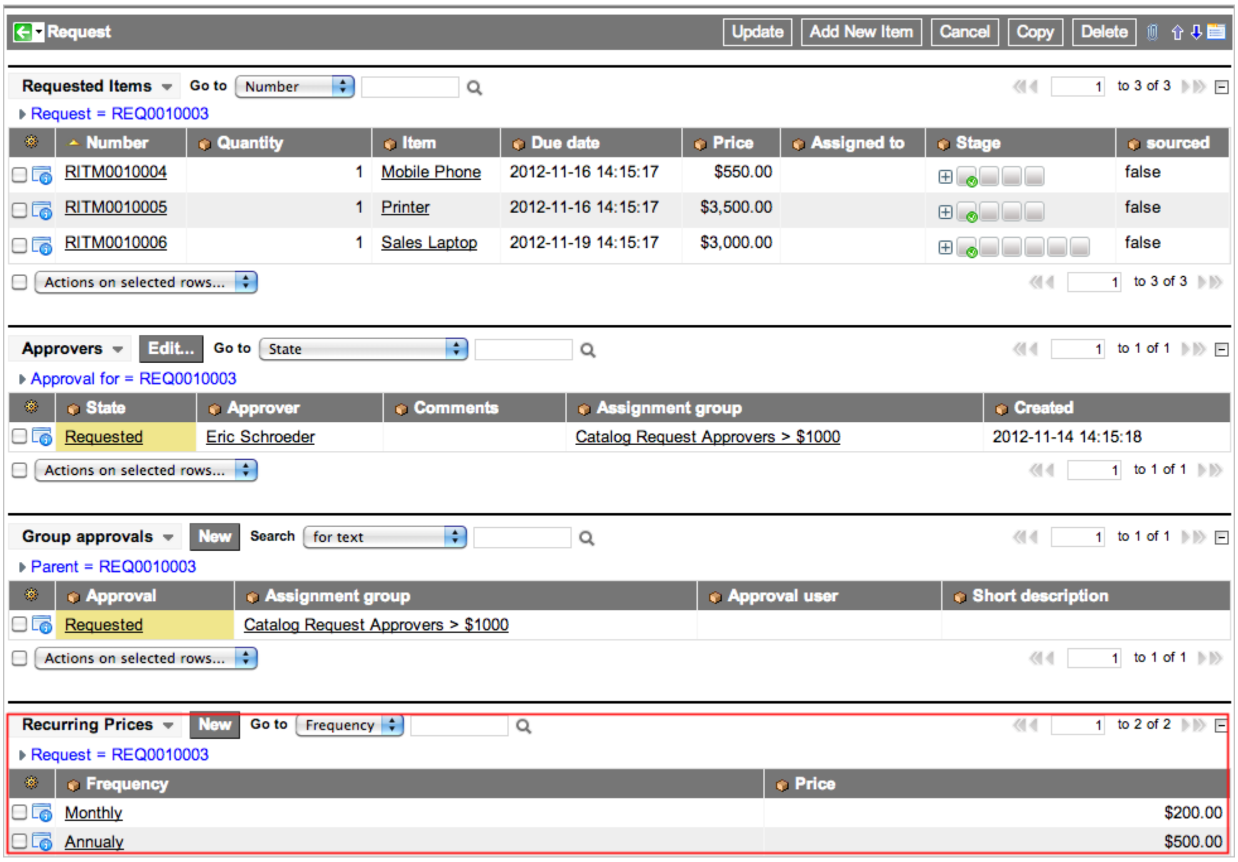
The price and the recurring frequency are set on the catalog item record. After the price and frequency are set, the recurring price appears in the catalog, catalog search results, the catalog page for the item, the shopping cart, and the order summary screen.



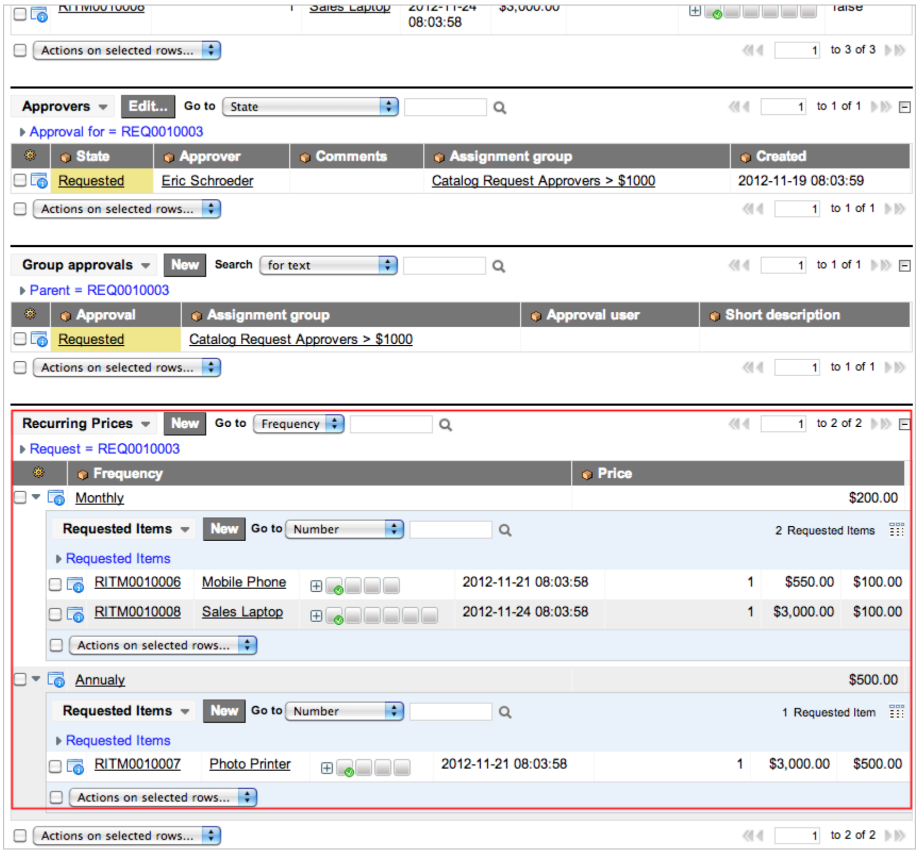
If multiple items with the same recurring price frequency are placed in the shopping cart, they are grouped together. The grouping makes it easier to see how much items cost for each frequency (for example, weekly, monthly, and annually). If the shopping cart contains items with and without recurring costs, they are grouped separately.



On a Request record, recurring prices are grouped by frequency and shown in the **Recurring Prices** related list. In the example below, two items each have a monthly recurring cost of $100.00 and their prices are grouped as a single record of $200.00 monthly. Another item with an annual recurring cost of $500.00 is listed as a separate record.



If a Request record contains multiple items with the same recurring frequency, click the arrow next to the corresponding Recurring Prices record to view details. In the example below, two items (the mobile phone and the sales laptop) each have a monthly recurring cost and are grouped together under the **Monthly** Recurring Prices record, but listed separately when the Recurring Prices record is expanded. Only one item has an annual recurring cost.

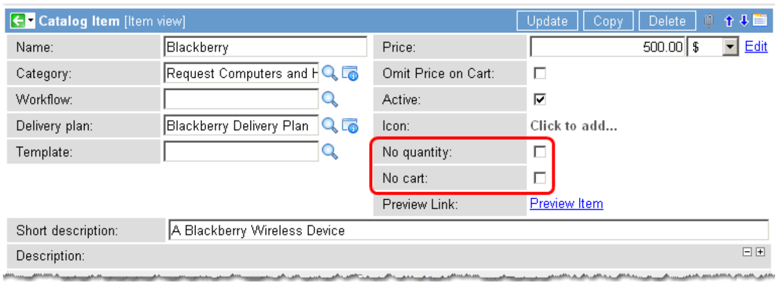


Hiding the Shopping Cart and Catalog Item Quantity

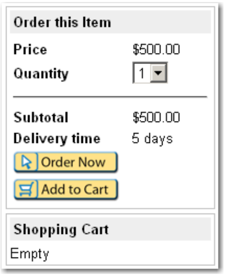
To hide the shopping cart or quantity for a service catalog item:

1. Navigate to **Service Catalog > Catalog Definition > Maintain Items**.
2. Select the catalog item.
3. Personalize the form to add the **No quantity** and **No cart** fields.
4. Select the **No quantity** check box to hide the quantity selector for anyone ordering that item.

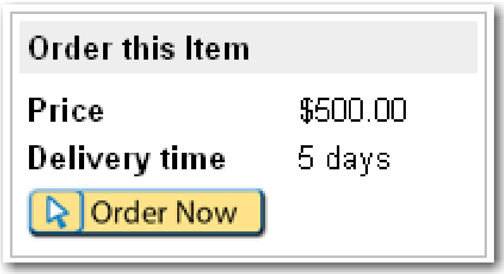
   Alternatively, you can make the quantity selector visible to certain roles with the **List of roles (comma-separated) that can use the quantity selector in the shopping cart** property (`glide.sc.allow.quantity`) on the Service Catalog Properties page.
5. Select the **No cart** check box to hide the shopping cart for the item.



Catalog item definition



Visible quantity and shopping cart



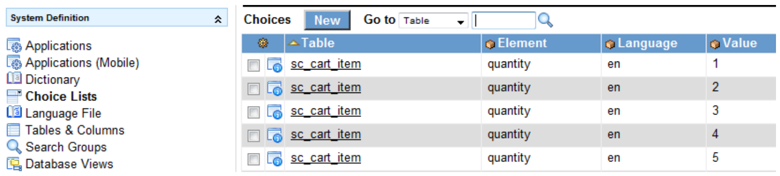
Hidden quantity and shopping cart

Defining Item Quantities

By default, the service catalog offers quantity choices from 1 to 5, allowing customers to select item quantities within that range. This quantity selection applies automatically to all items, assuming these items do not have the quantity selector hidden.

To increase or reduce the quantities that a customer can select for catalog items:

1. Navigate to **System Definition > Choice Lists**.
2. Filter the list for records with **Table** set to **sc_cart_item** and the **Element** set to **quantity**. The existing quantity choices appear.
3. Add quantity choices, modeling them after the existing ones.



Choice lists

For example, to increase the quantity range from 1-6, add a new **6** record.

To reduce the quantities available for catalog items, delete the relevant quantity records. For example, to reduce the quantity range to 1-3, delete the records for **4** and **5**.

Adding Ordered Item Links

Catalog administrators can provide a link on the ordered item screen, linking to more information about an item. After users order the item, they can click the link to receive details about the item. The URL text and link can be defined once and reused across multiple items - for example, for standard ordering terms and conditions. Ordered item links are available starting with the Calgary release.

Links are rendered with an added parameter providing the sys_id of the requested item. For example, if the link is mylink.com, it is rendered as mylink.com?req_item=abcde12345. This can be useful for virtual provisioning situations.

To create an ordered item link:

1. Navigate to **Service Catalog > Catalog Definition > Ordered Item Links**.
2. Click **New**.
3. Enter a **Name** for the ordered item link.
4. Enter **Link text** to display as the link. For example, **Click here to see more information about the iPhone**.
5. Enter the exact **Link URL**. For example, http://www.mylink.com.
6. Click **Submit**.

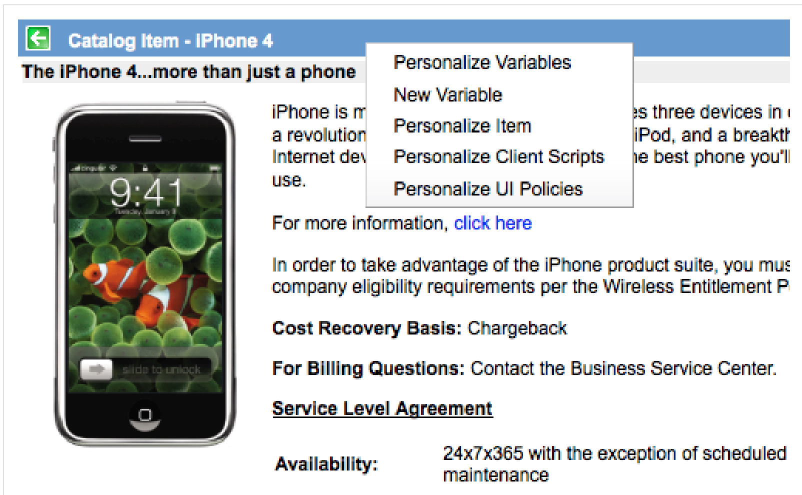
To add an ordered item link to an item:

1. Navigate to **Service Catalog > Catalog Definition > Maintain Items**.
2. Click an item **Name**.
3. In the **Ordered Item Link** field, click the reference lookup icon.
4. Click an ordered item link **Name**.
5. Click **Update**.

Editing Catalog Items from the Service Catalog

To edit a catalog item from the service catalog listing, right-click the header and select one of the following options:

- **Personalize Variables:** add or remove variables from an item.
- **New Variable:** create a new variable for the item.
- **Personalize Item:** edit the item definition.
- **Personalize Client Scripts:** edit the catalog client scripts for the item.
- **Personalize UI Policies:** edit the catalog UI policies for the item.



Catalog item listing

Service Catalog Variables

Overview

Administrators and users with the catalog_admin role can define service catalog variables.

A service catalog variable provides the ability to capture and pass on information about choices a customer makes when ordering a catalog item. Some variables can be defined to affect the order price, depending on the selected value.

For example, a *New PC* catalog item can use a variable called "Memory", which provides choices to allow customers to select extra memory, for associated extra prices.

Variables can be stored, accessed from multiple places, and passed between tasks in a process when fulfilling a request.

Variables can be displayed on the Requested Item and Catalog Task forms after an item has been ordered.

ServiceNow provides a full set of variable types.

Defining and Associating Variables

Variables can be associated with individual catalog items, so that they are available in any request for the item (regardless of the workflow or execution plan used). Variables can also be associated with variable sets and re-used across multiple catalog items.

See Using Service Catalog Variables for instructions on adding service catalog variables to catalog items.

Passing Variables Between Tasks

Variables are passed from one step of the fulfillment process to another. They can be processed by any workflow or execution plan that is associated with the requested items.

For example, consider the following tasks within the fulfillment process for a *New PC* catalog item.

1. Procure PC.
2. Install corporate standard software.
3. Set up email account.

4. Deliver and set up PC for requester.

Step 4 may require a piece of information from step 3 (the email account credentials for setting up email on the PC). If steps 3 and 4 are executed by different fulfillment groups, you can use variables to make this information visible in the tasks for steps 3 and 4, and to pass it between groups so that the second group can access the email account credentials.

Managing Request Fulfillment

Managing Request Fulfillment

Overview

When a user orders a catalog item, ServiceNow creates a request and attaches the catalog item attached to it. The processing of this request (*request fulfillment*) is driven by a fulfillment process that must be defined.

This process lets administrators automate requesting approvals, assigning requests, and fulfilling requests, using tools similar to those used elsewhere in task administration or workflow.

To define the fulfillment process, administrators need to:

1. Set up fulfillment groups to perform the work.
2. Define the fulfillment processes those groups use to perform the work.

Setting up Fulfillment Groups

Fulfillment groups perform the tasks related to fulfilling an order. This can include approving an order based on characteristics such as content and price, or any direct action required to complete the order, such as loading software or installing hardware. Any existing user group (in **User Administration > Groups**) can be assigned fulfillment tasks.

To create a group specifically for order fulfillment:

1. Navigate to **Service Catalog > Catalog Policy > Fulfillment Groups**.
2. Click **New**.
3. Fill in the Group form as described under creating groups.

   These groups have the type **catalog** and are assigned the catalog and itil roles, but are otherwise normal groups.

Defining Fulfillment Processes

Each catalog item uses a fulfillment process, to define the request fulfillment process when that item is ordered.

Fulfillment processes are used when ordering standard catalog items, but are not used for some extended types of catalog item, such as content items.

Selecting a Fulfillment Process

Request fulfillment processes can be defined using either a *workflow* or (less typically) an *execution plan*.

Note: *ServiceNow typically recommends using workflows for your request fulfillment processes.*

Workflows

Service catalog workflows allow administrators to easily define a complex, multi-step process for fulfilling and approving the request.

Service catalog workflows can be defined using the graphical workflow editor, allowing you to:

- Edit workflows graphically.
- Modify activities and conditions.
- Define transitions between workflow activities.
- Summarize workflow progress through stages.
- Validate workflows to identify potential problems
- Publish workflows for other users.

ServiceNow typically recommends using workflows for request fulfillment processes.

For more information, see Defining a Service Catalog Workflow.

Execution Plans

Execution plans allow you to describe simple, linear processes.

Although execution plans are useful in some circumstances (for example, if you need to build your processes programmatically or through imports), ServiceNow typically recommends using workflows for request fulfillment processes.

For more information, see Using Execution Plans.

Service Catalog Workflows

Overview

Administrators can use the graphical workflow editor to create workflows to drive request fulfillment. Workflows can generate and assign approvals, generate and assign tasks, or run scripts and subflows.

Service catalog workflows can be attached to particular requests in one of the following ways:

- Manually, on the Catalog Item form
- Automatically based on conditions
- Automatically if there are no other workflows attached

See Defining a Service Catalog Workflow for more details.

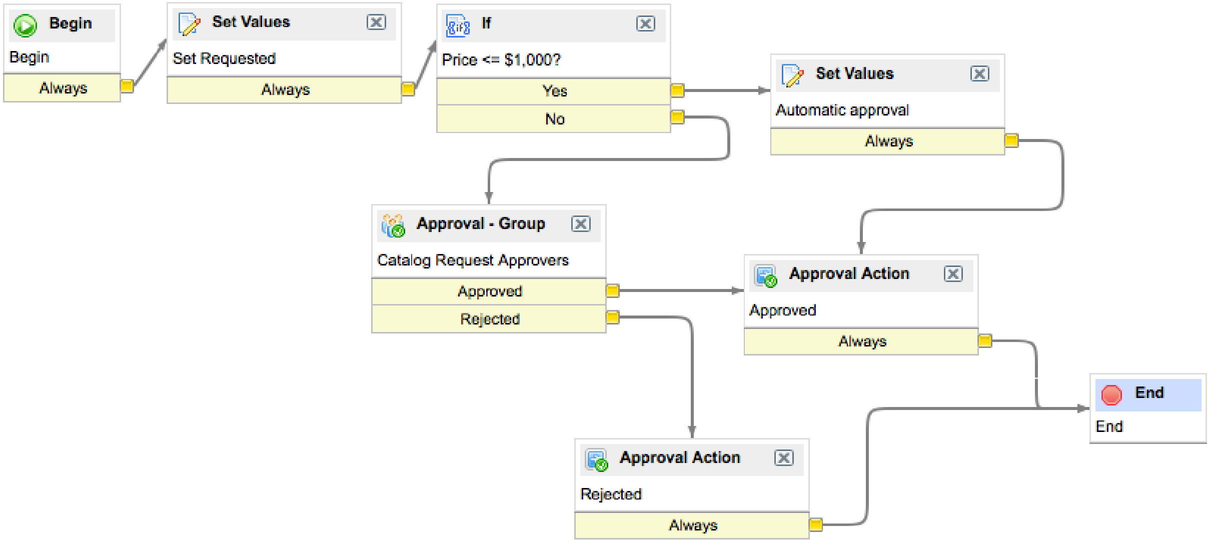
Default Service Catalog Workflows

The following service catalog workflows are provided with the base system:

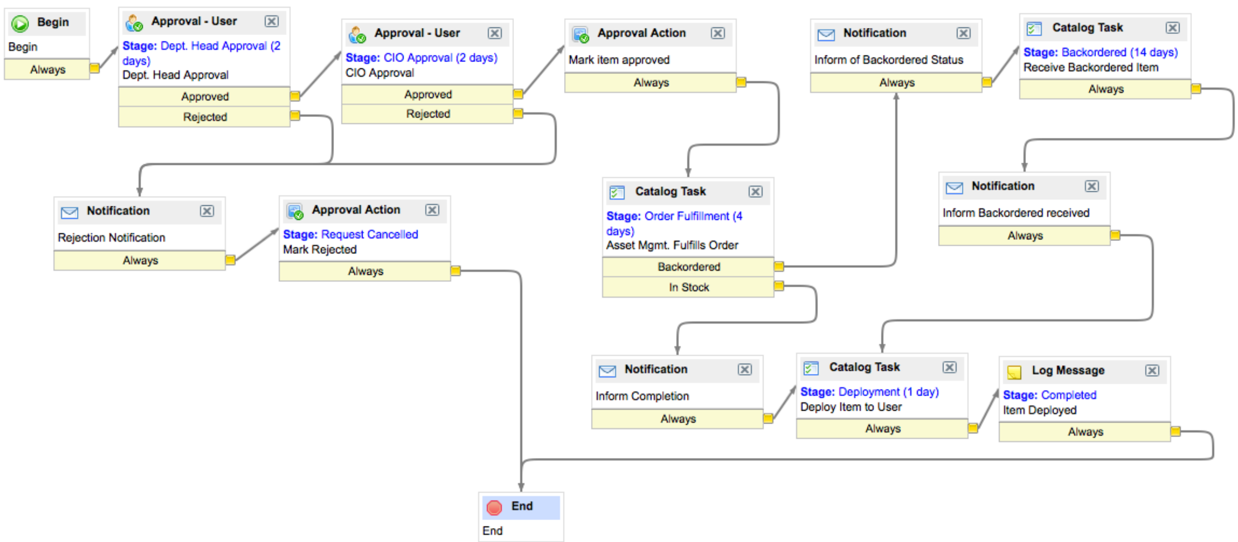
- **Service Catalog Request:** a simple workflow that fulfills a simple order
- **Service Catalog Item Request:** a more complex workflow that fulfills a more complex order.

Use these examples to see how the workflow engine can work for the service catalog.

Service Catalog Request



Service Catalog Item Request



Deleting Service Catalog Workflow Records

Workflow stage fields for service catalog workflows display when a user deletes a record required by the workflow (Dublin).

If a user deletes a catalog item and that catalog item has active request workflows running, the workflow stage field displays **Catalog item removed**. Similarly, deleting the associated workflow context causes the stage field to display **Workflow context removed**.

Using Execution Plans

Overview

An execution plan describes how a catalog item is procured, configured, and installed. Each execution plan contains one or more tasks.

For example, an organization might create an execution plan for delivering a corporate standard PC that contains these tasks:

1. Procure the PC from a supplier.
2. Configure the PC according to the requester's specifications.
3. Deliver the PC to the requester.

An execution plan is not specific to any one catalog item. There could be many different models of PC that a user can order, all using the same execution plan. It is usually not necessary to create a new execution plan for each individual catalog item in a mature service catalog.

Note: *Execution plans are not as powerful or flexible as workflows, and cannot be designed using a graphical editor. Execution plans are useful in some circumstances (for example, if you need to build your processes programmatically or through imports), but ServiceNow typically recommends using workflows for request fulfillment processes.*

Execution Plan Tasks

An execution plan contains one or more execution plan tasks. Each task represents work that needs to be completed by a specific group as part of the overall request fulfillment process.

Administrators create templates for tasks as part of defining an execution plan. The catalog tasks themselves are then created when the relevant item is requested, based on these task templates.

See Creating Execution Plan Tasks for more information.

Using Execution Plans

Once you have created the execution plan record, you can then associate it with catalog items, create task templates, and associate any catalog item variables to tasks if required.

You can also:

- Apply conditions to tasks.
- Create approval tasks, allowing you to "roll back" to a previous step if the approver rejects the request.

Extended Catalog Item Functions

Using Order Guides

Overview

An order guide gives customers an easy way to order multiple related items as a single request, It uses :

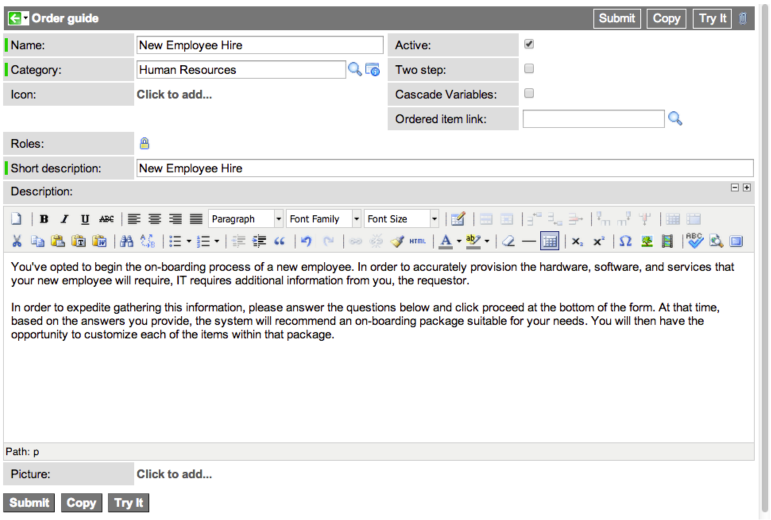
- A single initial screen, where the customer fills in some initial information.
- A set of selected catalog items based on conditions derived from the initial information.

For example, a New Employee Hire order guide contains items that new employees typically receive, from which the customer selects appropriate items and options. If the new employee selects the optional business cards, that item is then included as part of the request.

Creating Order Guides

To create an order guide:

1. Navigate to **Service Catalog > Catalog Definition > Order Guides**.
2. Click **New**.
3. Enter these order guide details and save the record.
    - **Name:** enter the guide name to appear in the catalog.
    - **Two step:** select the check box to enable two step ordering (rather than the default three step ordering).
    - **Category:** select the category heading under which the guide appears.
    - **Short description:** enter a summary of the order guide purpose.
    - **Description:** enter a description that appears on the first page of the order guide. Apply formatting by using the HTML editor.
4. Define any cascading variables.
5. Define any rules for including items in the order guide.



Order guide definition

Enabling Two Step Order Guides

By default, order guides present the user with a three step process:

1. **Describe Needs:** the initial screen
2. **Choose Options:** filling in details for the items ordered
3. **Check Out:** review and, if necessary, edit item information before submitting the request

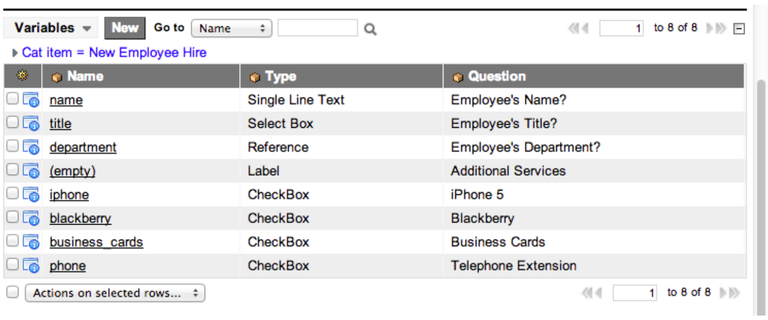
Order guides allow for an optional two step mode that omits the Check Out step. Upon completion of the Choose Options step, the selected items are automatically ordered.

To set an order guide to use the two step checkout model, select the **Two step** check box in the Order guide form.

Adding Cascading Variables

You can add cascading variables to an order guide, to be used by catalog items requested.

1. Open the order guide definition.
2. In the **Variables** related list, click **New**.
3. [Required] Enter a **Name** so this variable can be used in the rule base.
4. Enter the remaining variable details and save the record.

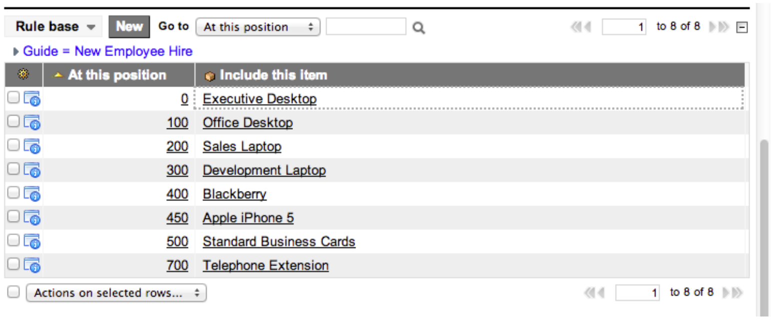


Order guide cascading variable

Note: *These variables must have corresponding variables defined in the relevant catalog items. See Cascading Order Guide Variables for more information.*

Creating Rules

The *rule base* lists all the rules that apply to the order guide. These rules determine the conditions under which items are included in the order guide.



Rule base

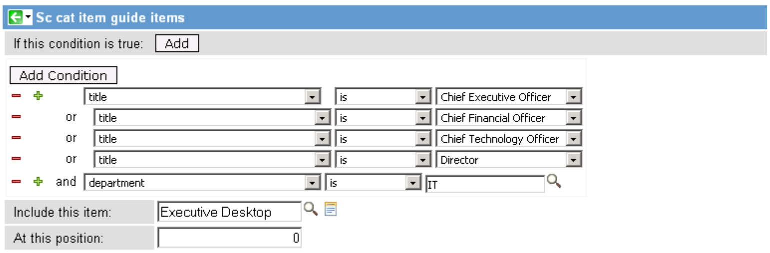
To create rules for an order guide:

1. Open the order guide definition.
2. In the **Rule base** related list, click **New**.
3. Set these values:
   - One or more conditions that must be true for this rule to apply.
   - The item to include in the order guide if this rule is true.
   - The position within the order guide to place this item.
4. Click **Submit**.

Setting rule conditions

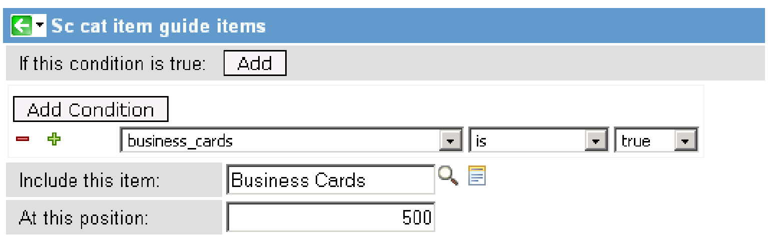
Any question asked of the user in the order guide can be used as part of a condition.

For example, this rule states that if the new employee's job title is CEO or CFO or CTO or Director, and the department is IT, then an executive desktop is included in their New Hire order.



Example 1

This rule states that if the new employee selects the business_cards check box, then business cards appear in the order.



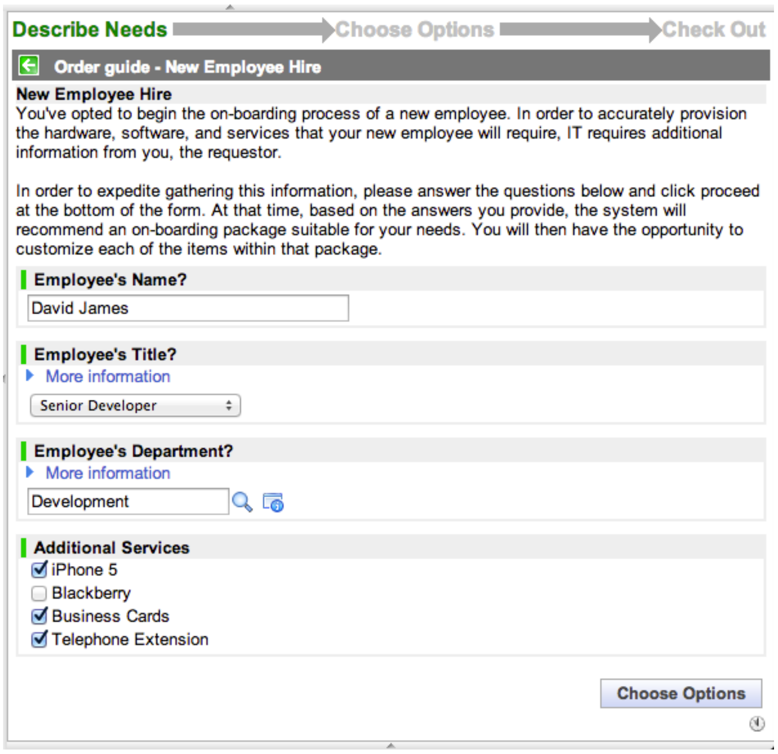
Example 2

Example of use

This example shows an order guide in operation.

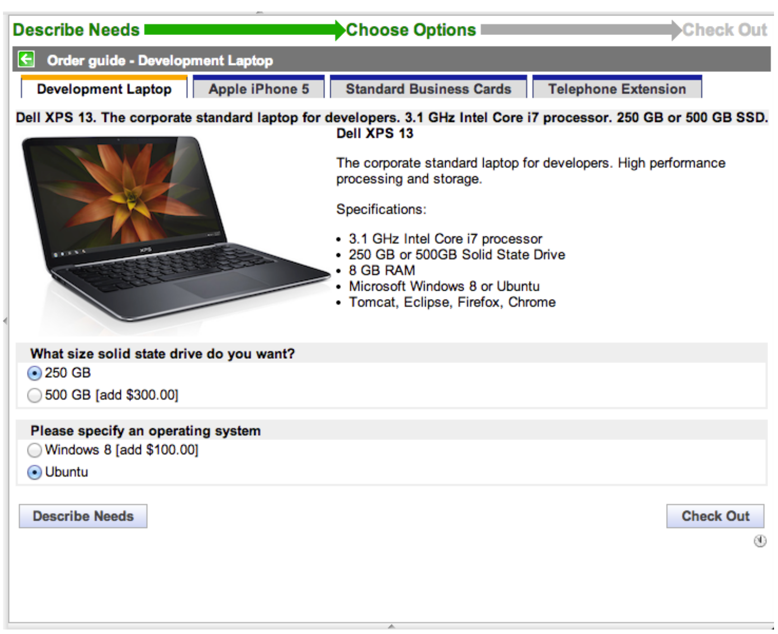
1. The system prompts the user for information:

2. The system uses the rule base to generate options:

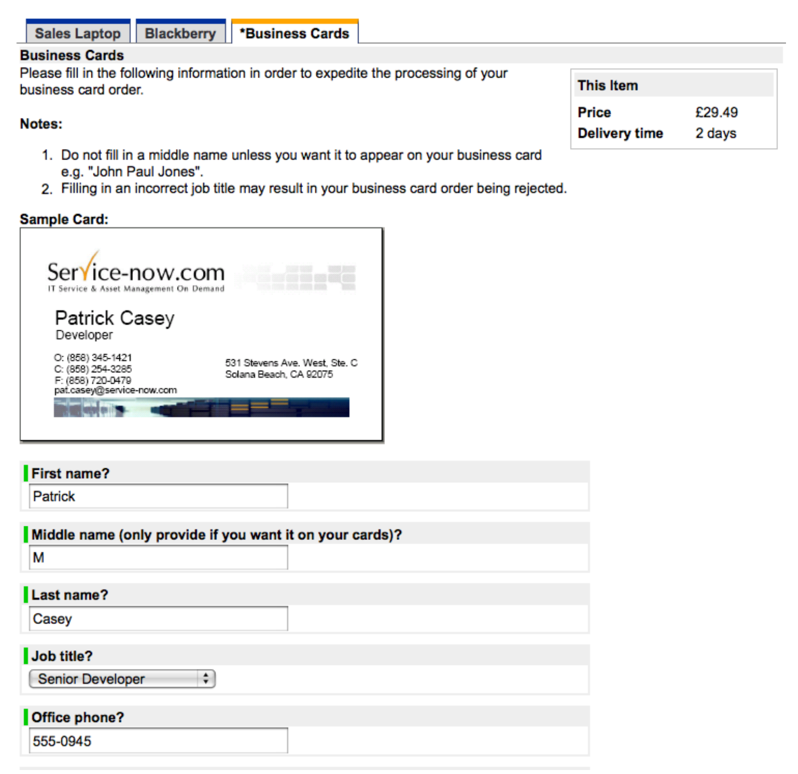


Order guide

3. The user configures each item:

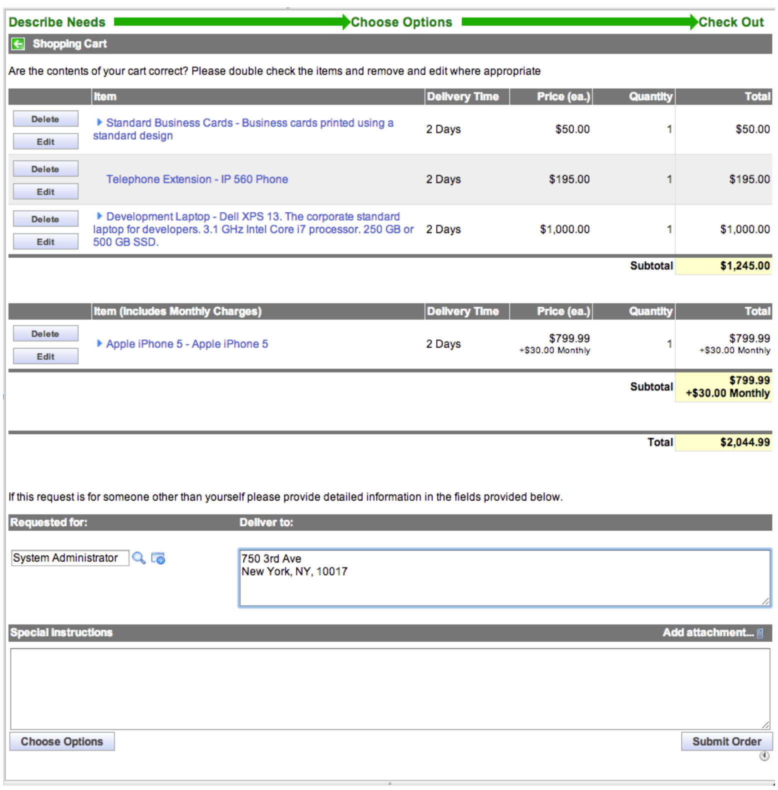


Order guide



Order guide

4. The system takes the user to checkout:



Order guide

Creating Record Producers

Overview

A record producer is a specific type of catalog item that allows end users to create task-based records (such as incident records) from the service catalog. Record producers provide a simplified alternative to the regular ServiceNow form interface for creating records.

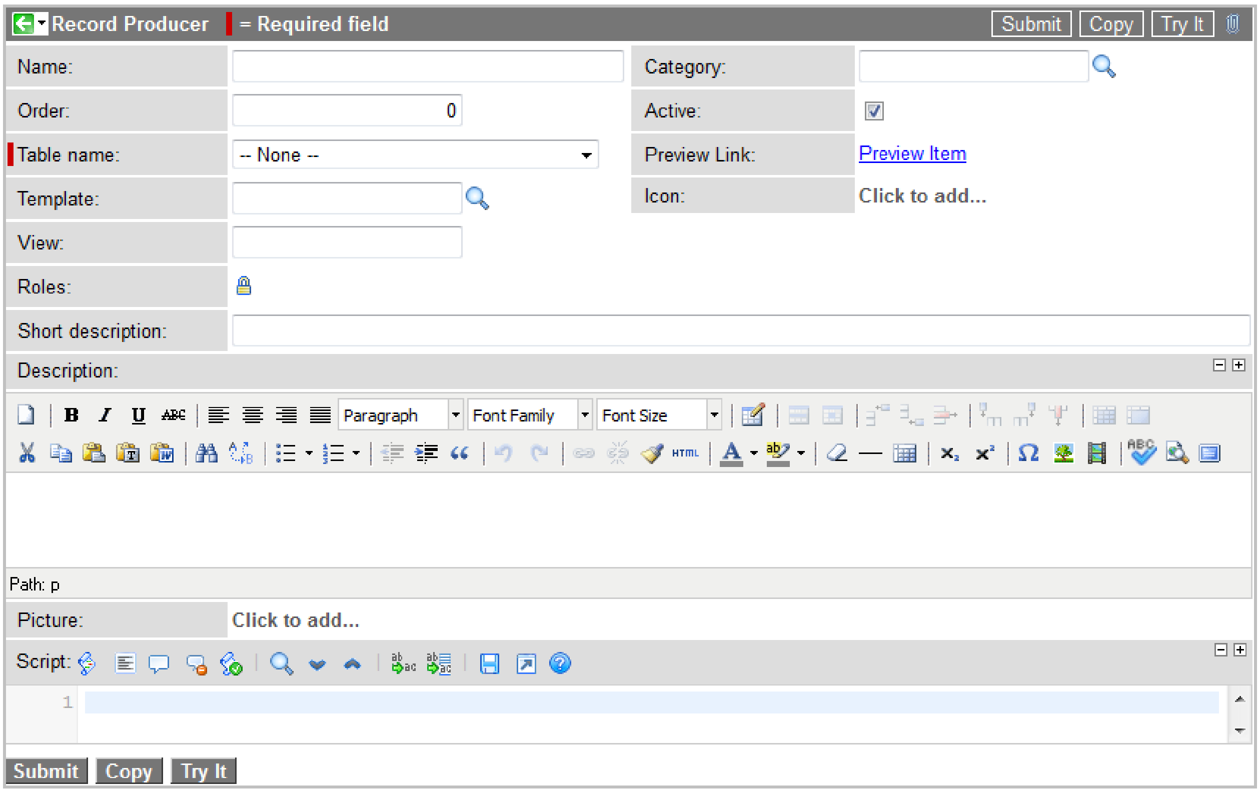
You can define a record producer through the service catalog or from the table record.

Note: *ServiceNow does not recommend creating Requested Item records from record producers. Requested Item records should only be created using catalog items.*

Defining Record Producers Through the Service Catalog

1. Navigate to **Service Catalog > Catalog Definition > Record Producers**.
2. Click **New** or select the record producer to edit.
3. Fill in the fields on the Record Producer form (see table).
4. Click **Submit**.

   After you submit the form, ServiceNow adds the **Variables** and **Variable Sets** related lists.
5. Open the record again to define variables for the record producer.



Field	Description
Name, Category, Order, Icon, Roles, Active, Short description, Description, Picture	As for standard catalog items.
Table name	Table in which the record producer creates records.
Template	Static assignments for fields on the created record.
View	CMS views in which the item is visible.
Preview Link	Link that opens a preview of the item.
Script	Scripts that should be run to dynamically assign values to specific fields on the created record.

Populating Record Data

To populate data with a record producer, use any combination of the following methods:

- Create a variable on the record producer with the same name as the field in the target record. For example, a variable named `caller_id` on a **Create a New Incident** record producer populates the **caller_id** field on the new incident record. Use a variable type that corresponds to the field type.
- Define a template to assign a static field value for all records created by the record producer.
- Define a script that uses any variable or server-side objects and functions to assign values.
  - Use *current.*FIELD_NAME** to reference fields on the record being created.
  - Use *producer.*VARIABLE_NAME** to reference values entered by the end user.

Redirect After Submitting a Record Producer

To redirect an end user to a particular page after submitting a record producer, define the redirect link in the **Script** field using *producer.redirect*. For example, the following code redirects a user to their homepage after the record producer is submitted:

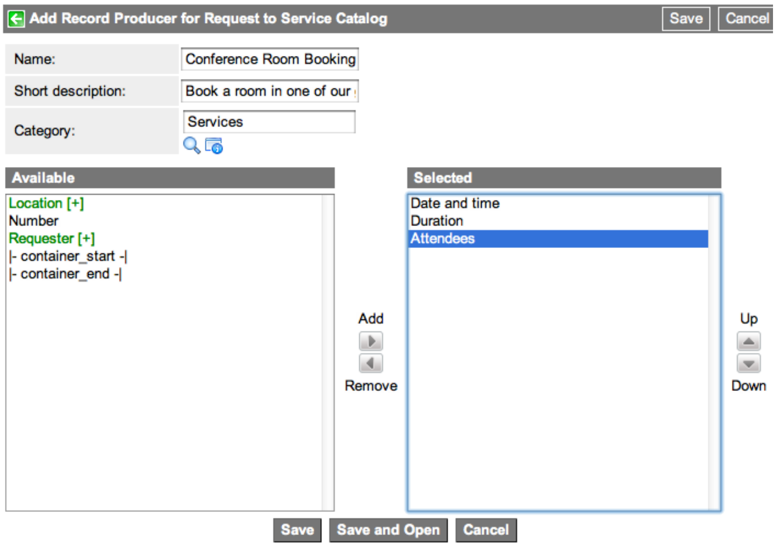
```
producer.redirect="home.do";
```

Creating Record Producers from Tables

To save time, you can create service catalog record producers directly from a table record (starting with the Dublin release). In versions prior to Dublin, you can only create record producers through the service catalog.

To create a record producer from a table record:

1. Navigate to **System Definition > Tables** and open the table record.
2. Under **Related Links**, click **Add to Service Catalog**.
3. Complete the **Name**, **Short Description**, and **Category** fields as you would for service catalog items.
4. Use the slushbucket to select the fields and the order in which you want them to appear. To use container variables, select **|- container start -|** and **|- container start -|**.



Record producer

5. Click **Save and Open** to open the record producer and define additional options. Alternatively, click **Save** to return to the table record.
    - A record producer is created with these values:
        - **Table name:** table record opened in step 1
        - **Name**, **Short Description**, and **Category:** information entered in step 3
    - A variable is created for each of the selected fields with these values:
        - **Name: Column name** of the field
        - **Type:** variable type that corresponds to the field type
        - **Order:** position selected in the slushbucket (for example, **100** for the first field and **200** for the second field)
        - **Question: Column label** of the field
    - If the field type is **Choice**, a corresponding question choice is created for each field choice.
    - The new record producer is added to the **Record Producers** related list on the table record.

Creating Variables for Field Types

Use these variable types to collect data for the corresponding field types.

Field type	Recommended variable type
True/False	Yes/No
Reference	Reference
Date or Due Date	Date
Date/Time	Date/Time
Choice or any field with an associated choice list	Select Box
HTML	HTML
List or UI Action List	List Collector
String with length greater than 256	Multi Line Text
String, Integer, Decimal, Floating Point Number	Single Line Text
All other field types	Multi Line Text

Adding Content Items

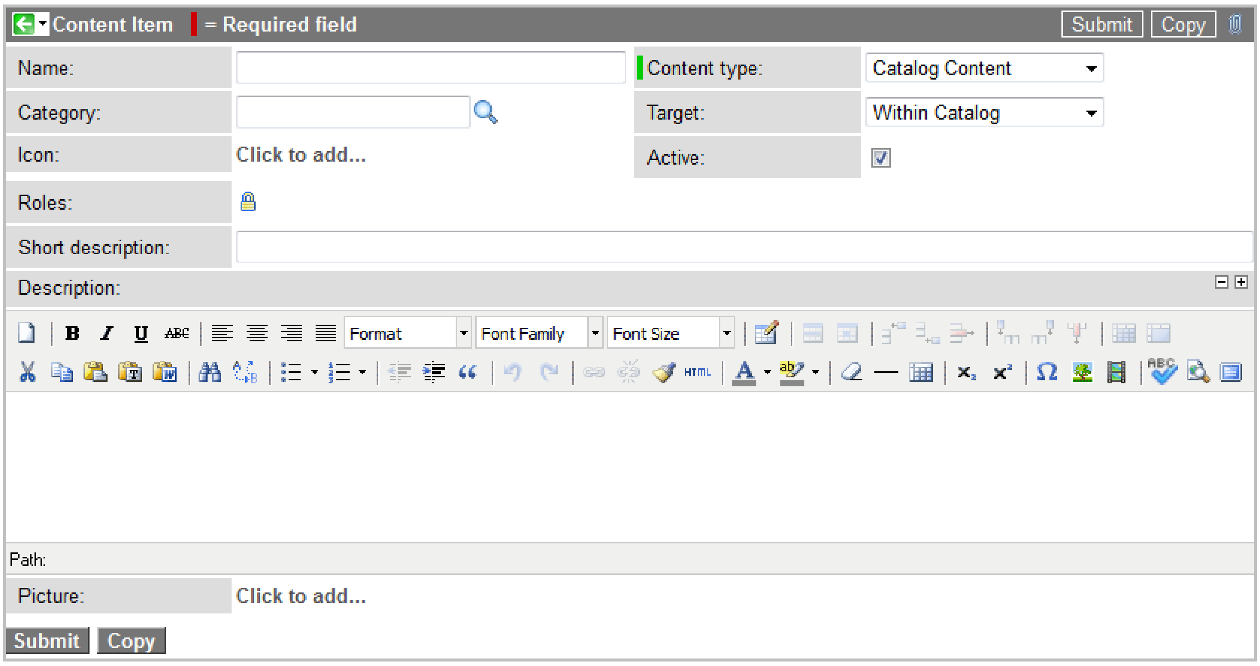
Overview

A content item is a service catalog item that provides information instead of goods or services. Content items may reference knowledge articles, static blocks of text, or external web-based content.

Content Items

To define content items:

1. Navigate to **Service Catalog > Catalog Definition > Content Items**.
2. Click **New**.
3. Fill in the form to define the item (see table).
4. Click **Submit**.



Field	Description
Name, Category, Icon, Roles, Active,	As for standard catalog items.
Content type	The type of information to display when a user selects the item. • **KB Article:** a ServiceNow knowledge article available to users with the specified roles. • **Catalog Content:** a static block of text, formatted with HTML. • **External Content:** documents or web pages stored outside the ServiceNow instance.
Target	Location where the content appears. • **Within Catalog:** displays the content within the catalog iframe window (which may not render some external websites properly). • **New Window/Tab:** displays the content in a new browser window or tab. Choose this option if the content does not display properly within the catalog.
URL	The full URL (including the prefix http://'" **or** https://'") for external content to display when a user selects the item. This field is available only if the **Content type** is **External Content**. **Note:** Since ServiceNow instances are accessed via HTTPS, links to HTTP sites may result in a mixed mode content error or warning, depending on browser security settings.

KB article	The knowledge article to display when a user selects the item. This field is available only if the **Content type** is **KB Article**.
Short description, Description, Picture	As for standard catalog items. These fields are only available if the **Content type** is **Catalog Content**.

Extended Variable Usage

Service Catalog Variable Sets

Overview

The process of creating variables can be repetitive if there is a large number of similar catalog items with the same set of questions. For example, a catalog administrator can define ten types of servers, all of which ask the same five questions (using the same variables).

A *variable set* is a modular unit of variables that is defined once, but can be shared between catalog items. Changes made to a variable set affect all items that use the variable set.

Note: *Variables in a set use the same rules as other variables to determine when the variables in a set appear on a task. For example, variables must either be global or be attached directly to an item.*

Creating a Variable Set

Following are the basic steps for creating a variable set.

1. Navigate to **Service Catalog > Catalog Variables > Variable Sets**.
2. Click **New**.
3. Enter a **Name**.
4. Enter an **Order** number. (For more information, see Variable Set Order.)
5. Select **Display title** to give the variable set a title. (For more information, see Adding a Title and Header Bar to Variable Sets.)
6. Enter text in the **Title** field.

   Titles are translated using the set name as the language key.
7. Select a **Layout** from the choice list.
8. Enter a **Description**.
9. Click **Submit**.

Adding Variables to a Variable Set

After you create a variable set, a related list appears for adding variables.

1. Navigate to **Service Catalog > Catalog Variables > Variable Sets**.
2. Click a variable set **Name**.
3. In the **Variables** related list, click **New**.
4. Follow the standard steps for creating variables.

Linking a Set to a Catalog Item

A variable set can be linked to one or more catalog items. To link a variable set to a catalog item:

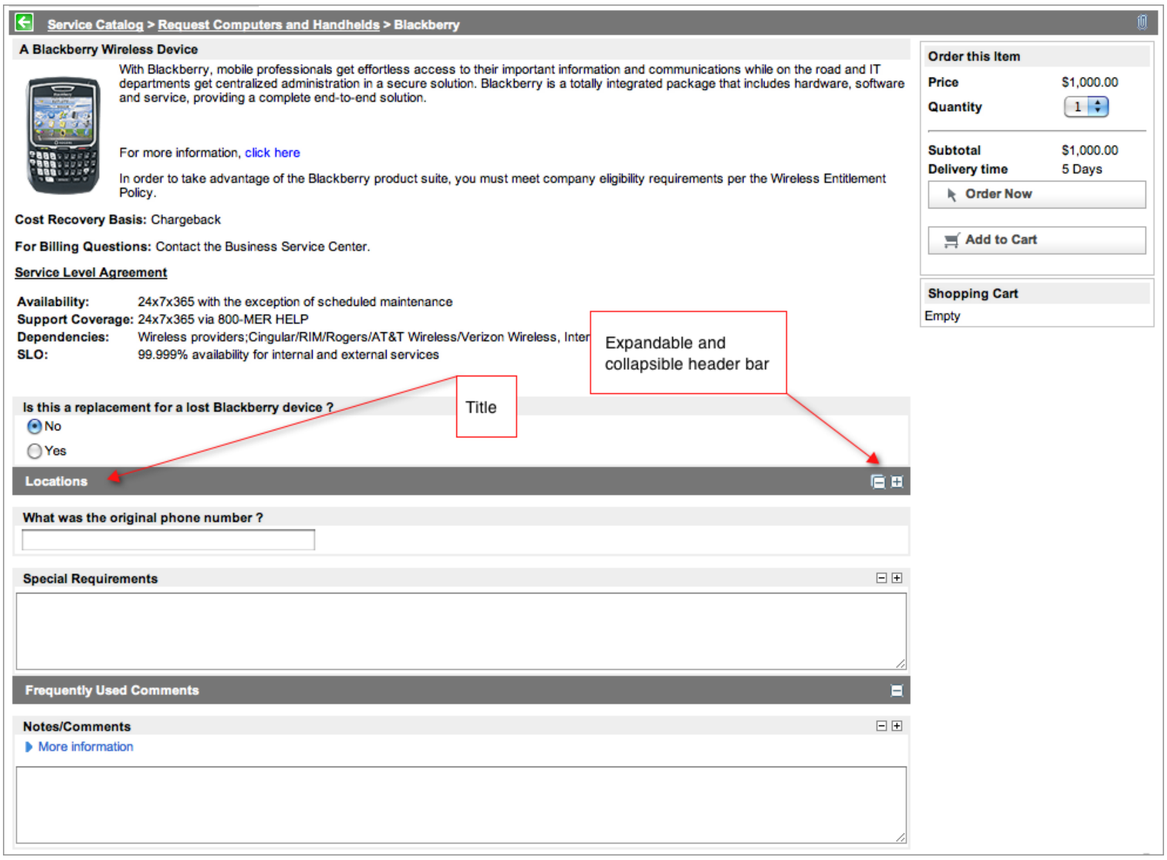
1. Navigate to **Service Catalog > Catalog Definition > Maintain Items**.
2. Click the **Name** of a catalog item.
3. In the **Variable Sets** related list, click **Edit**. (If the **Variable Sets** related list is not visible, personalize the form.)
4. In the **Collection** list, select a variable set.
5. Click the **Add** button to move the variable set to the **Variable Sets List**.
6. Click **Save**.

Adding a Title and Header Bar to Variable Sets

Variable sets can display an optional title, along with a section header bar to collapse and expand the section, when the relevant catalog item is requested.

1. Navigate to **Service Catalog > Catalog Variables > Variable Sets**.
2. Select a variable set.
3. Select **Display Title** to give that variable set its own sub-heading with collapse and expand buttons on the right.

   If an item uses multiple variable sets, only those sets with **Display Title** selected show the title and header bar.



Note: *When a user requests the item, all check box variables are grouped together under a default heading of* ***Options****. To override this default heading, insert a variable of type* ***label****, with an order that puts it directly above the check box variables.*

Variable Set Order

A variable set is a discrete unit with an order number. All variables within the set are included wherever the set is included.

For example, when specifying the following order for some example variables and a variable set:

- Variable A (100)
- Variable B (200)
- Variable C (300)
- Variable Set 1 (250) - containing three variables (VS1, VS2, VS3) with incremental order values (150, 250, 350)

The order in which the variables are displayed is:

- Variable A (100)
- Variable B (200)
- Variable Set 1 (250): Variable VS1 (150)
- Variable Set 1 (250): Variable VS2 (250)
- Variable Set 1 (250): Variable VS3 (350)
- Variable C (300)

Note: *To force variable sets to merge with each variable placed according to its individual order regardless of inclusion in a variable set, set the property* `glide.catalog.set.compatibility.merge` *to* ***true****.*

Reference Qualifiers for Service Catalog Variables

Overview

Service catalog reference variables can use advanced reference qualifiers to restrict the available values for choice-based variable types based on the result of a script or other variables on the catalog item.

For example, you could create a *Location* reference variable with a reference qualifier that restricts the available location choices to those associated with groups in which the **Requested For** user is a member.

Note: *This functionality requires knowledge of scripting.*

The information on this page is for Calgary and later releases. For Berlin and earlier releases, see Advanced Reference Qualifiers for Service Catalog Variables.

Using Reference Qualifiers

The `ref_qual_elements` attribute can be used to specify dependent fields in the following cases:

- Catalog Item ordering screen
- Requested Item form
- Catalog Task form

In the JavaScript function called by the advanced reference qualifier, the current record (`current`) is in scope, allowing access to additional information. For example:

- Variable information for the ordered item is available with `current.variables.`*`variablename`*.
- The **Request for** individual is `current.cart.requested_for`.

Example of Filter Names

Following is an example script of a function. The function defines an advanced reference qualifier for a user (sys_user) reference variable that filters the available user records to those where **Name** contains the value of a variable named **textmatch** on the same item:

```
function getBlackberryUsers() {
  var answer = '';
  var includes = current.variables.textmatch;
  var usr = new GlideRecord('sys_user');
  usr.addQuery('name','CONTAINS',includes);
  usr.query();
  while (usr.next()) {
    if (answer.length > 0) {
      answer += (',' + usr.sys_id);
      }
    else {
      answer = '' + usr.sys_id;
      }
    }
  return 'sys_idIN' + answer;
  }
```

Service Catalog Variable Pricing

Overview

The price or recurring price of a catalog item can be modified with variables, increasing the base price for that item. This is useful if you want the price or recurring price dynamically calculated based on how a user completes the ordering form. For example, the cost of an item can be increased based on options the user selects, such as extra memory in a PC.

The recurring price feature is available in the Calgary release.

Variable Types That Can Affect Pricing

The following variable types can be configured to modify the total cost of an item:

- Check box
- Multiple choice
- Reference
- List collector
- Select box
- Lookup select box

Check Box

A check box variable can modify the price or recurring price of the item being ordered. Use the **Price if checked** or **Recurring price if checked** field to specify the price or recurring price difference for that variable when the option is selected.

Multiple Choice

For a multiple choice variable, there are two options:

- Specify the price difference in the **Price** field on the question choices.
- Specify the recurring price difference in the **Recurring Price** field on the question choices.

When a selection is made, the total ordering price or recurring price is dynamically updated to reflect the choice that was selected.

Select Box

Operates the same as a multiple choice variable, but is displayed as a select box.

Reference

A reference variable includes a **Pricing implications** field. If this field is selected and there is a **price**, **u_price**, **recurring_price** or **u_recurring_price** field on the referenced table, the value of that field is used to modify the cost of the item being ordered when a reference value is selected.

For example, suppose that computers in Phoenix cost $100 more than the ordering price and computers in Boise cost $100 less than ordering price.

1. Set up a reference variable to the Location [cmn_location] table.
2. Put a **u_price** field on the Location table.

3. Set that field to 100 for the Phoenix location.
4. Set that field to -100 for the Boise location.

When ordering a computer, if Phoenix is selected as the location for this variable, the ordering price is increased by $100. If Boise is selected, the ordering price is decreased by 100.

List Collector

Operates the same as a reference variable, but uses the **List table** field to specify the table being referenced. Since it is a list collector variable; multiple selections can be made that all modify the ordering price or recurring price.

Lookup Select Box

A lookup select box variable is more powerful than a reference variable. Specify the following when creating a lookup select box variable:

- **Lookup from table:** the table from which values are queried.
- **Lookup value field:** the field on the lookup table whose value is used as the variable's value (typically sys_id).
- **Lookup label field(s):** a comma-separated list of fields on the lookup table whose values are used to display the selections in the select box. For example, **manufacturer,name** for the Software [cmdb_ci_spkg] table would display selections as:
  - **Microsoft | Excel**
  - **Adobe | Photoshop Elements**
- **Lookup price field:** the field whose value is used to modify the price of the item being ordered.
- **Lookup recurring price field:** the field whose value is used to modify the recurring price of the item being ordered.

Extending the Pricing Model

The service catalog pricing model can be extended. The calculations to determine catalog item price or recurring price are done in the *CatalogPriceCalculator* or *CatalogRecurringPriceCalculator* script include. From the left navigation pane, select **System Definition > Script Includes**.

Auditable Service Catalog Variables

Overview

All changes to *named* service catalog variables are recorded in the audit history. Auditing is enabled by default for service catalog variables.

These changes are displayed in:

- The requested item history for all variables associated with the item.
- The catalog task history for all variables that would normally be used by the task.

This auditing allows service catalog administrators to view a full history of changes to variables that may have affected the parent record record, including their creation.

Note: *Unnamed variables are not audited*

Enabling Auditing

To enable auditing:

1. Navigate to **System Properties > Service Catalog**.
2. Locate the line **Audit changes to Service Catalog Variables**.
3. Select the **Yes** check box to enable auditing.

Viewing Audit History

To view the audit history of a variable:

1. Navigate to **Self-Service > Requested Items**.
2. Open a requested item to display the Requested Item form.
3. For the purpose of demonstration, change the value of a service catalog variable inside the item.
4. Right-click the header bar and select **History > List**.
5. The Record History form contains a related list detailing all changes to the requested item, including all variables.

Note: *Audits of variables are prefixed with "VARIABLE:".*

Scriptable Service Catalog Variables

Overview

You can use scripting to reference any variables used by a request item from any table, referencing them as:

```
current.variables.<variable_name>
```

Where current refers to the current record, and <variable_name> is the name of your variable.

Note: *In order to reference a variable from JavaScript, it must have a name.*

Usage Examples

Print a Variable

```
var original = current.variables.original_number;
gs.print(original);
```

Set a Variable

```
current.variables.name = "Auto-Generated:" + current.variables.asset_tag;
```

Create an Inventory Item with fields set from variables

```
doCreation();
function doCreation() {
   var create = current.variables.create_item;
   if (create == 'true') {
       // we want to create an asset
       var computer = new GlideRecord('cmdb_ci_computer');
       computer.initialize();
       computer.asset_tag = current.variables.asset_tag;
       computer.serial_number = current.variables.serial_number;
       computer.name = current.variables.name;
       computer.manufacturer = current.variables.company;
       computer.insert();
   }
}
```

Notes and Limitations

1. You can only *set* a variable in a *before* business rule. Variables set in an *after* rule are not written to the database.
2. There is nothing in place to prevent namespace collision with variables. Creating two variables named `computer_speed` would result in only one of them showing up; the second one would overwrite the first one.
3. Date/time variables use the same time zone formatting and storage rules as all other dates in the system. They are stored internally in GMT, but translated into the user's local time zone and format for display.

Reporting on Service Catalog Variables

Overview

Administrators can create list reports that use service catalog variables and request items, or service catalog variables and tasks.

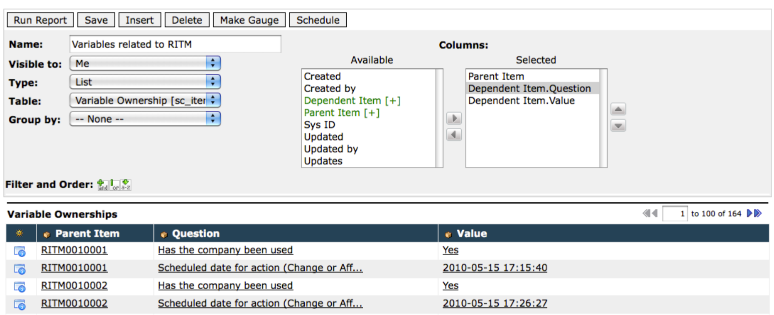
Creating a Report with Request Items

You can create a report with service catalog variables and request items using the Variable Ownership [sc_item_option_mtom] table.

List reports on this table use two main fields:

- **Parent Item:** references the request item
- **Dependent Item:** references the variables

To use variables in the report, dot-walk into the dependent item fields.



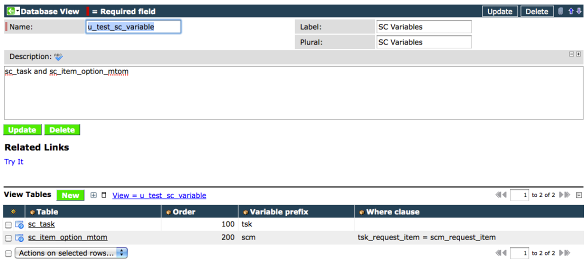
Variable Ownership list report

Creating a Report With Tasks

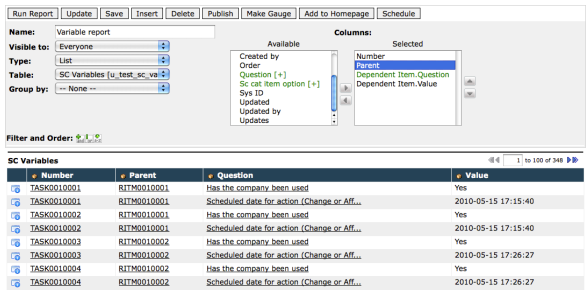
You can create a report with service catalog variables and the associated task records by creating a database view between the Catalog Task [sc_task] and Variable Ownership [sc_item_option_mtom] tables. (The Database Views plugin must be active to create these reports.) The elements available for reports on the database view are:

- **Parent Item:** references the request item
- **Task fields:** appear in the available columns
- **Dependent Item:** references the variables

To use variables in the report, dot-walk into the dependent item fields.



Create the database view



Database view list report

Controlling Page Layout Using Variables

Overview

Variables can control some aspects of layout for service catalog item pages. There are different options including one-column and two-column layouts.

Variable Set Layout Management

By default, all variables are laid out in a single vertical column, stacked from top to bottom. This is a simple way to display information, but can result in empty space to the right of questions.

Variables with several possible choices defined, such as multiple choice variables, are considered a single entity in layouts. All choices are displayed as a single, contiguous unit.

Variable sets can have one of three layouts.

- **1 column wide:** variables stack from top to bottom in a single column. This is the default layout.
- **2 columns wide, alternating sides:** variables are laid out in two columns with variables placed alternately in the left and right columns.
- **2 columns wide, one side, then the other:** the first half of the variable set is laid out sequentially in the left column and the second half is laid out sequentially in the right column.

When using a multiple column layout, consider editing question text so the questions are shorter and text does not need to wrap to a second line.

For example, create the following variables in order:

- Alfa 100
- Bravo 200
- Charlie 300
- Delta 400
- Echo 500
- Foxtrot 600

The possible layouts are:

1 Column Wide	2 Columns Wide, alternating sides	2 Columns Wide, one side, then the other

Alfa	Alfa	Bravo	Alfa	Delta
Bravo	Charlie	Delta	Bravo	Echo
Charlie	Echo	Foxtrot	Charlie	Foxtrot
Delta				
Echo				
Foxtrot				

To define a variable set layout:

1. Navigate to **Service Catalog > Catalog Variables > Variable Sets**.
2. Click a variable set **Name**.
3. In **Layout**, select an option.

Using Container Variables

For additional layout options, add container variables. For **Container Start**, select a layout, such as **1 Column Wide**. **Container End** simply closes a container. For more information, see Container Start and Container End.

You can also add container variables to variable sets.

Service Catalog Configuration

Service Catalog Access Controls

Overview

The service catalog supports several ways to control access to a catalog item or category (called catalog entitlements):

- Role
- Company, Department, Group, User, or Location
- Custom Script

A service catalog item with no specific access rules is available to all users. If access conditions are specified, only users who meet all conditions have access.

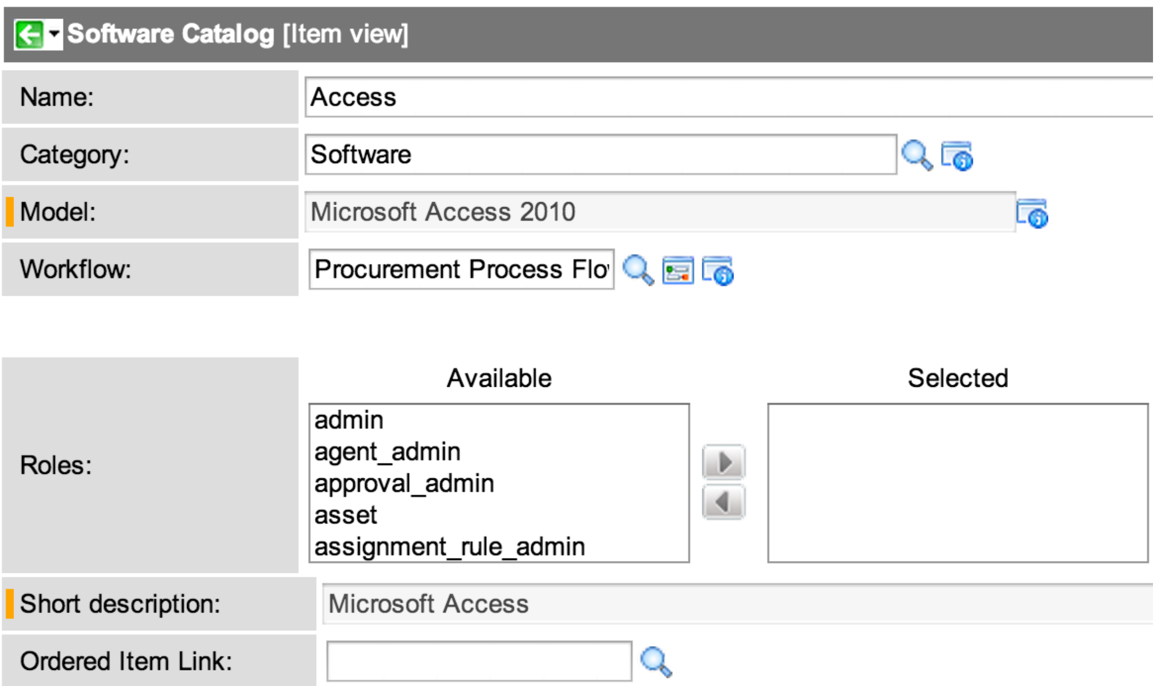
Access rules are checked in the following order: roles, scripts, access grants, and denials. If any conditions deny the user access, then the user does not have access.

Controlling Access by Role

By default, individual catalog items and categories do not have access restrictions. Administrators can grant or deny access to a service catalog item or category based on role.

1. Navigate to **Service Catalog > Maintain Items** or **Service Catalog > Maintain Categories**.
2. Open the relevant catalog item or category.
3. Add the required roles to the **Roles** field.

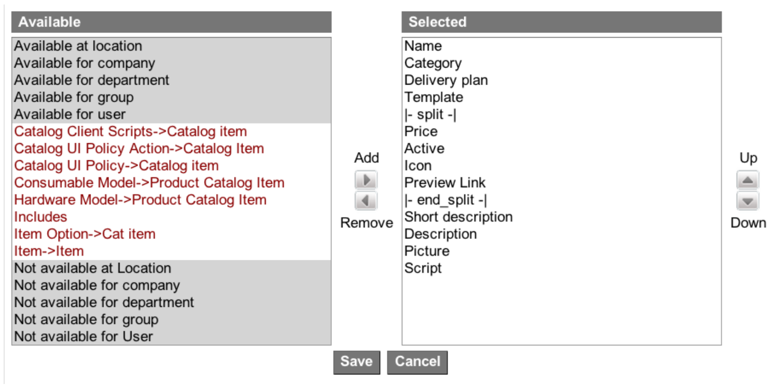
   You may need to personalize the form or change to **Default View** to see the **Roles** field.



Controlling Access by Company, Department, Group, User, or Location

To grant or deny access to a service catalog item or category by company, department, group, user, or location:

1. Navigate to **Service Catalog > Maintain Items** or **Service Catalog > Maintain Categories**.
2. Open the relevant catalog item or category.
3. Personalize the form to add the appropriate **Available** or **Not available** lists.



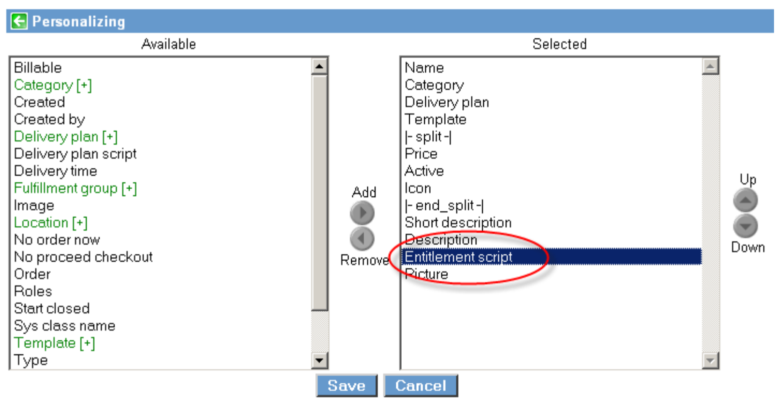
Personalize the related lists

4. Add the companies, departments, groups, users, or locations to the appropriate list.
5. Click **Update**.

Controlling Access with a Script

To control access to a service catalog item or category with a custom script:

1. Navigate to **Service Catalog > Maintain Items** or **Service Catalog > Maintain Categories**.
2. Open the relevant catalog item or category.
3. Personalize the form to add the **Entitlement Script** field.



Personalize the form

4. In the **Entitlement Script** field, enter the access control script.

Sample Script

The following example script grants access to a catalog item named *French Blackberry* to users with a language of **Fr** (French):

```
gs.log('Running Entitlement script for French Blackberry');

if(gs.getUser().getLanguage() == 'fr')
    true;
else
    false;
```

The following example script could be used to distinguish between two categories of users (one with full access and one with restricted access) on the catalog of services:

```
var userid = gs.getUserID();
var gr = new GlideRecord('sys_user');
```

```
gr.get('sys_id', userid);

if(gr.source)
    true;
else
    false;
```

Overriding Entitlement Scripts by Role

1. Navigate to **Service Catalog > Properties**.
2. Enter the roles for which to override the entitlement script (grant access) in the property **List of roles (comma-separated) that can override normal entitlement checking inside the catalog. A role of "itil" means that the itil role can order any catalog item, even one protected by entitlement restrictions**.

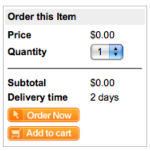
Hiding Prices

Overview

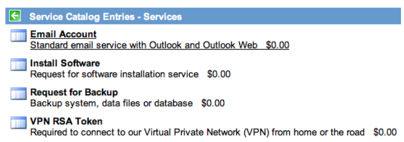
Administrators can configure options to hide item prices in the service catalog. Item prices may appear in:

- A catalog item listing in the **Order This Item** panel
- A subcategory listing
- The shopping cart panel
- The edit cart page
- The order confirmation page
- The order status or summary screen

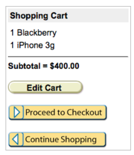
As illustrated below:



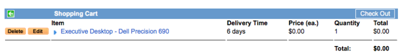
Order this item panel



Subcategory



Shopping cart panel



Edit cart

Order confirmation

Order status

Administrators can hide prices for specific items or for all items.

Hiding Prices for Specific Items

To hide prices for a specific item:

1. Navigate to **Service Catalog > Catalog Definition > Maintain Items**
2. Open the relevant item definition.
3. Select the **Omit Price on Cart** check box.

The price does not appear in the catalog listing or when the item is added to the cart. The price is listed as "-" on the edit cart, order confirmation, and order status screens.

Hiding Prices for All Items

To hide prices for all items:

1. Navigate to **Service Catalog > Catalog Policy > Properties**.
2. Select an option for **When to show prices and sub-totals on the Service Catalog Cart**.
   - **Show only non-zero prices:** (default) if an item price is zero, the price does not appear in the catalog listing or when the item is added to the cart. The price is listed as zero on the edit cart, order confirmation, and order status screens.
   - **Always show prices:** prices are displayed for all items on all service catalog screens.
   - **Never show prices:** prices are omitted for all items on all service catalog screens.

Localizing Prices

Overview

The ServiceNow service catalog fully supports localized currencies for item prices and options.

Note: *ServiceNow has the concept of a* system basis currency*: the system default currency, based on the global locale property. All currency values are automatically converted to this basis currency before aggregation or conversion. Do not change the global locale setting after you have data in the system, as this can cause aggregations to calculate and display incorrectly under some circumstances.*

Price and Currency

ServiceNow supports two similar but different **money** field types, both of which are used in the service catalog.

Price

A price represents the cost of the catalog item. A computer might cost $1000, or the provisioning of an email account may require an $80 chargeback.

ServiceNow supports these pricing models:

- **Calculated:** [Default] the price of each item is always quoted in the user's session language. For example, a UK user sees the price of her computer in pounds, while a Japanese user sees his price in yen. The numeric value of the price is adjusted using the most current conversion rates available.
- **Fixed:** the price of an item is always quoted in a particular currency. An item priced at $1,000 is always priced at $1,000, even if viewed by a UK or Japanese user.

- **Multiple:** a fixed price is specified for each currency. For example, you can specify that all US users pay USD 1000 for a computer, while the Japanese price is JPY 120,000, regardless of exchange rate.

Note: *If no fixed price is set, ServiceNow uses the calculated price (default).*

Currency

A currency field represents an actual *spend*, that is, a discrete quantity of money spent at a particular point in time. ServiceNow keeps track of the currency that was actually spent, and the time it was spent.

On a related list, a currency field uses one of these modes. Toggle between them by clicking the globe icon next to the currency field.

- **As spent:** displays the amount in the currency that was spent. This may mean that some items in the list show yen, while others show pounds or dollars, for example, £100.00.
- **Calculated:** displays all amounts in the session display currency, for example, $137.86.
- **Reference:** displays all amounts in the currency entered and includes the system basis currency in square brackets, for example, £100.00 [$137.86].

On a form, a currency field always displays the amount in the currency entered.

Note: *If you reference a currency field in script, its reported value is whatever the current user's session would show. There are also several utility functions you can call against a currency field to get data back in a variety of forms. For more information, see Scripting Currency and Price Fields.*

Displaying Currencies to Catalog Users

You can specify that an item and its options use any of the pricing models described above. For example, set a fixed price for the default iPhone item in pounds (£), and it always appears in pounds (£), even for a US-based user. The same is true for variables with pricing implications. If you specify a fixed price on the options, then ServiceNow displays the option in that currency on the form for ordering the item.

However, prices of items displayed to end users in the shopping cart and checkout screens are always shown in the logged-in user's session currency. For example, if a US-based end user adds a £100.00 item to his cart, the shopping cart shows the equivalent value in US dollars.

Localizing Prices and Currency

Setting Prices

The **Price** field on catalog items uses the *price* data type, meaning that catalog items and lists of catalog items show localized prices. By default, price fields use the *calculated* pricing model, meaning that regardless of the currency used to enter the price, ServiceNow converts that value to the user's session currency and displays that converted value to the user.

When defining catalog items, catalog administrators can specify a currency and a value for the item, and can alter the pricing model by clicking the **Edit** link beside the **Price** field on the Catalog Item form.

Using Locales

ServiceNow uses the language and country specified in the user record to determine the currency and pricing model for a particular user. If these records are incomplete or incorrect, currency and pricing models are based on the default locale. For more information, see Defining Locales.

Setting Conversion Rates

ServiceNow downloads a currency conversion table from the European Central Bank nightly by default. You can adjust the frequency of this behavior or disable it entirely.

1. Navigate to **System Scheduler > Scheduled Jobs**.
2. Open the job named **ECB Exchange Rate Load**.
3. Modify the schedule, as needed.

After the job runs, rates are stored in and loaded from the Exchange Rate [fx_rate] table. Navigate to **System Localization > Exchange Rates** to see them.

Using Your Own Conversion Table

ServiceNow bases all currency conversions on the rates stored in the Exchange Rate table. To load your own conversion rates rather than the automatically downloaded rates:

1. Navigate to **System Scheduler > Scheduled Jobs**.
2. Open the job named **ECB Exchange Rate Load**.
3. In the **Trigger type** field, select **-- None --**.
4. Enter new exchange rates either manually or with an import set.

Reporting

Currency and price fields can be queried and reported. Reports and aggregations are always display values in the session currency.

Default Reports

ServiceNow provides two reports that address service catalog spending:

- **This Year's Spending by Department:** spending for the previous 12 months, broken down by month and department.
- **This Year's Spending by Location:** spending for the previous 12 months, broken down by month and requester location.

To run these reports, navigate to **Reports > View / Run** and go to the **Requested Item** section.

Creating Custom Reports

When creating custom reports, please consider the following:

- Reports display currency values converted to the currency of the user running the report. A shared report displays in Euros if run by a German, but in US dollars if run by an American. Currency conversions do apply.
- Scheduled reports generally run as the user who scheduled them. So a report scheduled by an American and emailed to three Europeans shows values in US dollars.

UI Policy

Overview

UI policies can be used to define custom process flows for tasks. UI policies are also useful when applied to service catalog items.

Keep the following points in mind when creating service catalog UI policies:

- A catalog UI policy applies to either a specific catalog item or any item that uses a specific variable set.
- A catalog UI policy can only contain and reference variables that are defined for the catalog item or variable set.
- Exactly like UI policy conditions, the variables in a service catalog UI policy condition must be visible (even if hidden by UI policy or read-only) on the form for the condition to be tested.
- Service catalog UI policies are applied to variables and variable sets of catalog items ordered in the service catalog. Policies can also be applied when the variables are present in a Requested Item or Catalog Task form (Calgary release).
- Catalog UI policies are supported for catalog items viewed in a service catalog wizard (Calgary release).

Creating a UI Policy for Service Catalog Items

1. Navigate to **Service Catalog > Catalog Policy > Catalog UI Policies**.
2. Click **New**.
3. In the **Applies to** field, select **A Catalog Item**.
4. Fill in the remaining fields (see table).
5. Click **Submit**.

The following fields are available on the Catalog UI Policy form.

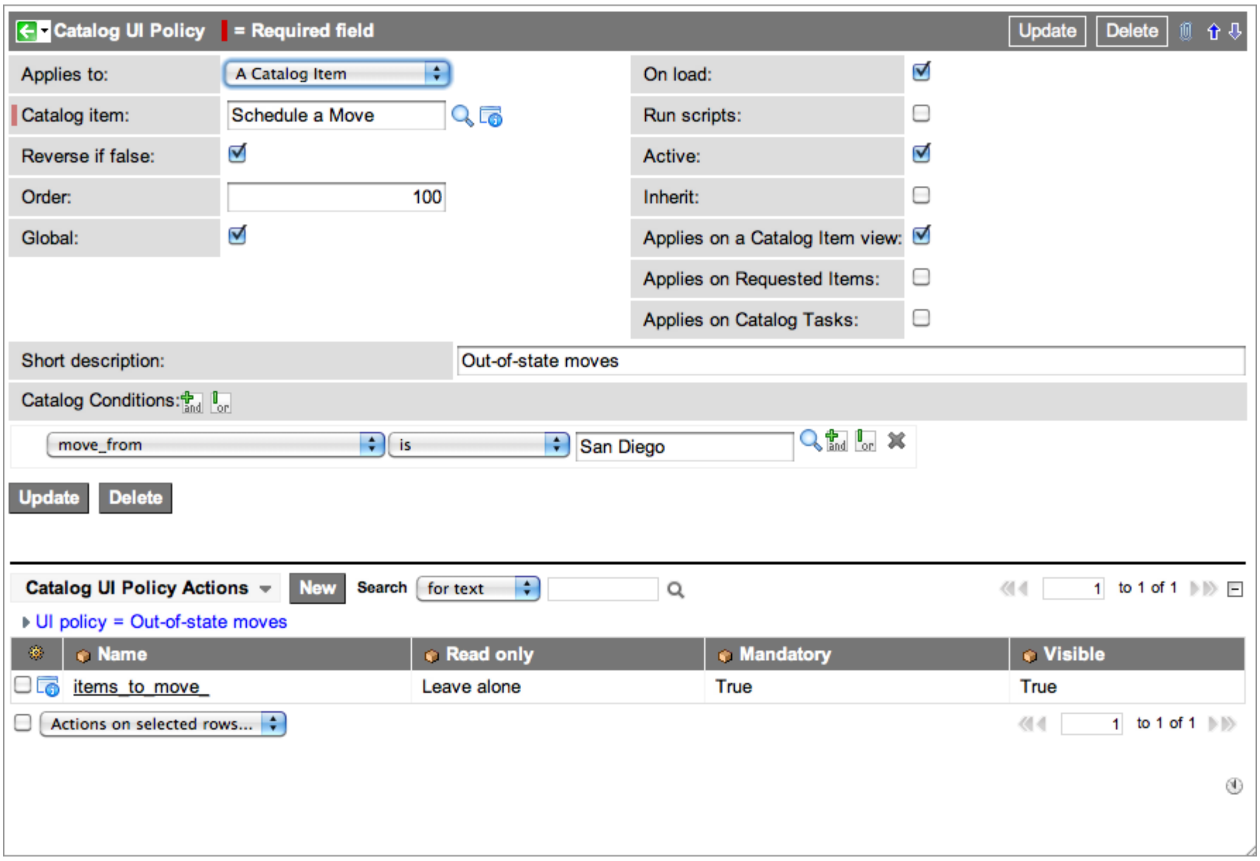
Field	Description
Applies to	Select the type of item this UI policy applies to: • **A Catalog Item:** enables the **Catalog item** field. • **A Variable Set:** enables the **Variable set** field.
Catalog Item or Variable Set	Select the catalog item or a variable set this UI policy applies to. The field name and options available depend on the **Applies to** selection.
Reverse if false	Select the check box to reverse the UI policy if the **Catalog Conditions** statement evaluates to *false*.
Order	Enter the sequence in which this UI policy is evaluated if more than one matching UI policy exists. The order is evaluated from the lowest value to the highest value.
Global	This field is not used for the service catalog.
On load	Select the check box to apply the UI policy when the form is loaded. Clear the check box to apply the policy only when the form is changed.
Run scripts	Select the check box to use the **Execute if true** and **Execute if false** scripting fields. Scripts are necessary to apply a UI policy other than **Read Only**, **Mandatory**, or **Visible**. For example, you must create a script to apply a UI policy to a specific role.
Active	Select the check box to enable the UI policy. Clear the check box to disable it.
Inherit	This field is not used for the service catalog.
Applies on a Catalog Item view	Select the check box to apply the UI policy to catalog items within the order screen. (This option is available in the Calgary release.)

Applies on Requested Items	Select the check box to apply the UI policy on a Requested Item form. (This option is available in the Calgary release.) **Note:** This is not supported if the Requested Item record is created via a record producer. ServiceNow does not recommend creating Requested Item records from record producers.
Applies on Catalog Tasks	Select the check box to apply the UI policy on a Catalog Task form. (This option is available in the Calgary release.)
Short description	Enter a brief description (example, *Out-of-state move*).
Catalog Conditions	Create conditions for the UI policy using catalog item variables. The policy is applied if the conditions evaluate to *true*. For example, a catalog item of **Schedule a Move** may have a condition of **move_from > is > San Diego**. The UI policy is applied when an employee schedules a move from San Diego to any other company location.

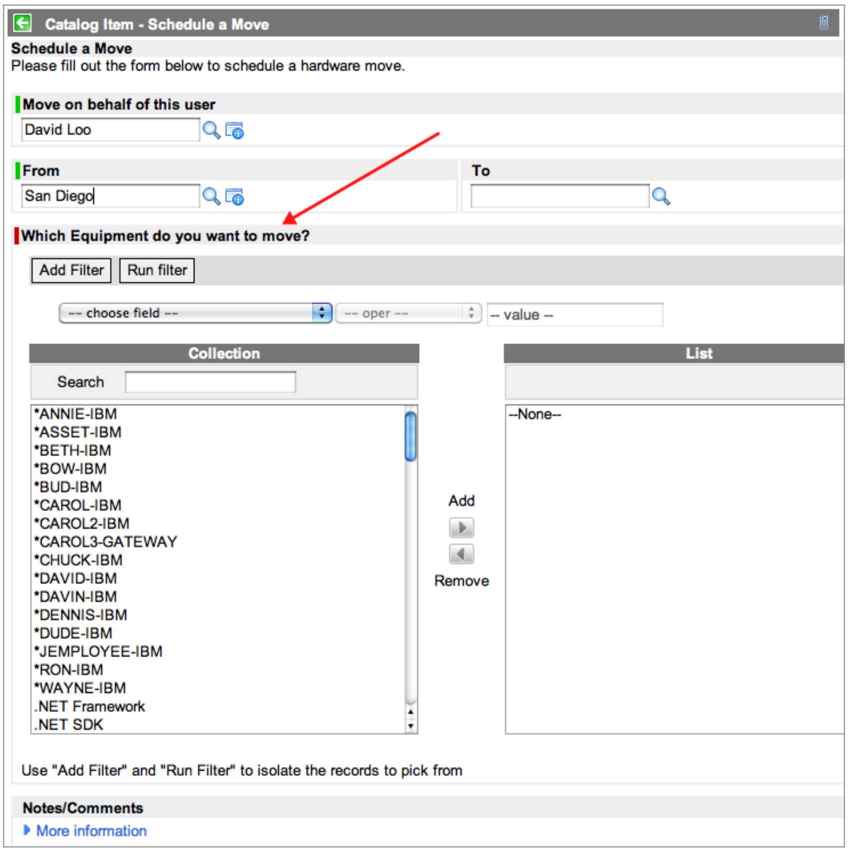
Example Service Catalog UI Policies

Example 1

The following UI policy displays a mandatory field of all items to be moved when a user schedules a move from San Diego to any other office location.

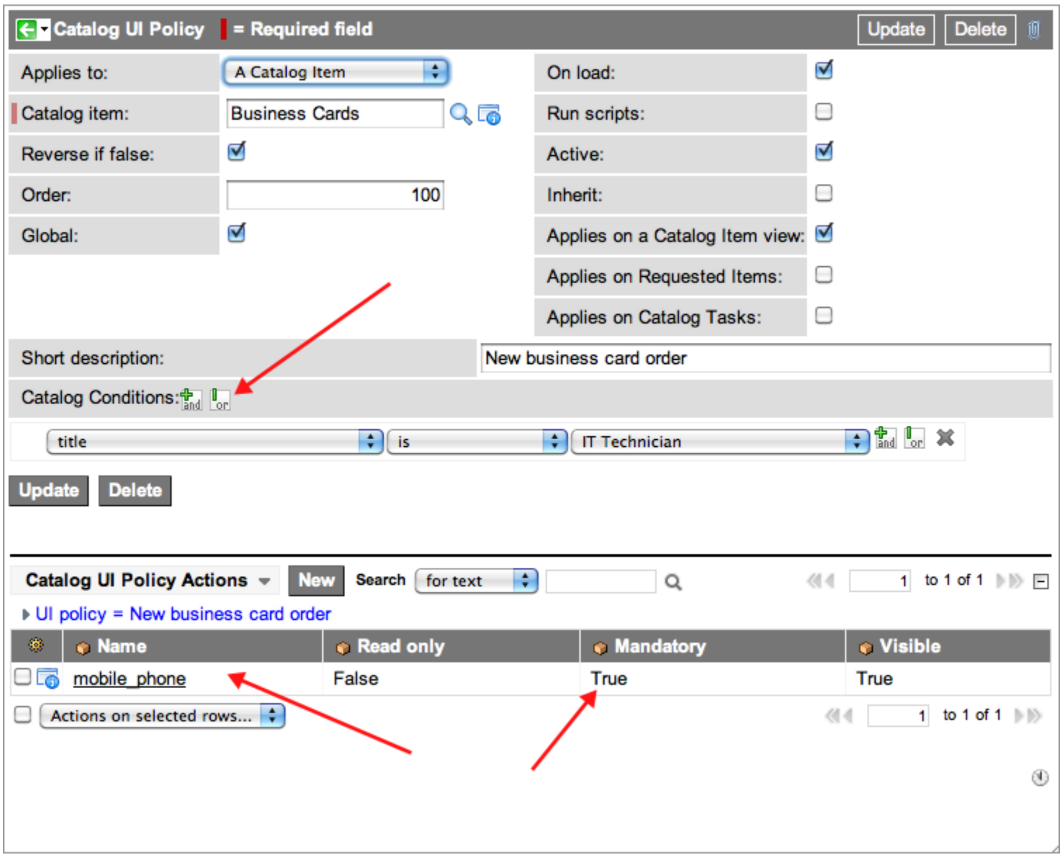


When a user schedules their move and selects **San Diego** in the **From** field, a mandatory field appears and asks the question, *Which equipment do you want to move?*



Example 2

The following UI policy makes it mandatory for anyone with the title **IT Technician** to add a mobile phone number when ordering business cards.



Using Renderers

Overview

Administrators and catalog administrators can use and modify renderers to control how categories are displayed on the service catalog homepage.

Two default renderers are provided with the base system:

- **Category Only:** shows only the category homepage image and description.
- **Default:** shows a list of catalog items for that category.

Renderers use UI Macros to provide the rendering instructions. Administrators can create UI Macros to implement new renderers: for example, a UI macro for use with a renderer which shows the category homepage image, the description and the first two catalog items in a category.

This feature is available with the Calgary release.

Selecting a Renderer for a Category

To select a renderer for a category:

1. Navigate to **Service Catalog > Catalog Definition > Maintain Categories**.
2. Open the desired category.
3. In the **Homepage renderer** field, select the renderer to use with this category.

4. Select a renderer to use with that category:

This renderer is then applied to that category on the service catalog homepage.

Creating or Modifying a Renderer

To create custom renderers to define a specific look and feel for associated categories:

1. [Optional] Create a UI macro to define specific rendering instructions if required.

   Note: only administrators can create or modify UI macros.
2. Navigate to **Service Catalog > Catalog Definition > Renderers**.
3. Click **New** and select the **Homepage Category Renderer** type.

4. Enter a **Name** for the renderer.
5. In the **Macro** field, select the UI macro to use.
6. Select the **Render category title** check box to display the category title bar on the homepage.
7. Click **Submit**.

   This new renderer is added to the list of available choices when defining a category.

To modify one of the default renderers, follow a similar process, selecting one of the default renderers listed.

Data Lookup

Overview

The Data Lookup and Record Matching Support for Service Catalog plugin offers similar features to the general Data Lookup and Record Matching Support plugin. Use the plugin for service catalog to perform data lookups for variables on service catalog item screens, on requested items, and on catalog tasks as a user fills out the values contained in variables. The plugin for service catalog is available in the Calgary release.

Administrators and users with the catalog_lookup_admin and catalog_lookup_manager roles can create and use service catalog data lookups.

Activating the Plugin

This plugin is automatically active on new instances running Calgary or later.

Customers who upgrade from a previous version can manually activate the Data Lookup and Record Matching Support for Service Catalog plugin. Activating the plugin also activates the Data Lookup and Record Matching Support plugin. For information about the Data Lookup and Record Matching Support plugin and implications for existing priority lookup rules when activating, see Data Lookup and Record Matching Support

Click the plus to expand instructions for activating a plugin.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.

   If the plugin depends on other plugins, these plugins and their activation status are listed.
3. [Optional] Select the **Load demo data** check box.

   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when first activating the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

Roles

The following roles participate in the process of creating and using service catalog data lookups:

Role	Description
catalog_lookup_admin	Similar to data_lookup_admin. Can create, update, and delete catalog data lookup definitions, matcher variable definitions, and setter variable definitions.
catalog_lookup_manager	Can read catalog data lookup definitions, matcher variable definitions, and setter variable definitions. The role can be granted to anyone using catalog data lookups so they can see the definitions for which they are creating rules. As required, grant create, read, write, or delete access to the individual data lookup rules tables created to delegate maintenance.

Creating Custom Catalog Data Lookups

Creating a new catalog data lookup is similar to creating a normal, custom data lookup except when creating the catalog data lookup definition record.

1. Create a custom data lookup table. It must extend the Data Lookup Matcher Rules [dl_matcher] table.
2. Add data lookup values to the data lookup table.
3. Create a catalog data lookup definition record.
4. (Optional) Create a data lookup module.

Step 1. Create a Custom Data Lookup Table

Create a custom table to store lookup data. The custom table must extend the Data Lookup Matcher Rules [dl_matcher] table.

For example, you can create a Server Offering Lookups table to store information about matcher offerings (bronze, silver, and gold) and associated setter values (memory and disk space) for each matcher offering.

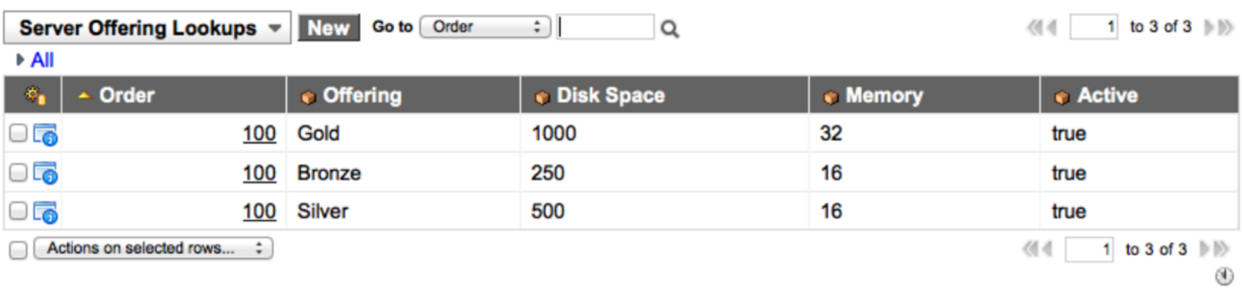
Step 2. Add Data Lookup Values to the Data Lookup Table

The columns of a data lookup table contain both matcher and setter field data. Each data lookup is a query that searches for a row containing values that match the matcher fields. The data lookup then returns the value listed in the setter fields.

For example, you can define the matching settings for bronze, silver, and gold offerings as described previously.

To add values to the lookup table:

1. In the navigation filter, enter the name of the new custom lookup table.
2. Personalize the list and create appropriate fields for the lookup table.
3. From the table list, click **New** and enter appropriate matcher and setter field values. For example:



Note: *Each row in a data lookup table must be unique.*

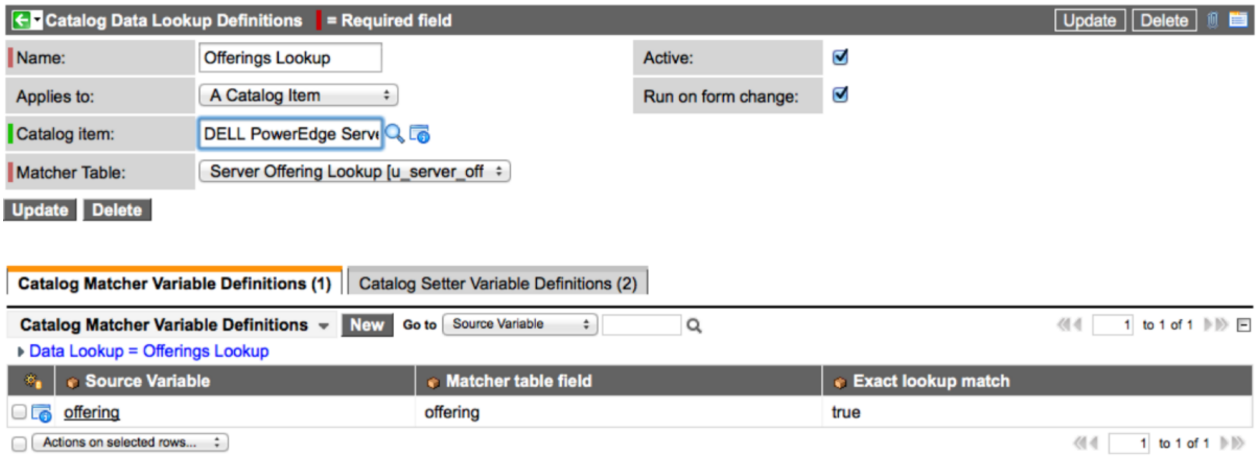
Step 3. Create a Catalog Data Lookup Definition Record

Note: **The **Run on Insert** and **Run on Update** options are not supported for catalog data lookups. Other options operate the same as for normal data lookup.*

To create a catalog data lookup definition record:

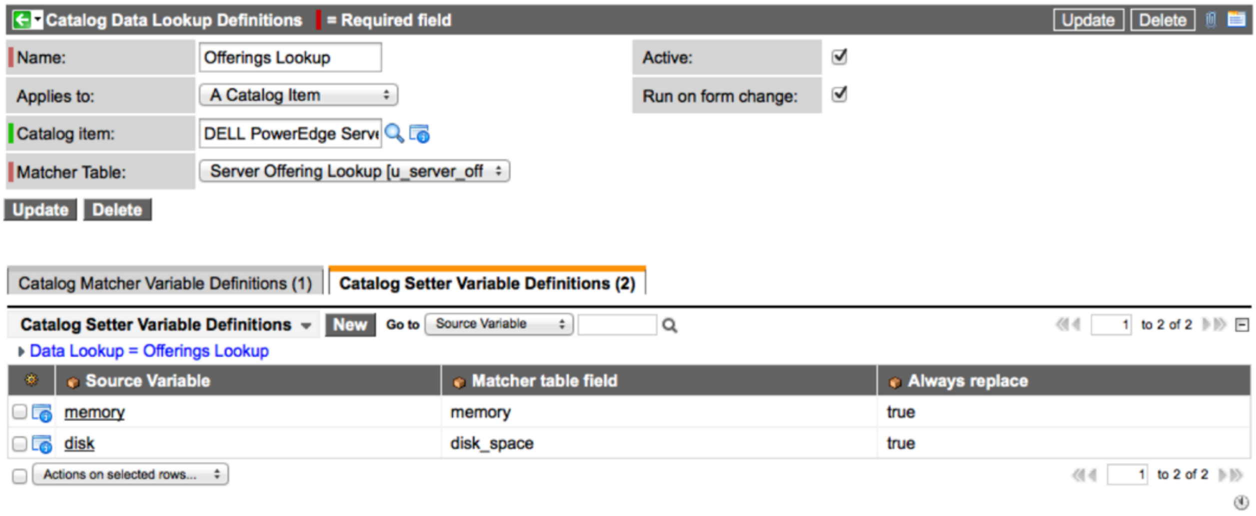
1. Navigate to **System Policy > Rules > Data Lookup Definitions**.
2. Click **New**.
3. Select **Catalog Data Lookup Rule**.
4. Enter a **Name**.
5. In **Applies to**, select catalog item or variable set.
6. In **Catalog item/Variable set**, select a specific item or set.
7. Select a **Matcher Table**.
8. Select other options, as required.
9. Right-click the form header and click **Save**.
10. From the **Catalog Matcher Variable Definitions** related list, click **New**.
11. In **Source Variable name**, select the variable name of the item or variable set that contains the values to be matched.
12. In **Matcher table field**, select the field from the matcher table that contains the value to be matched.
13. Fill in the other fields, as appropriate.

    For example:



14. Click **Submit**.
15. From the **Catalog Setter Variable Definitions** related list, click **New**.
16. In **Source Variable name**, select the variable name for the item or variable set to be updated.
17. In **Matcher table field**, select the field from the matcher table that contains the value to be set.
18. Fill in the other fields, as appropriate.
19. Click **Submit**.
20. Click **Update**.

    For example:



Catalog Data Lookup Definitions Fields

Field	Description
Name	Enter a unique name to identify the definition record.
Applies to	Select what the data lookup rule should apply to: a catalog item or a variable set.
Catalog item/Variable Set	Select the catalog item or variable set to which the data lookup rule should apply.
Matcher Table	Select the table containing the lookup values. This table normally begins with a **u_** prefix.
Active	Select this check box to run this catalog data lookup rule. Clear the check box to ignore this catalog data lookup rule.
Run on form change	Select this check box to automatically look up values whenever a user changes a variable value on a catalog item or form. This is the only supported method for catalog data lookup rules. **Note:** This does not include changes automatically made by other data lookup rules, such as the priority lookup rules.

Matcher Variable Definition Fields

The matcher variable definitions determine when a data lookup occurs. A data lookup only occurs on a variable with matcher variable definitions. The data lookup uses the values of the source variable to look up one or more values from the matcher table.

Field	Description
Data Lookup	Displays the name of the parent data lookup definitions record.
Source variable name	Select the variable from the source item or variable set that contains the data to match.
Matcher table field	Select the field from the matcher table that contains the data to match.
Exact lookup match	Select this check box to require the matcher table to contain a matching row for every possible combination of values (including blank values). Clearing this check box means that any blank values in the matcher table match any value. For example, suppose the **Offering** field is blank in the matcher table. When this check box is selected, there is a match only when the value in the source table row is blank. When this check box is cleared, the blank matcher field value matches any value in the source table field. **Note:** If the lookup does not require an exact match, matcher table rows containing blank values are treated as wild cards, matching all values.

Setter Variable Definition Fields

The setter fields determine what variable the data lookup changes when the matching conditions are true.

Field	Description
Data Lookup	Displays the name of the parent data lookup definitions record.
Source variable field	Select the variable from the source item or variable set that the data lookup updates.
Matcher table field	Select the field from the matcher table that provides the new value for the update.
Always replace	Select this check box to replace any existing value with a value from the data lookup. Clear this check box to ignore the update if the field has an existing value.

Step 4. Create a Data Lookup Module

Optionally, you can create a new module for the data lookup table you created in Step 1. Creating a new module enables you to customize the application navigator and organize information by user role. The module should have these properties:

Property	Required Value
Table	Select the data lookup table you created in Step 1. For example **u_server_offerings_lookup**.
Link type	List of Records

Troubleshooting

If the custom data lookup definition rules are not behaving as expected, check for the following conditions.

- Ensure that the matcher variable is not read-only. Since users cannot change read-only variables, there cannot be an on form change event for read-only variables.
- Ensure that the data in the matcher table is correct.
- If the lookup requires an exact match, verify that there is a matcher table row for each possible combination (including blank values). The lookup fails if cannot find a matching value.
- If the variable is an option or check box, it always has a value, so you must select **Always replace**.
- Verify that you have not created a recursive rule, such as:

  If Variable A = 1, then Variable B =2. If Variable B = 2, then Variable A = 2.

Service Catalog Properties

Overview

To configure service catalog system properties, navigate to **Service Catalog > Catalog Policy > Properties**.

Note: *See Available System Properties for a list of general system properties.*

Service Catalog Properties

The following service catalog properties are available with all releases. Further properties are available starting with the Calgary release, and starting with the Dublin release

Name	Description
glide.approval_engine.sc_request	Service Catalog Requests approval engine. • **Learn more:** Approval Engines
glide.approval_engine.sc_task	Service Catalog Tasks approval engine. • **Learn more:** Approval Engines
glide.sc.allow.checkout.clone	Enable cloning requests during checkout. • **Learn more:** Enabling Bulk Requests
glide.sc.allow.clone.roles	List of roles (comma-separated) that can use bulk ordering functionality. Blank means all users. • **Learn more:** Enabling Bulk Requests
glide.sc.allow.quantity	List of roles (comma-separated) that can use the quantity selector in the shopping cart. Blank means all users. • **Learn more:** Defining Catalog Item Quantities
glide.sc.approval.hover	Show the current pending approver's name in the stage widget mouseover.
glide.sc.audit variables	Audit changes to Service Catalog variables. • **Learn more:** Auditable Service Catalog Variables
glide.sc.can_search	List of roles (comma-separated) that can search the Service Catalog. Blank means all users. • **Learn more:** Base System Roles
glide.sc.category.canview.override	List of roles (comma-separated) that override entitlements so that they can view any category within the Service Catalog. • **Default value:** admin and catalog_admin • **Learn more:** Service Catalog Access Controls
glide.sc.checkout.cancel	Allow ess users the option to cancel their requests from the checkout screen. • **Learn more:** Enabling a Two-Step Checkout
glide.sc.checkout.cancel.condition	Condition under which users may cancel a request (previous property must also be true). Requires **Allow ess users the option to cancel their requests from the checkout screen** to be *True*.
glide.sc.checkout.request.number	Show the request item number for each line item on the checkout screen. • **Default value:** false • **Learn more:** Enabling a Two-Step Checkout
glide.sc.checkout.twostep	Use the two step catalog checkout model. • **Default value:** false • **Learn more:** Enabling a Two-Step Checkout

glide.sc.checkout.twostep.back	Show the 'Back to Catalog' button on the two step checkout screen. • **Learn more:** Enabling a Two-Step Checkout
glide.expert.checkout.twostep	Use the two step checkout model when placing a Catalog Order from a Wizard • **Default value:** true • **Learn more:** Enabling a Two-Step Checkout
glide.sc.checkout.task.display	Show Tasks related to Requests on the Order Status page (the screen you see in the Service Catalog after a successful order is placed). • **Learn more:** Enabling a Two-Step Checkout
glide.sc.continue.shopping.target	URL to redirect to upon clicking of the "Continue Shopping" button inside of a cart. Blank means default behavior (go back one screen). Specifying a fixed URL such as "catalog_home.do?sysparm_view=catalog_default" will always return to the fixed URL. • **Learn more:** Service Catalog Order Guides
glide.sc.delivery_summary.name	Use delivery task name instead of short_description for the delivery plan summarizer field. • **Default value:** false (uses the short_description) • **Learn more:** Using Execution Plans
glide.sc.enhance.labels	Append pricing information to option labels. • **Learn more:** Service Catalog Variable Pricing
glide.sc.entitlement.override	List of roles (comma-separated) that can override normal entitlement checking inside the catalog. A role of "itil" means that the itil role can order any catalog item, even one protected by entitlement restrictions. • **Learn more:** Service Catalog Access Controls
glide.sc.ess.description	Field name to use for the description column of the checkout form. If blank, the default is used. • **Default value:** short_description
glide.sc.home.filter	List of content types (comma-separated) to allow on the catalog homepage. Blank allows all content types. • **Learn more:** Content Types
glide.sc.price.display	When to show prices and sub-totals on the Service Catalog Cart. • **Learn more:** Hiding Prices in the Service Catalog
glide.sc.req_for.roles	List of roles (comma-separated) that can view the "Requested for" widget in the Service Catalog. Blank means all users. Controls access to the **Requested For** widget on the catalog homepage. Users with access to this widget can request items for another person. Users without access can place orders in their own name, only.
glide.sc.reset_cascade	On an order guide, reset cascading variable values on an item when a user goes back using the "describe needs" button. • **Learn more:** Service Catalog Order Guides
glide.sc.restrict.quantity.changes	Prevent changes to requested item quantity when approved (except for catalog_admin users). • **Learn more:** Defining Catalog Item Quantities
glide.sc.round.delivery.times	Round all delivery plans > 1 day to the nearest day (true) or display the precise time (false).
glide.sc.search.rowcount	Number of Service Catalog matches returned for global searches. Limits the number of results returned by a catalog search to improve search performance.
glide.sc.variable.snapshot	Render variables on a request item as they appear on the order panel, followed by the delivery plan variables (true), or merge the two based on their order values (false).
glide.sc.request_for.columns	Additional columns for the "request for" Service Catalog widget. Choose fields in the sys_user table. Must be semicolon separated.
glide.sc.request_for.order_by	Ordering of matches for the "request for" Service Catalog widget. Choose fields in the sys_user table.

Starting With Calgary

The following service catalog properties are available starting with the Calgary release.

Name	Description
glide.sc.homepage.show.collapse	Toggle whether the expand/collapse icon is rendered for category widgets on the service catalog homepage. • **Type:** true | false • **Default value:** false • **Learn more:** Managing the Service Catalog Homepage
glide.sc.item.cannot_add_to_request	List of class names for catalog items that cannot be added to an existing request. • **Type:** string • **Default value:** sc_cat_item_guide, sc_cat_item_producer, sc_cat_item_wizard
glide.sc.item.cannot_try_it	List of class names for catalog items that do not use the default "Try It" UI Action. • **Type:** string • **Default value:** sc_cat_item_guide, sc_cat_item_producer, sc_cat_item_wizard, sc_cat_item_service
glide.sc.item.not_normal_cart_item	List of class names for catalog items that do not generate a normal cart item. • **Type:** string • **Default value:** sc_cat_item_guide, sc_cat_item_producer, sc_cat_item_wizard
glide.sc.item.cannot_show_price	List of class names for catalog items that do not show the price in listings. • **Type:** string • **Default value:** sc_cat_item_guide, sc_cat_item_producer, sc_cat_item_wizard
glide.sc.item.cannot_show_search	List of class names for catalog items that do not have the search field displayed. • **Type:** string • **Default value:** sc_cat_item_guide, sc_cat_item_producer, sc_cat_item_wizard
glide.sc.guide.tab.validate	Validate mandatory fields when switching tabs in 'Choose Options' section of Order Guides. • **Type:** true | false • **Default value:** true
glide.sc.max_items	Number of Catalog Items or Categories to preview in a section. • **Type:** integer • **Default value:** 5 • **Learn more:** Defining Catalog items and Service Catalog Categories
glide.sc.show_additional.cats	Show the additional categories section when viewing a catalog item. • **Type:** true | false • **Default value:** true • **Learn more:** Defining Catalog items and Service Catalog Categories
glide.sc.cat_view_use_popup_for_details	When browsing a category use the popup icon to show item details. • **Type:** true | false • **Default value:** false • **Learn more:** Service Catalog Categories
glide.sc.auto_expand	Number of Catalog Items to expand in browsing and search when not using popup icons to view details. • **Type:** integer • **Default value:** 2 • **Learn more:** Defining Catalog items
glide.sc.use_breadcrumb_links.cms	Use links for breadcrumbs rendered in Service Catalog pages accessed via a CMS site. This enables users with the CMS administrator (content_admin) role to choose whether service catalog breadcrumbs are displayed with or without links for greater navigation control. • **Type:** true | false • **Default value:** false

glide.sc.use_sub_cat_section	In category view display subcategories in a panel. • **Type:** true | false • **Default value:** true • **Learn more:** Service Catalog Categories
glide.sc.search.disabled_cats	Service catalog searches return items in inactive categories. Search results can include catalog items in non-accessible categories, as specified by the active flag or by security constraints. • **Type:** true | false • **Default value:** true
glide.sc.search.cms_page	Service catalog search page for CMS. This is the name of the CMS page or UI page for searching in CMS or ESS. The page is loaded in the current frame. • **Type:** string • **Default value:** catalog_find_cms (a UI page that calls the **catalog_find_macro** UI macro)
glide.sc.home.cms_page	Service catalog home page for CMS. This is the name of the CMS page or UI page that renders the service catalog homepage in CMS or ESS. The page is loaded in the top level. • **Type:** string • **Default value:** order_things (a CMS page that displays a splash menu of service catalog categories)

Starting with Dublin

The following service catalog properties are available starting with the Dublin release.

Name	Description
glide.sc.placeholder.image	Name of placeholder picture for items that do not have a picture defined. ServiceNow provides several possible images, from **sc_placeholder_image-01.png** to **sc_placeholder_image-14.png**. Also, the default can be replaced with a custom image. • **Default value:** sc_placeholder_image.png • **Learn more:** Storing Images in the Database
glide.sc.mobile.home.category.render	Specify how sub-categories will be rendered in the Mobile UI, prior to user selection of Card or List layout. Note: the service catalog homepage parent category layout cannot be modified from the default Card layout. • **Type:** choice list • **Default value:** card • **Learn more:** Creating and Configuring Categories
glide.sc.mobile.limit.description	Limit descriptions in category and item listings to two rows in the Mobile UI. • **Type:** true | false • **Default value:** true

Configuring the Checkout Process

Enabling a Two-Step Checkout

Overview

Administrators and users with the catalog_admin role can enable and configure the two-step checkout model.

The service catalog defaults to a one-step checkout model. When a user clicks **Proceed to checkout** or **Order now**, items in the shopping cart are ordered and the order summary or status screen appears. The one-step checkout model runs in the following order:

Press Checkout > Order Summary

The service catalog also supports a two-step checkout model. Under this model, when a user clicks **Proceed to checkout** or **Order now**, an order confirmation screen appears, allowing the user to edit the order, choose a delivery location, or upload an attachment before submitting the order. The two-step checkout model runs in the following order:

Press Checkout > Order Confirmation Screen > Submit Order > Order Summary

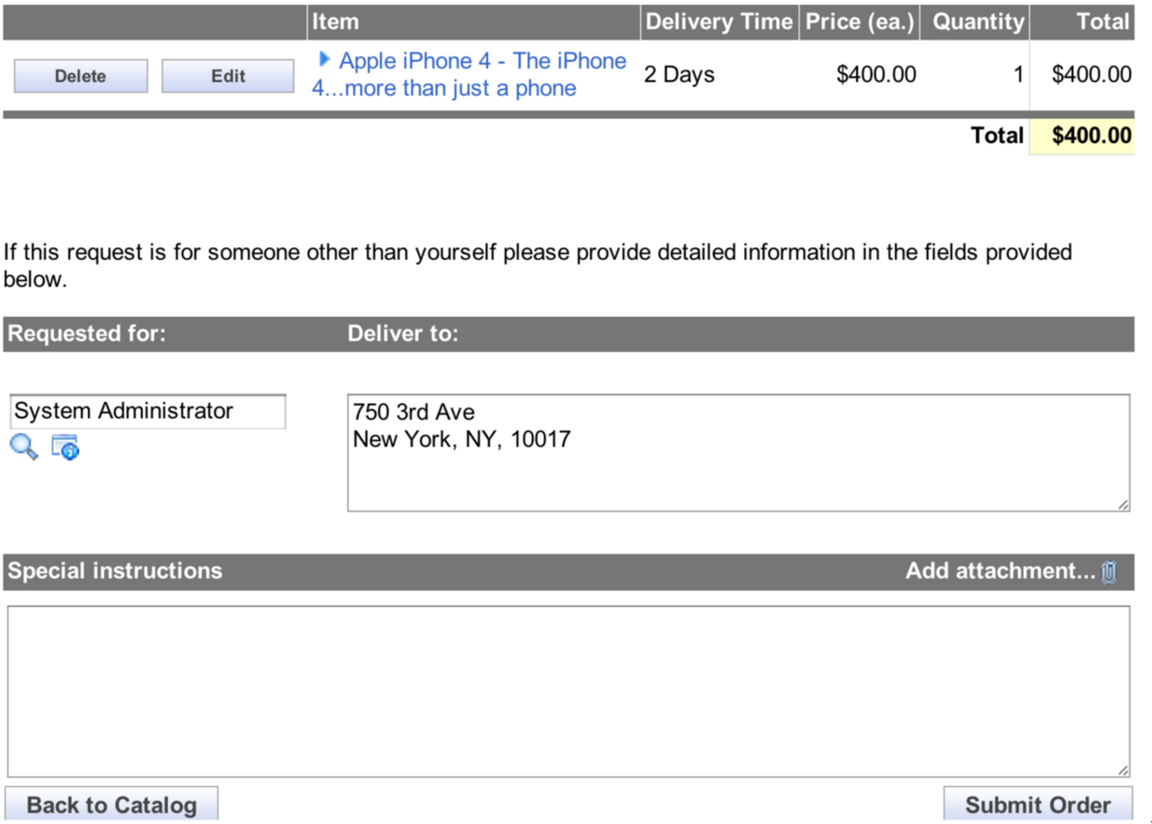
Enabling the Two-Step Checkout Process

To enable the two-step checkout process:

1. Navigate to **Service Catalog > Catalog Policy > Properties**.
2. Locate the property **Use the two step catalog checkout model (default false)**.
3. Select the **Yes** check box to enable the two-step process.
4. Locate the property **Show the 'Back to Catalog' button on the two step checkout screen**.
5. Select the **Yes** check box to provide a button that navigates back to the catalog from the order confirmation screen (default). Clear the check box to hide the button.

Controlling how the Requester's Location is Defined

In the two-step checkout model, end users may specify a recipient, delivery address, and special instructions for an order.



Administrators can control how the delivery address is populated. By default this is defined by the client script called *set location*.

When the two-step checkout process is enabled, the *set location* script retrieves the address of the user and enters formatted details in the **Deliver to** field.

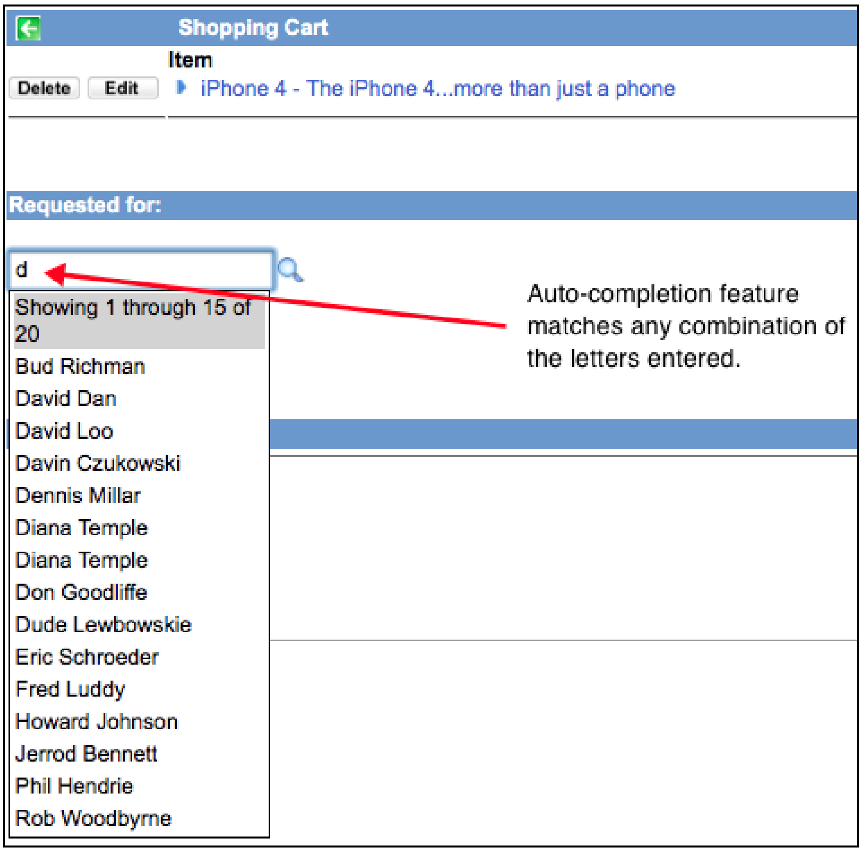
Requester Search Results

After you enable the two-step checkout process, the **Requested for** field appears on the Request form. The **Requested for** field references the User [sys_user] table and has an auto-complete feature. Two service catalog properties (**Service Catalog > Catalog Policy > Properties**) enable an administrator to add columns to the search results for this field and to order the list by one of the columns.

Property	Description
Additional columns for the "request for" Service Catalog widget. (`glide.sc.request_for.columns`)	Choose fields from the User [sys_user] table. Must be semicolon separated.
Ordering of matches for the "request for" Service Catalog widget. (`glide.sc.request_for.order_by`)	Choose fields from the User [sys_user] table.

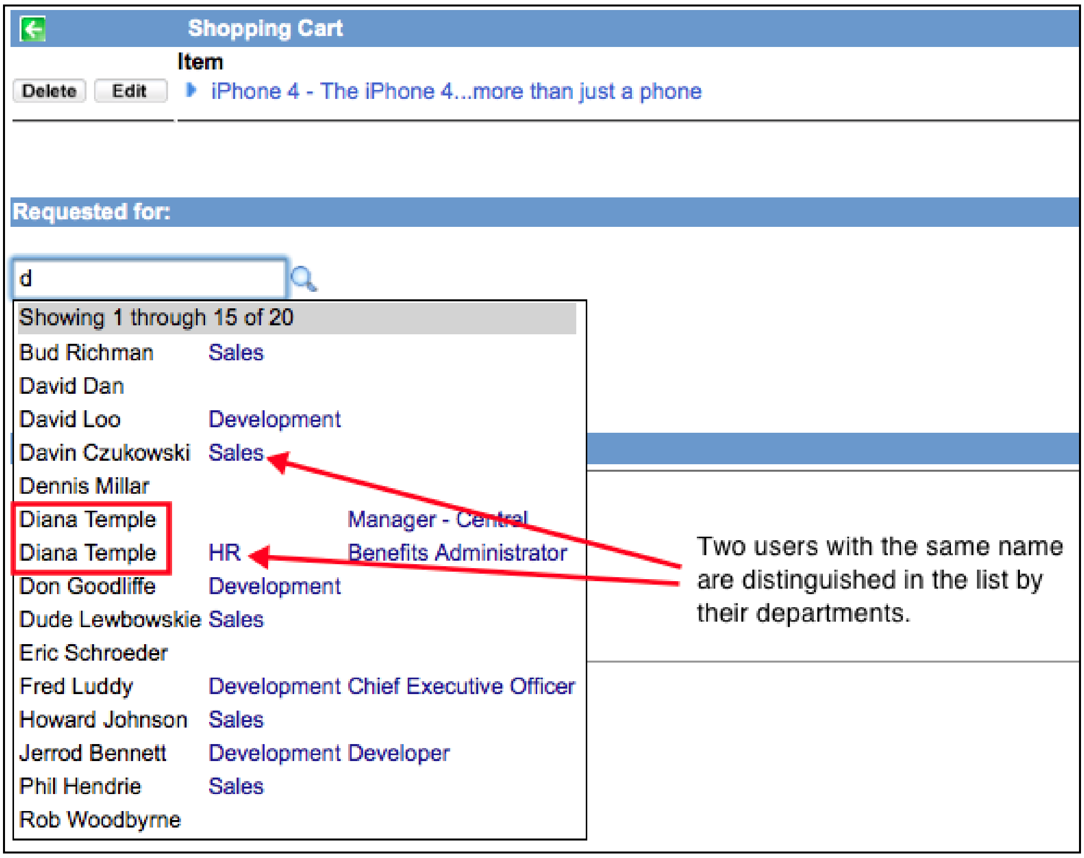
Using Properties to Refine Search Results

The auto-completion feature returns values that contain an exact match to the letter combination entered.



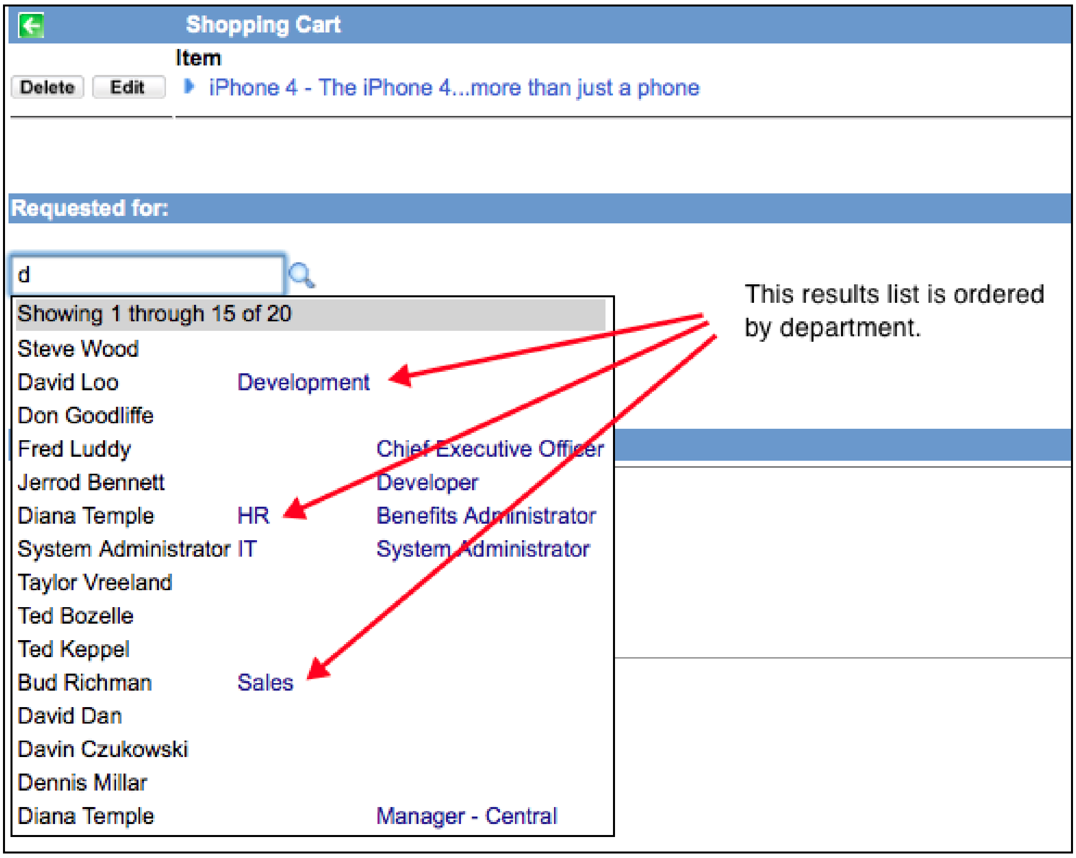
Administrators can use the **Additional columns for the "request for" Service Catalog widget** (`glide.sc_request_for.columns`) property to add columns to this list, to further refine the search results, and help determine which user to select when two users have the same name.

In this example, the property is set to display two additional columns, **Department** and **Title**:



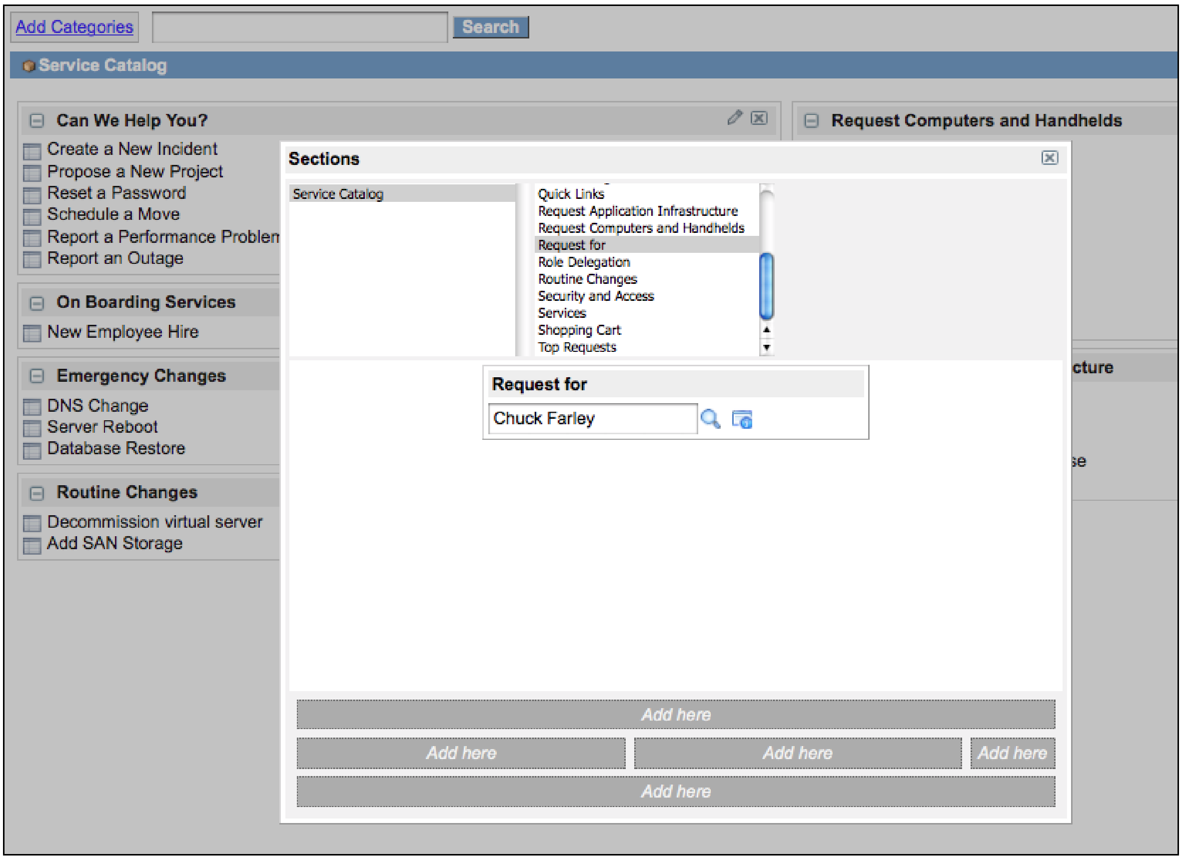
Administrators can use the **Ordering of matches for the "request for" Service Catalog widget.** (`glide.sc.request_for.order_by`) property to configure the columns to sort by one of the values.

In this example, the is set to sort the results list by **department**.



Auto-completion also applies to the **Request for** field, which can be added to the service catalog homepage.

1. Navigate to **Service Catalog > Catalog**.
2. Click **Add Categories**
3. Select **Request for**.
4. Place the category on the page.



Enabling Bulk Requests

Overview

Bulk requests allow a customer to create up to 10 copies (*clones*) of the same service catalog request, for different users, without having to enter the same information multiple times.

For example, you can request a new piece of equipment for several specified people, generating multiple individual requests quickly from a single initial request.

Enabling Bulk Requests

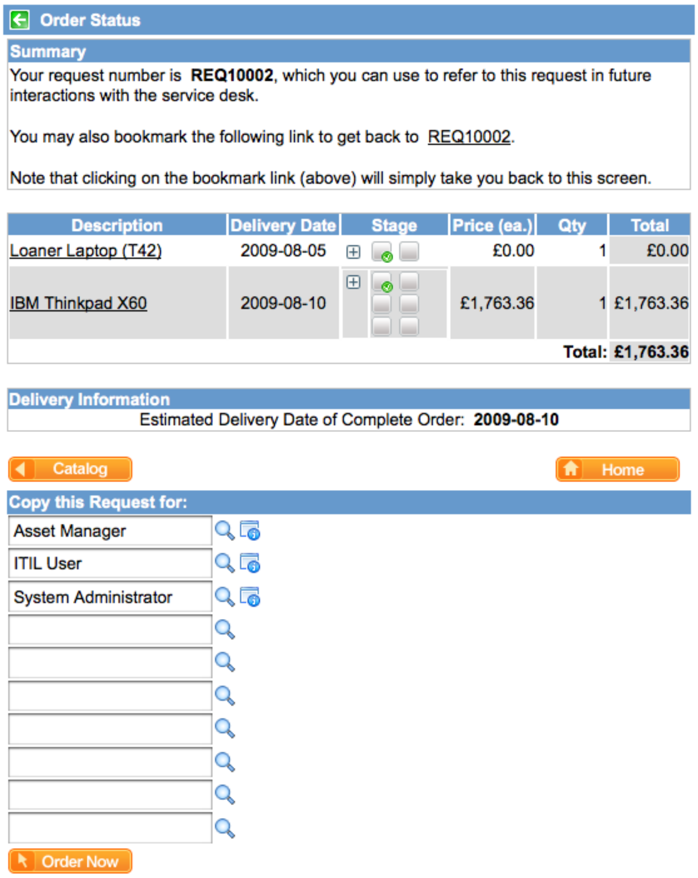
To enable bulk requests:

1. Navigate to **Service Catalog > Properties**.
2. Locate the property **Enable cloning requests during checkout**.
3. Select the **Yes** check box.
4. Locate the property **List of roles (comma-separated) that can use bulk ordering functionality.**
5. Enter the roles for which to enable bulk service catalog requests, or leave the field blank to enable the function for all users with a role.
6. Click **Save**.

Using Bulk Requests

To use bulk requests:

1. Create a service catalog request.
2. On the checkout page, enter the users for whom to copy the request (up to 10) in the **Copy this Request for:** reference fields.
3. Click **Order Now**.



Example: A duplicate request is created for the Asset Manager, an ITIL User, and the System Administrator.

Flexible Checkout and Delivery Forms

Overview

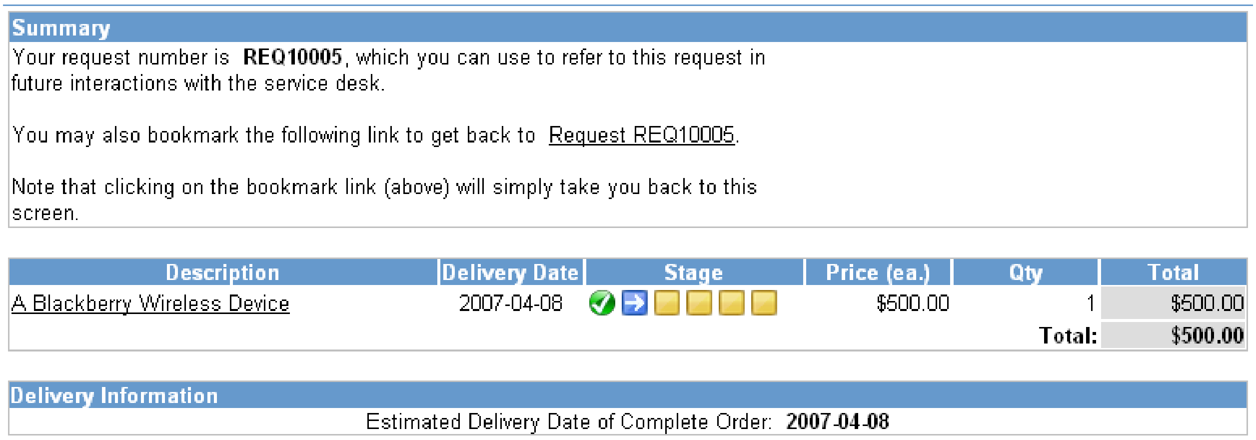
Some service catalog forms, such as the checkout form, are generated from templates, instead of being data-driven like other forms in the system, such as the Incident form.

Template-driven forms provide enhanced look-and-feel over standard data-driven forms, but they provide more limited control over the form content.

ServiceNow includes several options that provide administrators some control over content in service catalog's template-driven forms.

Modifying the Checkout Form

By default, the checkout forms list the **Description**, **Delivery Date**, **Stage**, **Price**, **Quantity** and **Total** columns. For example:



Administrators can use specific control options, to modify the checkout form to:

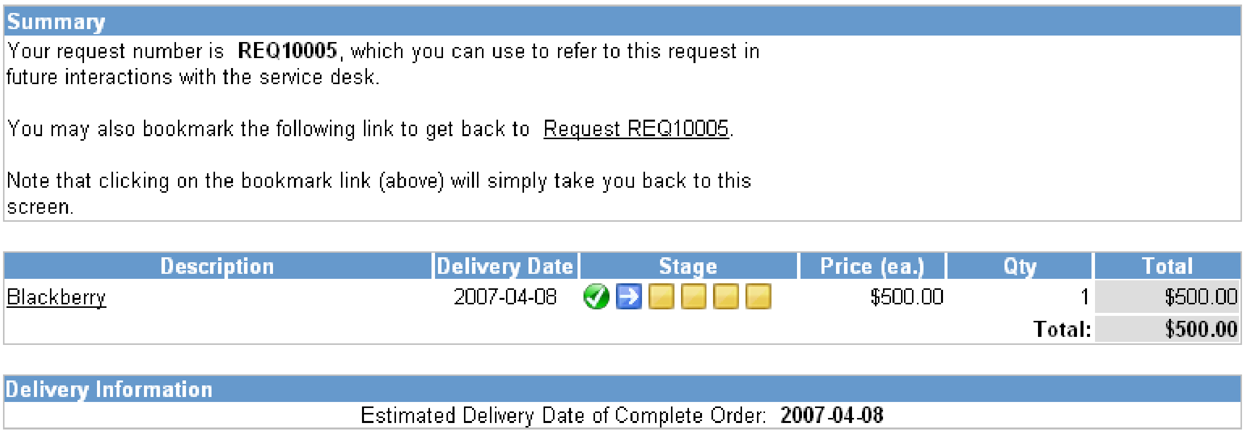
- Use an alternate description field
- Add the request item number for each line

Using an Alternate Description Field

By default, the **short_description** column of the catalog item appears as the item description.

To specify an alternate description field:

1. Navigate to **Service Catalog > Catalog Policy > Properties**.
2. Locate the property **Field name to use for the description column of the checkout form. If blank, the default (short_description) is used**.
3. Enter the name of the alternative field (a column in the Catalog Item [sc_cat_item] table) and save it. For example, if you selected **name**:



Summary

Your request number is **REQ10005**, which you can use to refer to this request in future interactions with the service desk.

You may also bookmark the following link to get back to Request REQ10005.

Note that clicking on the bookmark link (above) will simply take you back to this screen.

Description	Delivery Date	Stage	Price (ea.)	Qty	Total
Blackberry	2007-04-08		$500.00	1	$500.00
				Total:	**$500.00**

Delivery Information

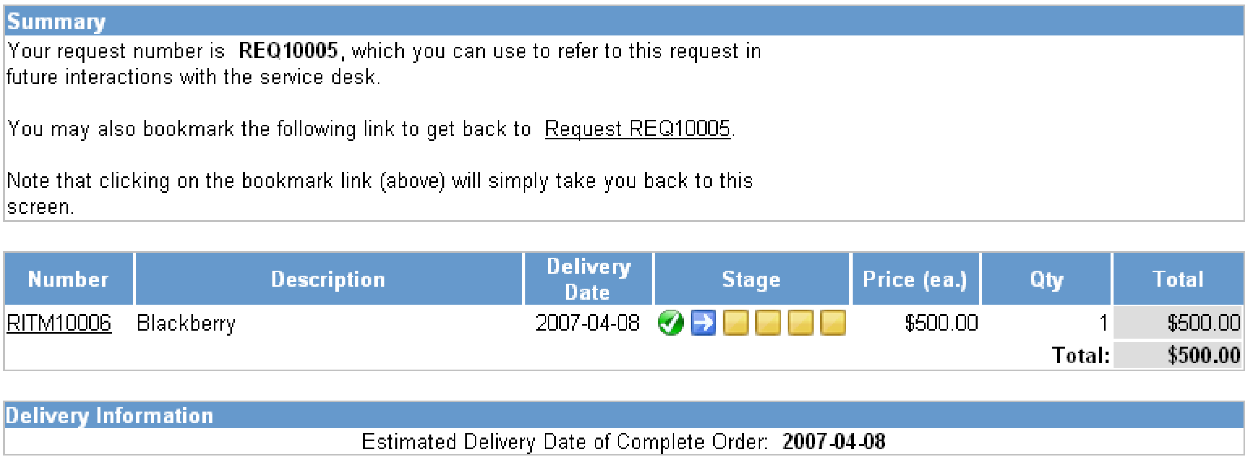
Estimated Delivery Date of Complete Order: **2007-04-08**

Adding the Request Item Number

By default, the request item number is not displayed in the list.

To display this number as an additional column:

1. Navigate to **Service Catalog > Catalog Policy > Properties**.
2. Locate the property: **Show the request item number for each line item on the checkout screen (default false)**.
3. Select the **Yes** check box to add the number column to the checkout form.



Summary

Your request number is **REQ10005**, which you can use to refer to this request in future interactions with the service desk.

You may also bookmark the following link to get back to Request REQ10005.

Note that clicking on the bookmark link (above) will simply take you back to this screen.

Number	Description	Delivery Date	Stage	Price (ea.)	Qty	Total
RITM10006	Blackberry	2007-04-08		$500.00	1	$500.00
					Total:	**$500.00**

Delivery Information

Estimated Delivery Date of Complete Order: **2007-04-08**

Modifying Catalog Button Properties

ServiceNow provides CSS properties that allow administrators to configure the look and feel of the buttons used in the service catalog, such as the Proceed to Checkout button, including the image used for the button background. Administrators can also modify catalog button properties. The ability to modify button properties is available starting with the Calgary release.

1. Navigate to **System Properties > CSS**
2. Locate and modify the following properties controlling button behavior:

- **Service Catalog button image:** the image to use for the button.

  Append an *x* to the filename, for example *white_gray_bevel.pngx* for the image *white_gray_bevel.png*.

  To modify the image itself, navigate to **System UI > Images**, locate the image from there, and make any changes required. See Storing Images in the Database for more information.

- **Service Catalog button image with mouse over:** the image to use when the mouse cursor hovers over the button.
- **Service Catalog button border:** the CSS value defining the border attributes for the button.

- **Service Catalog button text color:** the CSS value defining the text color for the button.

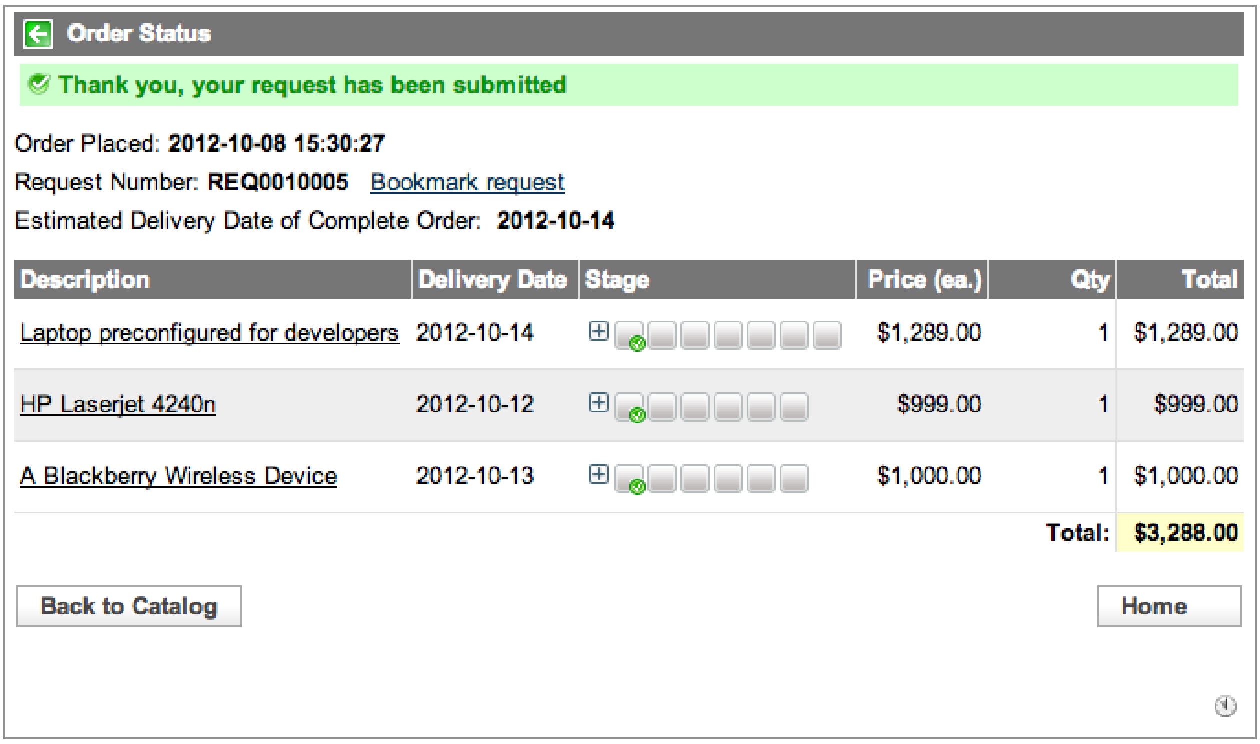
3. Save these values to apply them to your service catalog.

Note: *You may need to flush the server cache to see these changes, by entering the following in the browser's address bar:* ***<your_instance_name>.service-now.com/cache.do***

Modifying the Delivery Screen

In the final checkout step, a summary screen provides a list of all items and services ordered.

Warning: Modifying the Order Status or Summary screen requires advanced scripting and a knowledge of Jelly. Also, the upgrade process skips updates to the summary screen after a customization. With these constraints in mind, you can modify the summary screen by editing the com.glideapp.servicecatalog_checkout_view UI page.



Using Service Catalog Scripting

Creating a Catalog Client Script

Overview

Client-side scripts can add dynamic effects and validation to forms. Scripts can apply to service catalog items or variable sets, allowing catalog designers and administrators to use the same functionality that is available on other forms.

You can use client side scripts to:

- Get or set variable values.
- Hide or display variables.
- Make variables mandatory or not.
- Validate form submission.
- Add something to the cart programatically.
- Order something immediately.

Differences

Catalog client scripts are very similar to standard client scripts, with these important differences:

- Instead of selecting a table, like Incident, for the script, select a catalog item or variable set. Because there may be a large number of catalog items in the system, select a catalog item or variable set using a reference field instead of the choice list that the standard Client Script form uses.
- When using an `onChange()` catalog client script, it is linked to a particular variable instead of a field. The system automatically populates the variable name selection list with any named variables from the catalog item or variable set selected.

Considerations

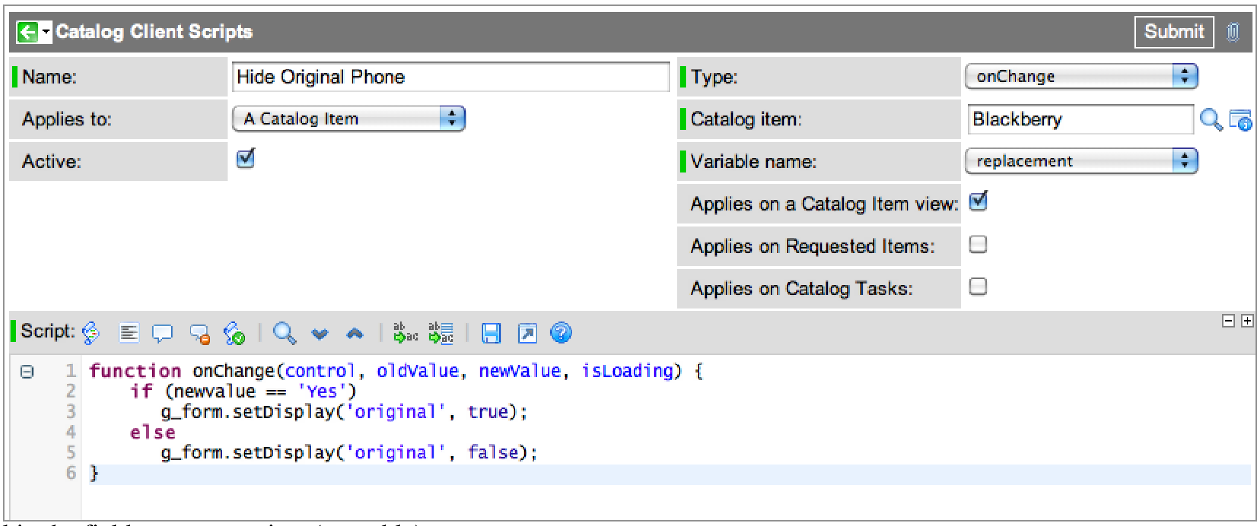
Keep the following points in mind when creating catalog client scripts:

- For a variable to be accessible via catalog client script, it must have a variable name. Variables without names do not appear in the list of available variables.
- Catalog client scripts run when a user orders an item from the service catalog. Catalog client scripts can also run when some or all variables, or variable sets, for a catalog item are displayed, when a user requests that item.
- When using standard client scripts on a Requested Item or Catalog Task form, make a note of fields with the same name as variables. If a table field and a variable of the same name are both present on a form, the table field will be matched when accessed using a script. In this scenario, specifically address the variable by prefixing its name with "variables.", for example: `g_form.setValue('variables.replacement', 'false');`
- If you are using record producers to pass variables from the service catalog to other types of records, these variables are made visible in those records with a variable editor, such as the Change Variable Editor UI formatter on Change request forms. You can manipulate these variables using standard client script methods, such as `setDisplay`, `setMandatory`, `setValue`, and `getValue`.
- Catalog client scripts can be used for catalog items included in a wizard. (Available in the Calgary release.)

Creating a Catalog Client Script

1. Navigate to **Service Catalog > Catalog Policy > Catalog Client Scripts**.

   A list of existing, custom catalog client scripts appears.
2. Click **New**.



3. Fill in the fields, as appropriate (see table).
4. Click **Submit**.

Field	Description
Name	Enter a unique name for the catalog client script.
Applies to	Select the item type this client script applies to: • **A Catalog Item:** enables the **Catalog item** field. • **A Variable Set:** enables the **Variable set** field.
Active	Select the check box to enable the client script. Clear the check box to disable the script.
Script	Enter the client script that should run on the service catalog item.
Type	Select when the script should run, such as **onLoad** or **onSubmit**.
Catalog item or Variable set	Select a catalog item or variable set from the list. The field name and options available depend on the selection in the **Applies to** field.
Applies on a Catalog Item view	Select the check box to apply the catalog client script to catalog items within the order screen. (Available in the Calgary release.)
Applies on Requested Items	Select the check box to apply the catalog client script on a Requested Item form. (Available in the Calgary release.)
Applies on Catalog Tasks	Select the check box to apply the catalog client script on a Catalog Task form. (Available in the Calgary release.)

Script Samples

Here are some examples of client scripts to perform common actions.

Get the Value of a Variable

Use the following syntax to obtain the value of a catalog variable. Note that the variable must have a name. Replace `variable_name` with the name of the variable.

```
g_form.getValue('variable_name');
```

Restrict the Number of Characters a User Can Type in a Variable

This is an example of a script that runs when the variable is displayed, rather than when the item is ordered.

```
function onLoad() {
  var sd = g_form.getControl('short_description');
  sd.maxLength = 80;
}
```

Update a Cart

To access and update cart contents, use the GlideAjax CartAjaxProcessor class.

Note: *Do not update the shopping cart using the client side implementation of GlideRecord to directly access the database and make cart updates, as this bypasses business rules to check whether the cart already existed. If you use GlideRecord, each cart update generates a new cart instead of updating the relevant cart, creating multiple duplicate carts.*

The following example shows how you can use GlideAjax to update a user's address stored in a cart:

```
// Get new address from field and pass to method that updates the cart
function onChange(control, oldValue, newValue, isLoading) {
    if (!isLoading && newValue != ''
            && g_form.getValue('location_listed') == 'Yes') {
        var loc = g_form.getReference('location');
        updateAddressInCart(g_user.userID,
            loc.u_street + '\n' + loc.u_city + ', ' + loc.u_state + ' '
 + loc.u_postal_code);
    }
}

function updateAddressInCart(userSysId, deliveryAddress) {
    var cart = new GlideRecord('sc_cart');      // Client Side
GlideRecord
    cart.addQuery('user', userSysId);              // pass in user id
    cart.query();                            // Execute the query
    if (cart.next()) {
        var ga = new GlideAjax('CartAjaxProcessor');              //
Class name
        ga.addParam('sysparm_action', 'set_delivery_address');      //
method name
        ga.addParam('sysparm_value', deliveryAddress);              //
```

```
method arg
        ga.addParam('sysparm_sys_id', cart.sys_id);                //
method arg
        ga.getXMLAnswer(function() {}, null, null);                //
updates cart on server
    }
}
```

Scripting Currency and Price Fields

Overview

Administrators can use currency and price fields in scripts.

You can obtain currency values using:

- **getReference** methods: apply to the system's base currency, when the item was created for the service catalog.
- **getCurrency** methods: apply to the currency used by the customer who ordered the item.
- **getSession** methods: apply to the currency of the currently logged-in user viewing the item, for example, a member of a fulfillment group processing the order.

Obtaining Currency Values

The following methods allow you to obtain currency values.

The **Example value returned** column shows details for an item :

- Created with an item price of $600 in a system base currency of USD (US dollars)
- Ordered by a British user in the GBP (British pounds) currency
- Viewed and fulfilled by a Spanish user in the EUR (Euro) currency

Method Name	Description	Example value returned
current.price.getReferenceCurrencyCode()	The currency ISO code, in the base system currency.	USD
current.price.getReferenceDisplayValue()	The price, including the currency symbol, in the base system currency.	$600.00
current.price.getReferenceValue()	A string representation of the field's value, in the system's basis currency.	600
current.price.getSessionCurrencyCode()	The currency ISO code, in the current user's currency.	EUR
current.price.getSessionDisplayValue() (or **current.price.getDisplayValue()** or **current.price**)	A string representation of the field's value, including the currency symbol, in the current user's currency.	€449.75
current.price.getSessionValue()	A string representation of the field's value, in the current user's currency.	449.75
current.price.getCurrencyCode()	The currency ISO code, in the currency used when the item was ordered.	GBP
current.price.getCurrencyDisplayValue()	A string representation of the field's value, including the currency symbol, in the currency used when the item was ordered.	£373.83
current.price.getCurrencyString()	A string including the ISO code as well as the price, in the currency used when the item was ordered.	GBP;373.83
current.price.getCurrencyValue()	The price, in the currency used when the item was ordered.	373.83

Note: *Currency symbols are displayed on the left or the right of the currency as appropriate.*

Examples

```
current.price=1000
```

This sets the current price = 1,000 of whatever the current session currency is. For example, for a Japanese user, the price is JPY (Japanese Yen) 1,000.

```
current.price='USD;1000'
```

Sets the current price = 1,000 USD.

Working with Values

Altering Values

You cannot simply add an integer to the return values, as they are returned as string values.

For example, with a current.price of 708.32, **current.price + 100** returns a value of 708.32100.

You can use a **parseFloat** function in your script to provide the correct result. For example, **parseFloat(current.price) + 100** will return 808.32.

Returning Predictable Values

Adding values to prices with scripts can be unpredictable. For example, consider a script which adds 100 to the price of our item.

```
var newPrice = parseFloat(current.price) + 100;
```

Running that script in the UK returns a value of **var newPrice = parseFloat(708.32) + 100** = 808.32.

However, if you run it in the US, this returns a value of **var newPrice = parseFloat(1000) + 100** = 1100.

To get the predictable values back into our item, regardless of the session under which the script runs, you can write your scripts to run in the system's base currency.

For example:

```
var newPrice = parseFloat(current.price.getReferenceValue()) + 100 ;
current.price
 = current.price.getReferenceCurrencyCode()
 + ';' + newPrice ;
```

Note: *This example still relies upon knowing what the base currency is, for the* ***100*** *to be meaningful.*

Service Catalog Script API

Overview

A scriptable API for the service catalog makes it easier to order from the catalog when using business rules.

The cart API allows you to order any quantity of catalog items, using the sys_id of the Catalog Item [sc_cat_item] you want. You can then set catalog variables to values as required, assuming the variables have names.

Note: *If the script runs as a result of a scheduled import, the script runs as* ***system*** *or as the user specified by that import in the* ***Run as*** *field. The script uses the specified user's cart. Each call to* `new Cart()` *empties the cart of the calling user, but does not use or empty any other carts.*

Examples

Ordering a single BlackBerry:

```
var cart = new Cart();
var item = cart.addItem('e2132865c0a8016500108d9cee411699'); var rc =
cart.placeOrder(); gs.addInfoMessage(rc.number);
```

Ordering twelve BlackBerries:

```
var cart = new Cart();
var item = cart.addItem('e2132865c0a8016500108d9cee411699', 12); var
rc = cart.placeOrder(); gs.addInfoMessage(rc.number);
```

Ordering an executive desktop and setting its OS:

```
var cart = new Cart();
var item = cart.addItem('e46305bdc0a8010a00645e608031eb0f');
cart.setVariable(item, 'os', 'Linux Red Hat');
var rc = cart.placeOrder();
gs.addInfoMessage(rc.number);
```

Creating a Custom Shopping Cart

Overview

Administrators with a knowledge of Jelly can customize the look and feel of service catalog carts, either globally or for specific catalog items, using our UI macro capability. Once created, a custom cart can be reused and linked to catalog items.

Creating a Cart

1. Navigate to **System UI > UI Macros**.
2. Click **New** to create a new macro.
3. Fill in the details and write the script to define your custom cart.

    Note: Designing a custom cart requires knowledge of the Jelly [1] expression language.

    For reference purposes, the default cart script appears in the list as *catalog_cart_default*.
4. Click **Submit**.

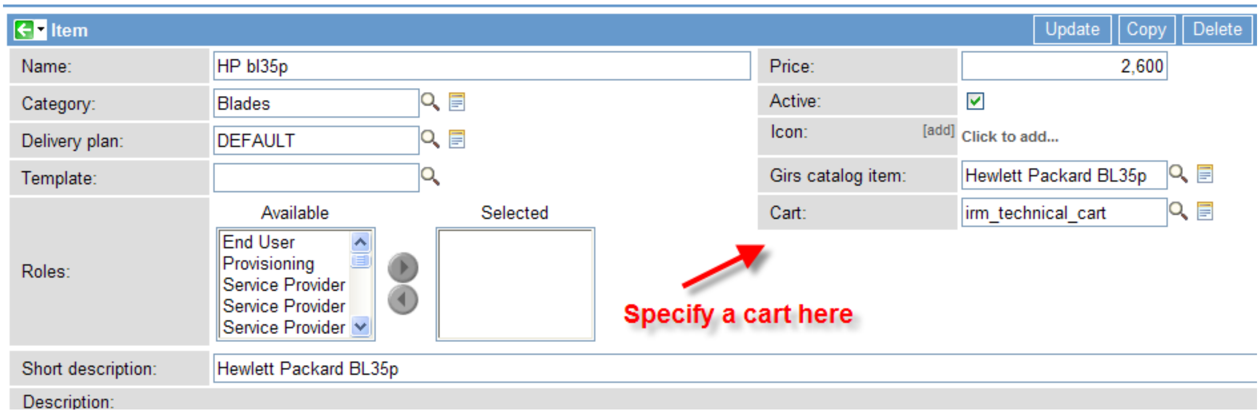
Linking a Cart to a Catalog Item

1. Navigate to **Service Catalog > Catalog Items**.
2. Open an item to test your new cart.
3. In the **Cart** field, select the cart to use.

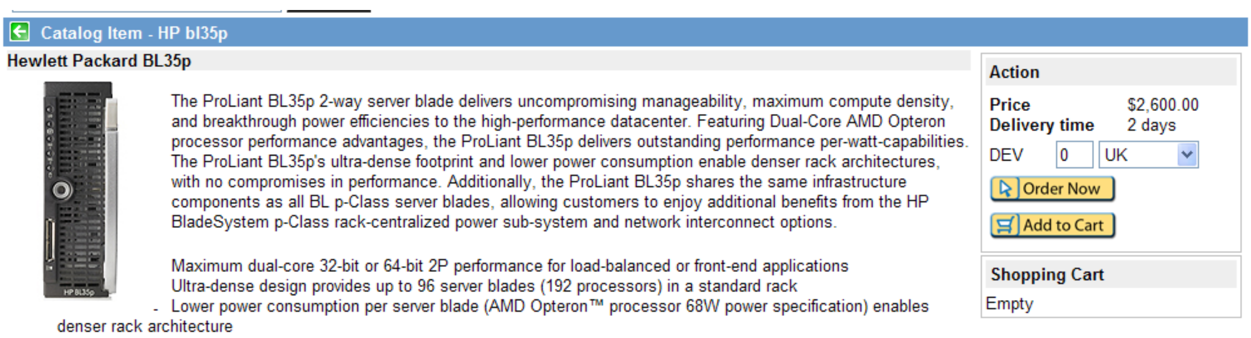
    You may need to personalize the form layout to add this field to the form.
4. Save the changed form.

Example

In the example below, the **HP bl35p** uses the **irm_technical_cart**.



This cart appears as follows:



References

[1] http://commons.apache.org/jelly/

Scriptable Assignment of Execution Plans

Overview

Each catalog item has an associated execution plan, used whenever an item of that type is ordered; if no plan is specified, ServiceNow uses the **default** plan. This default is effective for most organizations, but your execution plan may need to vary based on additional criteria.

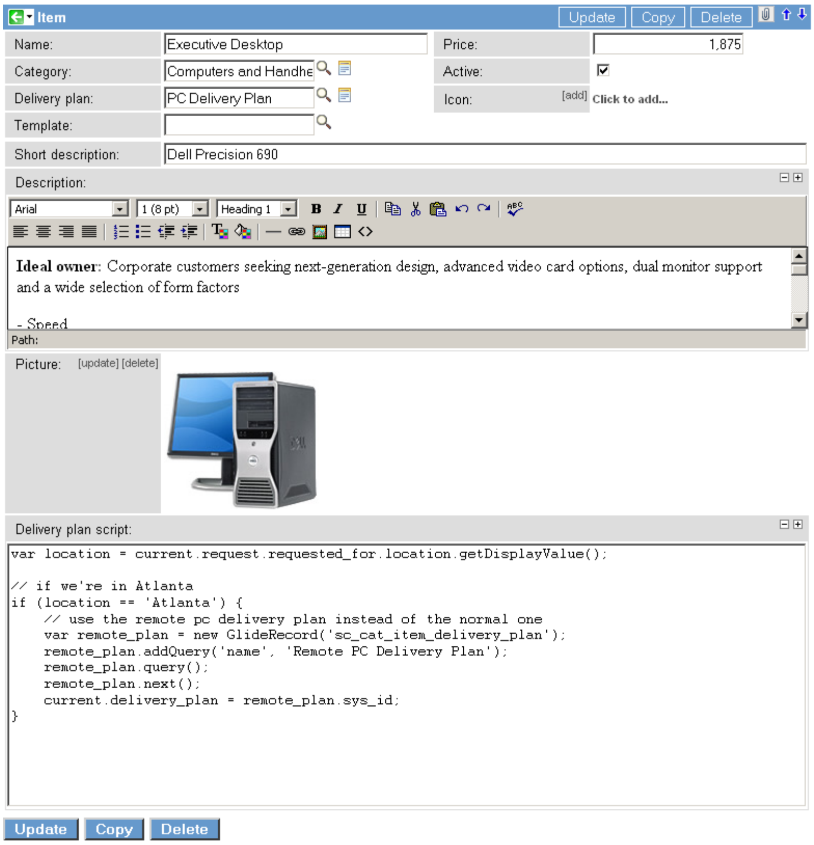
For example, in the base system service catalog, a request for a new PC always uses the *PC Delivery Plan*. However, this plan may need to vary for unusual circumstances - such as when a requester is working from home, at a remote location.

To provide this flexibility, you can use a script to override the default execution plan on a specific catalog item.

Using the Scripts

To make use of scriptable execution plans:

1. Navigate to **Service Catalog > Maintain Items**.
2. Select the relevant catalog item you wish to add the script to.
3. Personalize the catalog item form layout to add the execution plan script field, often named **Delivery Plan Script**.
4. Fill in the script details:



5. Update the item form with your changes.

The script runs each time that item is requested, selecting the execution plan to run with that item.

Writing the Scripts

Execution plan scripts can access the same global variables and other functions as in any other server side execution plan.

- **current** is the currently-requested catalog item, **sc_req_item**.
- **current.delivery_plan** is the assigned execution plan for this catalog item

The evaluated value from the script is used as the **sys_id** of the execution plan.

Simple example:

```
current.delivery_plan.setDisplayValue('PC Delivery Plan')
```

If an invalid value is returned, such as undefined or not found, then the existing assigned value is used.

More complex example:

```
getexecutionplan();
function getexecutionplan() {
var location = current.request.requested_for.location.getDisplayValue();
// if we're in Atlanta
if (location == 'Atlanta') {
   // use the remote pc delivery plan instead of the normal one
   var remote_plan = new GlideRecord('sc_cat_item_delivery_plan');
   remote_plan.addQuery('name', 'Remote PC Delivery Plan');
   remote_plan.query();
   remote_plan.next();
   current.delivery_plan = remote_plan.sys_id;
   return remote_plan_sys_id;
   {
   return current_delivery_plan;
}
```

In this example, any time a request is for a user in Atlanta, ServiceNow uses the *Remote PC Delivery Plan*. Otherwise, the execution plan is not overridden and ServiceNow uses the catalog item's normal execution plan, the *PC Delivery Plan*.

Limitations during script running

While the execution plan script runs:

- You cannot interact with any catalog tasks as catalog tasks are only created after the execution plan is selected.
- Some fields such as **total delivery time** and **due date** are not yet calculated, although the request itself is available within the script via **current.request**.
- Approvals have not yet been generated.

Article Sources and Contributors

Using the Service Catalog *Source*: http://wiki.servicenow.com/index.php?oldid=150434 *Contributors*: CapaJC, David.Bailey, Davida.hughes, G.yedwab, Guy.yedwab, Joseph.messerschmidt, Rachel.sienko, Rob.woodbyrne, Steve.wood, Vhearne

Placing a Request *Source*: http://wiki.servicenow.com/index.php?oldid=150398 *Contributors*: David.Bailey

Service Catalog Management *Source*: http://wiki.servicenow.com/index.php?oldid=198644 *Contributors*: Cheryl.dolan, David.Bailey, Davida.hughes, G.yedwab, Guy.yedwab, Jacebenson, Joseph.messerschmidt, Rachel.sienko, Steve.wood

Managing the Service Catalog Homepage *Source*: http://wiki.servicenow.com/index.php?oldid=190962 *Contributors*: David.Bailey, Suzannes

Service Catalog Categories *Source*: http://wiki.servicenow.com/index.php?oldid=193969 *Contributors*: CapaJC, David.Bailey, Guy.yedwab, Joseph.messerschmidt, Pat.Casey, Rachel.sienko, Steve.wood, Vhearne

Defining Catalog Items *Source*: http://wiki.servicenow.com/index.php?oldid=189576 *Contributors*: CapaJC, David.Bailey, G.yedwab, Guy.yedwab, Joseph.messerschmidt, Pat.Casey, Rachel.sienko, Rob.woodbyrne, Steve.wood, Vhearne, Wallymarx

Service Catalog Variables *Source*: http://wiki.servicenow.com/index.php?oldid=157782 *Contributors*: CapaJC, Cheryl.dolan, David.Bailey, Emily.partridge, Guy.yedwab, Joseph.messerschmidt, Pat.Casey, Rachel.sienko, Rob.woodbyrne, Suzannes, Vhearne, Wallymarx

Managing Request Fulfillment *Source*: http://wiki.servicenow.com/index.php?oldid=213542 *Contributors*: David.Bailey, G.yedwab, Guy.yedwab, Joseph.messerschmidt, Rachel.sienko

Service Catalog Workflows *Source*: http://wiki.servicenow.com/index.php?oldid=191250 *Contributors*: David.Bailey, G.yedwab, Guy.yedwab, Joseph.messerschmidt, Rachel.sienko, Steve.wood

Using Execution Plans *Source*: http://wiki.servicenow.com/index.php?oldid=157398 *Contributors*: David.Bailey

Using Order Guides *Source*: http://wiki.servicenow.com/index.php?oldid=195463 *Contributors*: CapaJC, Cheryl.dolan, David.Bailey, Gadi.yedwab, Guy.yedwab, John.andersen, Joseph.messerschmidt, Pat.Casey, Rachel.sienko, Steve.wood, Vhearne

Creating Record Producers *Source*: http://wiki.servicenow.com/index.php?oldid=189974 *Contributors*: CapaJC, Cheryl.dolan, David.Bailey, Eric.jacobson, Guy.yedwab, Jared.laethem, Jessi.graves, John.roberts, Joseph.messerschmidt, Rachel.sienko, Suzannes, Vhearne

Adding Content Items *Source*: http://wiki.servicenow.com/index.php?oldid=152567 *Contributors*: CapaJC, Cheryl.dolan, David.Bailey, Gadi.yedwab, Guy.yedwab, Joseph.messerschmidt, Mark.stanger, Pat.Casey, Rachel.sienko, Steve.wood, Vhearne

Service Catalog Variable Sets *Source*: http://wiki.servicenow.com/index.php?oldid=157269 *Contributors*: CapaJC, Cheryl.dolan, David.Bailey, Guy.yedwab, Joseph.messerschmidt, Pat.Casey, Rachel.sienko, Rob.woodbyrne, Steve.wood, Suzannes, Vhearne

Reference Qualifiers for Service Catalog Variables *Source*: http://wiki.servicenow.com/index.php?oldid=152643 *Contributors*: Cheryl.dolan, David.Bailey, Suzannes

Service Catalog Variable Pricing *Source*: http://wiki.servicenow.com/index.php?oldid=152739 *Contributors*: CapaJC, Cheryl.dolan, David.Bailey, Guy.yedwab, Joseph.messerschmidt, Rachel.sienko, Vhearne

Auditable Service Catalog Variables *Source*: http://wiki.servicenow.com/index.php?oldid=152539 *Contributors*: CapaJC, Cheryl.dolan, David.Bailey, Guy.yedwab, Joseph.messerschmidt, Pat.Casey, Rachel.sienko, Vhearne

Scriptable Service Catalog Variables *Source*: http://wiki.servicenow.com/index.php?oldid=152540 *Contributors*: CapaJC, Cheryl.dolan, David.Bailey, G.yedwab, Guy.yedwab, Joseph.messerschmidt, Neola, Pat.Casey, Rachel.sienko, Steve.wood, Vhearne

Reporting on Service Catalog Variables *Source*: http://wiki.servicenow.com/index.php?oldid=190780 *Contributors*: Cheryl.dolan, David.Bailey, Davida.hughes, Joseph.messerschmidt, Michael.randall, Rachel.sienko

Controlling Page Layout Using Variables *Source*: http://wiki.servicenow.com/index.php?oldid=152542 *Contributors*: Cheryl.dolan, David.Bailey

Service Catalog Access Controls *Source*: http://wiki.servicenow.com/index.php?oldid=203764 *Contributors*: CapaJC, Cheryl.dolan, Christen.mitchell, David.Bailey, Guy.yedwab, Joseph.messerschmidt, Pat.Casey, Rachel.sienko, Suzannes, Tyler.jones, Vhearne

Hiding Prices *Source*: http://wiki.servicenow.com/index.php?oldid=150423 *Contributors*: Christen.mitchell, David.Bailey, Guy.yedwab, Rachel.sienko, Steve.wood

Localizing Prices *Source*: http://wiki.servicenow.com/index.php?oldid=203766 *Contributors*: CapaJC, Cheryl.dolan, David.Bailey, G.yedwab, Guy.yedwab, Joseph.messerschmidt, Pat.Casey, Rachel.sienko, Steve.wood, Vhearne

UI Policy *Source*: http://wiki.servicenow.com/index.php?oldid=164236 *Contributors*: Cheryl.dolan, David.Bailey, Joseph.messerschmidt, Rachel.sienko, Steve.wood

Using Renderers *Source*: http://wiki.servicenow.com/index.php?oldid=150397 *Contributors*: David.Bailey

Data Lookup *Source*: http://wiki.servicenow.com/index.php?oldid=152719 *Contributors*: Cheryl.dolan, David.Bailey

Service Catalog Properties *Source*: http://wiki.servicenow.com/index.php?oldid=191410 *Contributors*: Daniel.taylor, David.Bailey

Enabling a Two-Step Checkout *Source*: http://wiki.servicenow.com/index.php?oldid=204622 *Contributors*: CapaJC, Cheryl.dolan, David.Bailey, Emily.partridge, Guy.yedwab, Joseph.messerschmidt, Pat.Casey, Rachel.sienko, Steve.wood, Suzannes, Vhearne

Enabling Bulk Requests *Source*: http://wiki.servicenow.com/index.php?oldid=199822 *Contributors*: David.Bailey, Guy.yedwab, Pat.Casey, Rachel.sienko, Vhearne

Flexible Checkout and Delivery Forms *Source*: http://wiki.servicenow.com/index.php?oldid=204628 *Contributors*: CapaJC, Cheryl.dolan, David.Bailey, Joseph.messerschmidt, Pat.Casey, Steve.wood, Vhearne

Creating a Catalog Client Script *Source*: http://wiki.servicenow.com/index.php?oldid=206717 *Contributors*: CapaJC, Cheryl.dolan, David.Bailey, Guy.yedwab, Joseph.messerschmidt, Mark.stanger, Neola, Rachel.sienko, Steve.wood, Vhearne

Scripting Currency and Price Fields *Source*: http://wiki.servicenow.com/index.php?oldid=204893 *Contributors*: CapaJC, Cheryl.dolan, David.Bailey, Guy.yedwab, Joseph.messerschmidt, Neola, Pat.Casey, Rachel.sienko, Vhearne

Service Catalog Script API *Source*: http://wiki.servicenow.com/index.php?oldid=204891 *Contributors*: CapaJC, Cheryl.dolan, David.Bailey, Guy.yedwab, Joseph.messerschmidt, Neola, Pat.Casey, Phillip.salzman, Rachel.sienko, Ray.lee, Steve.wood, Suzannes, Vhearne

Creating a Custom Shopping Cart *Source*: http://wiki.servicenow.com/index.php?oldid=204894 *Contributors*: CapaJC, Cheryl.dolan, David.Bailey, Eric.jacobson, Guy.yedwab, Jared.laethem, Joseph.messerschmidt, Pat.Casey, Rachel.sienko, Vhearne

Scriptable Assignment of Execution Plans *Source*: http://wiki.servicenow.com/index.php?oldid=204898 *Contributors*: CapaJC, Cheryl.dolan, David.Bailey, Emily.partridge, Guy.yedwab, John.roberts, Joseph.messerschmidt, Neola, Pat.Casey, Phillip.salzman, Rob.woodbyrne, Steve.wood, Vhearne

Image Sources, Licenses and Contributors

Image:service_catalog_homepage.png *Source*: http://wiki.servicenow.com/index.php?title=File:Service_catalog_homepage.png *License*: unknown *Contributors*: Suzannes

Image:service_catalog_categories_items.png *Source*: http://wiki.servicenow.com/index.php?title=File:Service_catalog_categories_items.png *License*: unknown *Contributors*: Suzannes

Image:service_catalog_search_dropdown.png *Source*: http://wiki.servicenow.com/index.php?title=File:Service_catalog_search_dropdown.png *License*: unknown *Contributors*: Suzannes

Image:service_catalog_results.png *Source*: http://wiki.servicenow.com/index.php?title=File:Service_catalog_results.png *License*: unknown *Contributors*: Suzannes

Image:service_catalog_search_result_amount.png *Source*: http://wiki.servicenow.com/index.php?title=File:Service_catalog_search_result_amount.png *License*: unknown *Contributors*: Suzannes

Image:service_catalog item.png *Source*: http://wiki.servicenow.com/index.php?title=File:Service_catalog_item.png *License*: unknown *Contributors*: Suzannes

Image:service_catalog_search_cancel.png *Source*: http://wiki.servicenow.com/index.php?title=File:Service_catalog_search_cancel.png *License*: unknown *Contributors*: Suzannes

Image:catalog_item_more_info.png *Source*: http://wiki.servicenow.com/index.php?title=File:Catalog_item_more_info.png *License*: unknown *Contributors*: Suzannes

Image:service_catalog_request_status.png *Source*: http://wiki.servicenow.com/index.php?title=File:Service_catalog_request_status.png *License*: unknown *Contributors*: Suzannes

Image:CatalogUse1.png *Source*: http://wiki.servicenow.com/index.php?title=File:CatalogUse1.png *License*: unknown *Contributors*: CapaJC

Image:CatalogUse2.png *Source*: http://wiki.servicenow.com/index.php?title=File:CatalogUse2.png *License*: unknown *Contributors*: CapaJC

Image:Order-submitted.png *Source*: http://wiki.servicenow.com/index.php?title=File:Order-submitted.png *License*: unknown *Contributors*: David.Bailey

File:Service_Catalog_modules.png *Source*: http://wiki.servicenow.com/index.php?title=File:Service_Catalog_modules.png *License*: unknown *Contributors*: David.Bailey

Image:Warning.gif *Source*: http://wiki.servicenow.com/index.php?title=File:Warning.gif *License*: unknown *Contributors*: CapaJC

Image:Service_Catalog_add_categories.png *Source*: http://wiki.servicenow.com/index.php?title=File:Service_Catalog_add_categories.png *License*: unknown *Contributors*: David.Bailey

Image:Service_Catalog_Mobile_Layout.png *Source*: http://wiki.servicenow.com/index.php?title=File:Service_Catalog_Mobile_Layout.png *License*: unknown *Contributors*: David.Bailey

Image:ServiceCatalogCategories.png *Source*: http://wiki.servicenow.com/index.php?title=File:ServiceCatalogCategories.png *License*: unknown *Contributors*: David.Bailey

Image:MaintainCategories-Dublin.png *Source*: http://wiki.servicenow.com/index.php?title=File:MaintainCategories-Dublin.png *License*: unknown *Contributors*: David.Bailey

Image:Service_Catalog_Item_Details.png *Source*: http://wiki.servicenow.com/index.php?title=File:Service_Catalog_Item_Details.png *License*: unknown *Contributors*: David.Bailey

Image:Icon-reference.png *Source*: http://wiki.servicenow.com/index.php?title=File:Icon-reference.png *License*: unknown *Contributors*: Guy.yedwab

Image:Additional.png *Source*: http://wiki.servicenow.com/index.php?title=File:Additional.png *License*: unknown *Contributors*: Guy.yedwab, Rachel.sienko

Image:recurring_price.png *Source*: http://wiki.servicenow.com/index.php?title=File:Recurring_price.png *License*: unknown *Contributors*: Suzannes

Image:recurring_shop_cart.png *Source*: http://wiki.servicenow.com/index.php?title=File:Recurring_shop_cart.png *License*: unknown *Contributors*: Suzannes

Image:recurring_price_rollup.png *Source*: http://wiki.servicenow.com/index.php?title=File:Recurring_price_rollup.png *License*: unknown *Contributors*: Suzannes

Image:recurring_price_request_rollup.png *Source*: http://wiki.servicenow.com/index.php?title=File:Recurring_price_request_rollup.png *License*: unknown *Contributors*: Suzannes

Image:Catalog_No_Quantity.png *Source*: http://wiki.servicenow.com/index.php?title=File:Catalog_No_Quantity.png *License*: unknown *Contributors*: Rachel.sienko, Steve.wood

Image:Catalog_Item_View_Quantity.png *Source*: http://wiki.servicenow.com/index.php?title=File:Catalog_Item_View_Quantity.png *License*: unknown *Contributors*: Steve.wood

Image:Catalog_Item_Hide_Quantity.png *Source*: http://wiki.servicenow.com/index.php?title=File:Catalog_Item_Hide_Quantity.png *License*: unknown *Contributors*: Steve.wood

Image:CatalogQuantity1.png *Source*: http://wiki.servicenow.com/index.php?title=File:CatalogQuantity1.png *License*: unknown *Contributors*: CapaJC, Guy.yedwab, Steve.wood

Image:Variable personalize.png *Source*: http://wiki.servicenow.com/index.php?title=File:Variable_personalize.png *License*: unknown *Contributors*: CapaJC, Pat.Casey, Rachel.sienko, Steve.wood

Image:SCWorkflow1.png *Source*: http://wiki.servicenow.com/index.php?title=File:SCWorkflow1.png *License*: unknown *Contributors*: Guy.yedwab

Image:SCWorkflow2.png *Source*: http://wiki.servicenow.com/index.php?title=File:SCWorkflow2.png *License*: unknown *Contributors*: Guy.yedwab

Image:Og text.png *Source*: http://wiki.servicenow.com/index.php?title=File:Og_text.png *License*: unknown *Contributors*: David.Bailey, Guy.yedwab, Pat.Casey, Rachel.sienko

Image:Og variable.png *Source*: http://wiki.servicenow.com/index.php?title=File:Og_variable.png *License*: unknown *Contributors*: David.Bailey, Pat.Casey

Image:Og rule.png *Source*: http://wiki.servicenow.com/index.php?title=File:Og_rule.png *License*: unknown *Contributors*: David.Bailey, Pat.Casey

Image:Og rule detail.png *Source*: http://wiki.servicenow.com/index.php?title=File:Og_rule_detail.png *License*: unknown *Contributors*: Pat.Casey

Image:Og rule detail3.png *Source*: http://wiki.servicenow.com/index.php?title=File:Og_rule_detail3.png *License*: unknown *Contributors*: Pat.Casey

Image:Og sample 1.png *Source*: http://wiki.servicenow.com/index.php?title=File:Og_sample_1.png *License*: unknown *Contributors*: David.Bailey, Guy.yedwab, Pat.Casey

Image:Og sample 2.png *Source*: http://wiki.servicenow.com/index.php?title=File:Og_sample_2.png *License*: unknown *Contributors*: David.Bailey, Guy.yedwab, Pat.Casey

Image:Og sample 4.png *Source*: http://wiki.servicenow.com/index.php?title=File:Og_sample_4.png *License*: unknown *Contributors*: Guy.yedwab, Pat.Casey

Image:Og sample 5.png *Source*: http://wiki.servicenow.com/index.php?title=File:Og_sample_5.png *License*: unknown *Contributors*: David.Bailey, Guy.yedwab, Pat.Casey

Image:Record Producer.png *Source*: http://wiki.servicenow.com/index.php?title=File:Record_Producer.png *License*: unknown *Contributors*: Cheryl.dolan

Image:RecordProducer.png *Source*: http://wiki.servicenow.com/index.php?title=File:RecordProducer.png *License*: unknown *Contributors*: Rachel.sienko

image:Content Item Form.png *Source*: http://wiki.servicenow.com/index.php?title=File:Content_Item_Form.png *License*: unknown *Contributors*: Cheryl.dolan

Image:variable_sections.png *Source*: http://wiki.servicenow.com/index.php?title=File:Variable_sections.png *License*: unknown *Contributors*: Suzannes

Image:SC_requestreport.png *Source*: http://wiki.servicenow.com/index.php?title=File:SC_requestreport.png *License*: unknown *Contributors*: Rachel.sienko

Image:SC_databaseview.png *Source*: http://wiki.servicenow.com/index.php?title=File:SC_databaseview.png *License*: unknown *Contributors*: Rachel.sienko

Image:SC_taskreport.png *Source*: http://wiki.servicenow.com/index.php?title=File:SC_taskreport.png *License*: unknown *Contributors*: Rachel.sienko

Image:Roles_granted.png *Source*: http://wiki.servicenow.com/index.php?title=File:Roles_granted.png *License*: unknown *Contributors*: David.Bailey

Image:Access control.png *Source*: http://wiki.servicenow.com/index.php?title=File:Access_control.png *License*: unknown *Contributors*: David.Bailey, Pat.Casey

Image:Entitlement script.png *Source*: http://wiki.servicenow.com/index.php?title=File:Entitlement_script.png *License*: unknown *Contributors*: Pat.Casey

Image:Catalog Order.png *Source*: http://wiki.servicenow.com/index.php?title=File:Catalog_Order.png *License*: unknown *Contributors*: Christen.mitchell

Image:Catalog subcategory.png *Source*: http://wiki.servicenow.com/index.php?title=File:Catalog_subcategory.png *License*: unknown *Contributors*: Guy.yedwab

Image:Shoppingcart.png *Source*: http://wiki.servicenow.com/index.php?title=File:Shoppingcart.png *License*: unknown *Contributors*: Rachel.sienko

Image:Edit cart.png *Source*: http://wiki.servicenow.com/index.php?title=File:Edit_cart.png *License*: unknown *Contributors*: Christen.mitchell

Image:Order Conformation.png *Source*: http://wiki.servicenow.com/index.php?title=File:Order_Conformation.png *License*: unknown *Contributors*: Christen.mitchell

Image:Order Summary.png *Source*: http://wiki.servicenow.com/index.php?title=File:Order_Summary.png *License*: unknown *Contributors*: Christen.mitchell

Image:catalog_ui_policy_calgary1.png *Source*: http://wiki.servicenow.com/index.php?title=File:Catalog_ui_policy_calgary1.png *License*: unknown *Contributors*: Suzannes

Image:catalog_ui_policy_calgary2.png *Source*: http://wiki.servicenow.com/index.php?title=File:Catalog_ui_policy_calgary2.png *License*: unknown *Contributors*: Suzannes

Image:catalog_ui_policy_calgary3.png *Source*: http://wiki.servicenow.com/index.php?title=File:Catalog_ui_policy_calgary3.png *License*: unknown *Contributors*: Suzannes

Image:Category-Renderer-Select.png *Source*: http://wiki.servicenow.com/index.php?title=File:Category-Renderer-Select.png *License*: unknown *Contributors*: David.Bailey

Image:Renderers-select.png *Source*: http://wiki.servicenow.com/index.php?title=File:Renderers-select.png *License*: unknown *Contributors*: David.Bailey

Image:Renderers.png *Source*: http://wiki.servicenow.com/index.php?title=File:Renderers.png *License*: unknown *Contributors*: David.Bailey

Image:Service_Catalog_Lookup1.png *Source*: http://wiki.servicenow.com/index.php?title=File:Service_Catalog_Lookup1.png *License*: unknown *Contributors*: David.Bailey

Image:Service_Catalog_Lookup0.png *Source*: http://wiki.servicenow.com/index.php?title=File:Service_Catalog_Lookup0.png *License*: unknown *Contributors*: David.Bailey

Image:Service_Catalog_Lookup01.png *Source*: http://wiki.servicenow.com/index.php?title=File:Service_Catalog_Lookup01.png *License*: unknown *Contributors*: David.Bailey

Image:Requested_for.png *Source*: http://wiki.servicenow.com/index.php?title=File:Requested_for.png *License*: unknown *Contributors*: David.Bailey

Image:Catalog_Auto_Complete1.png *Source*: http://wiki.servicenow.com/index.php?title=File:Catalog_Auto_Complete1.png *License*: unknown *Contributors*: Steve.wood, Suzannes
Image:Catalog_Auto_Complete2.png *Source*: http://wiki.servicenow.com/index.php?title=File:Catalog_Auto_Complete2.png *License*: unknown *Contributors*: Steve.wood, Suzannes
Image:Catalog_Auto_Complete7.png *Source*: http://wiki.servicenow.com/index.php?title=File:Catalog_Auto_Complete7.png *License*: unknown *Contributors*: Steve.wood, Suzannes
Image:Catalog_Auto_Complete4.png *Source*: http://wiki.servicenow.com/index.php?title=File:Catalog_Auto_Complete4.png *License*: unknown *Contributors*: Steve.wood, Suzannes
Image:Request clone2.png *Source*: http://wiki.servicenow.com/index.php?title=File:Request_clone2.png *License*: unknown *Contributors*: Guy.yedwab, Pat.Casey
Image:Checkout default.png *Source*: http://wiki.servicenow.com/index.php?title=File:Checkout_default.png *License*: unknown *Contributors*: Pat.Casey
Image:Checkout name.png *Source*: http://wiki.servicenow.com/index.php?title=File:Checkout_name.png *License*: unknown *Contributors*: Pat.Casey
Image:Checkout number.png *Source*: http://wiki.servicenow.com/index.php?title=File:Checkout_number.png *License*: unknown *Contributors*: Pat.Casey
Image:Caution-diamond.png *Source*: http://wiki.servicenow.com/index.php?title=File:Caution-diamond.png *License*: unknown *Contributors*: John.roberts
Image:catalog_order_summary_status.png *Source*: http://wiki.servicenow.com/index.php?title=File:Catalog_order_summary_status.png *License*: unknown *Contributors*: Suzannes
Image:catalog_client_script.png *Source*: http://wiki.servicenow.com/index.php?title=File:Catalog_client_script.png *License*: unknown *Contributors*: Suzannes
Image:Custom cart.png *Source*: http://wiki.servicenow.com/index.php?title=File:Custom_cart.png *License*: unknown *Contributors*: Pat.Casey
Image:Custom cart2.png *Source*: http://wiki.servicenow.com/index.php?title=File:Custom_cart2.png *License*: unknown *Contributors*: Pat.Casey
Image:Dp script.png *Source*: http://wiki.servicenow.com/index.php?title=File:Dp_script.png *License*: unknown *Contributors*: Pat.Casey