# Training:ServiceNow Certification

## Overview

ServiceNow offers a professional certification program. Certification assures employers and peers that you possess the skills and knowledge to perform the duties involved in ServiceNow system administration.

## Resources

To plan the ServiceNow education that will help you achieve your certification goals, review these resources.

- The Customer Training Roadmap (PDF) presents the certification paths for ServiceNow System Administrators, Developers, and Architects. Learn how our training offerings align with your certification goals.
- The ServiceNow Community Learning Center [1] provides free access to the Foundations elearning modules, webinars, and other training products.
- The ServiceNow Certification web page [2] lists the latest training and professional conference news and information.

## ServiceNow Certified System Administrator

Becoming a Certified System Administrator establishes a foundation for continuing to the advanced levels of ServiceNow knowledge. The Certified System Administrator Exam Blueprint (PDF) contains specific details about the prerequisites, learning objectives, and format of the ServiceNow Certified System Administrator exam. Sample questions are provided.

The Certified System Administrator exam should be taken after completing the 3-day System Administration course and it is recommended that certification candidates have at least six months to a year of ServiceNow system experience.

The System Administration course achieves these learning objectives through instruction taught by professional trainers, hands-on exercises, and live demonstrations. Visit the ServiceNow Training web page [3] for class descriptions and to view the list of ServiceNow Authorized Training Partners.

- Create new applications, modules, and tables
- Activate a plugin
- Personalize and create forms and fields
- Build reports, gauges, and homepages
- Set up a service level agreement (SLA) and monitor an SLA workflow
- Create and track service catalog requests and items with variables
- Create, monitor, modify, and publish service catalog workflows with approvals
- Follow the notification process back to the business rule
- Import data into the instance
- Create and use update sets to move customizations between systems
- Search, populate, and customize the knowledge base
- Create an access control rule (ACL)
- Create global security controls for attachments and cookies
- Create a chat room and set up help desk chat

## Register for a Certification Exam

ServiceNow offers several exam delivery options through the Kryterion Webassessor [4] testing platform. Schedule your exam appointment today!

### References

[1] http://community.service-now.com/learning-center
[2] http://www.service-now.com/knowledge.do?sysparm_document_key=kb_knowledge,56ebb431ff620000ebc5faa6497efe66
[3] http://www.service-now.com/instructor-led-training.do
[4] https://www.webassessor.com/servicenow

# Administering Applications and Modules

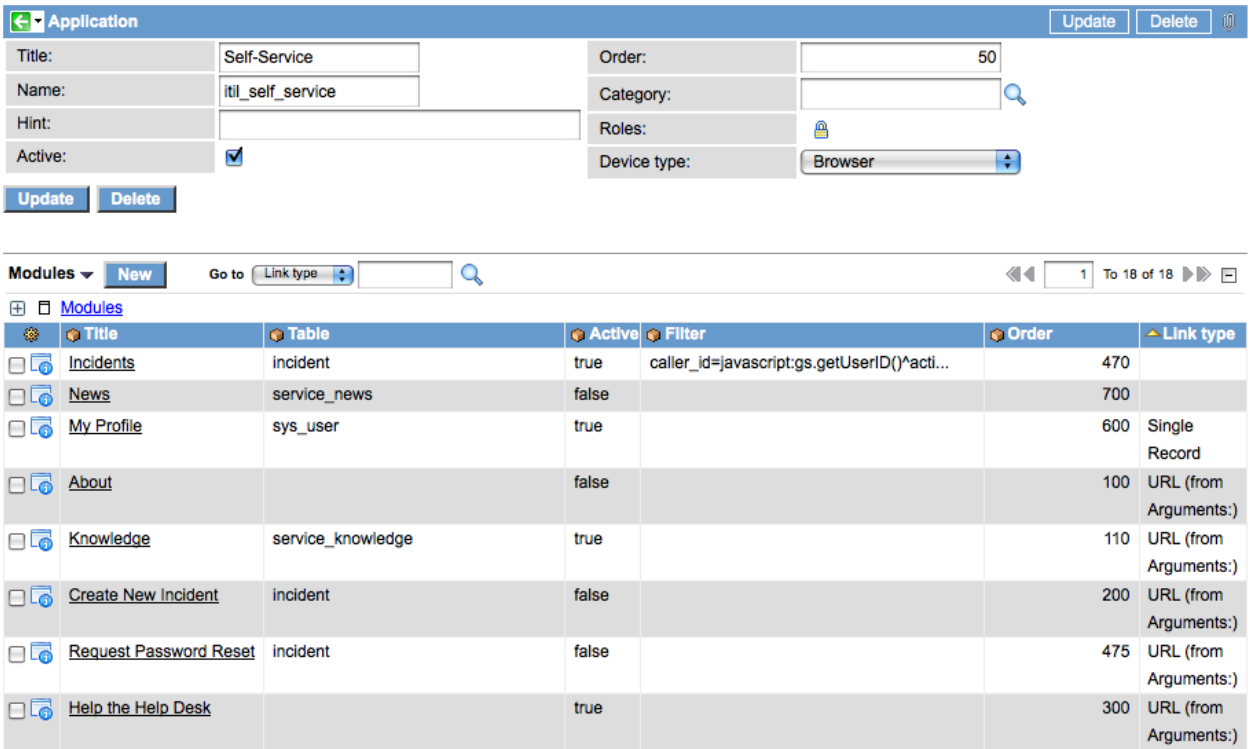
Functionality described here requires the **Admin** role.

## Overview

Administrators have the ability to create or modify applications and to create or modify the modules within those applications. The ability to modify the displays of Applications and their Modules grants the administrator control over what areas of the system are accessible by certain users and roles.

Application Modules appear as links under each application's navigation bar heading. They have the purpose of presenting the functionalities that make up an application.

Below is a screenshot of an application with its modules:



# Building a New Application

This video shows you how to build a new application on the platform:

Another method of creating an application and module is as part of the process of creating a table. For more, see Creating a Custom Table.

# Enabling and Disabling an Application or Module

To enable (show) or disable (hide) an application or module in ServiceNow, Navigate to **System Definition** --> **Applications**. Click the application you want to modify, and you'll be shown a list of its modules. The list of applications you see may default to only show active ones. If this is the case, click the "Applications" breadcrumb at the top of the page (to remove the "active = true" filter condition), and you'll then see all applications, whether active or not.

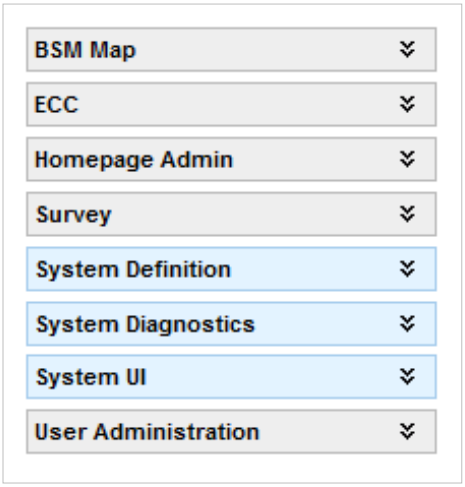
To enable/disable the entire application (e.g., ITIL Incident or Service Catalog), simply check or clear the **Active** box. You could also restrict the application to specific roles by adding or removing them here as desired. Click Update .

To enable/disable a specific module of an application, click the module's name to open it, and check or clear the **Active** box inside. You can restrict the module to specific roles by adding or removing roles here as desired. Click **Update** when finished.

As of the Aspen release, you can enable/disable multiple modules of an application at the same time. Select the check boxes next to the module names and from the **Actions on selected rows** drop-down list, select **Change active state**.

When changes are made to Applications or Modules, the application navigator will automatically refresh to display the changes.

## Modifying Application Categories




Application categories (navigate to **System Definition** --> **Application Categories**) let you put CSS style on application labels to make them stand out from other applications. This CSS style can include border-color, text color, background-color, etc.

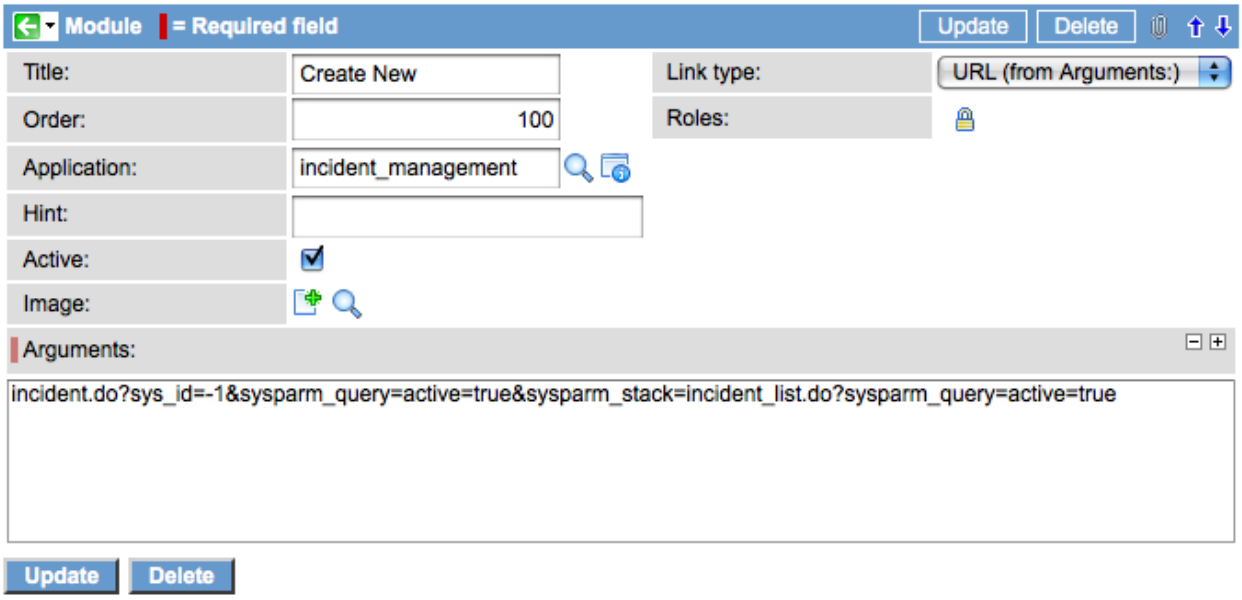
Out of box, there are three application categories:

- **Administration** (applications in this category have a light blue background)
  - Style: border-color: #a7cded; background-color: #e3f3ff;
- **Label** (applications in this category have a dark blue background)
  - Style: border-color: blue; background-color: rgb(102, 153, 204); color: white
- **Maint**
  - Style: border-color: blue; background-color: rgb(102, 153, 204); color: white

To add an application to a category, add it in the related list of applications at the bottom of the category record, or else you can specify the category on the application record.

# Defining a Module

Modules can be found, created, and edited by browsing to *System Definition -> Applications*. Click on the appropriate Application, and near the bottom of the screen will be the list of Modules associated with the application selected. A module's **Title** should clearly identify the module. The **Table Name** specifies which table this module is related to or operates on. The **Order** field specifies a number that is used to sort the order the modules appear under the menu list of an application. The **Application** field specifies the application (by name) to which this module belongs.




## Module Link Types

Module Link Types specifies what type of link would be created for this module under the Application menu:

| Field | Input Value |
|---|---|
| Homepage | Link that displays a homepage, selected using a reference field. |
| HTML (from Arguments:) | Places HTML in the application navigator. Is best used for more complicated links, where a flat URL is not customizable enough. |
| List Filter | Link that displays an unpopulated record list view with a blank filter open. This allows the user to quickly specify how they want to filter the table they are looking at. |
| List of Records | Link that displays the record list view for the table defined by Table Name. Allows for filters using the **Filter** field. |
| Map Page | Links to a Map Page using a reference field. See Using Map Pages |
| New Record | Link that displays a form for creating a new record for the table. |
| Populated Filter | Link that displays an record list view with a populated filter open. This presents the user with a filtered view of a table, along with the ability to quickly change the filter to a new one. |
| Run a Report | Automatically runs a predetermined report. |
| Script (from Arguments:) | Runs a script, as defined within arguments. |
| Search Screen | Link that displays a blank form for searching records in the table.<br>**Note:** use the parameter *&sysparm_result_view=view_name* to define the view the results are rendered in. |
| Separator | Creates a division between modules, with a name, that can be folded or unfolded. |

| | |
|---|---|
| Single Record | Link that displays a form for a single record from the table. |
| Survey | Links to a survey using a reference field. |
| URL (from Arguments:) | Any URL<br>• For internal links this should always be a relative link such as "./catalog_home.do?sysparm_view=catalog_default" or "catalog_home.do?sysparm_view=catalog_default"<br>• You should never use an absolute link to a ServiceNow instance like "https://service-now.com/demo/catalog_home.do?sysparm_view=catalog_default. This creates problems when you move an update set from a development instance to a production instance, because the URL still carries the reference to the development instance. |

## Specifying a View Name

A View is an organized and customized form or list of fields for a record of the data. In a module, we have the opportunity to designate a specific view name to associate with the link type. The view is customized and named when viewing the form or list of records and personalize the forms.

## Module Roles

A module can be restricted with roles just like an application. If this field is left blank, the application's roles will be inherited.

## Additional Queries

Additional queries can be appended to the module, to further define the filter for the returned list. For example, to filter active incidents that are assigned to the currently logged in user, you could use the following as an argument query:

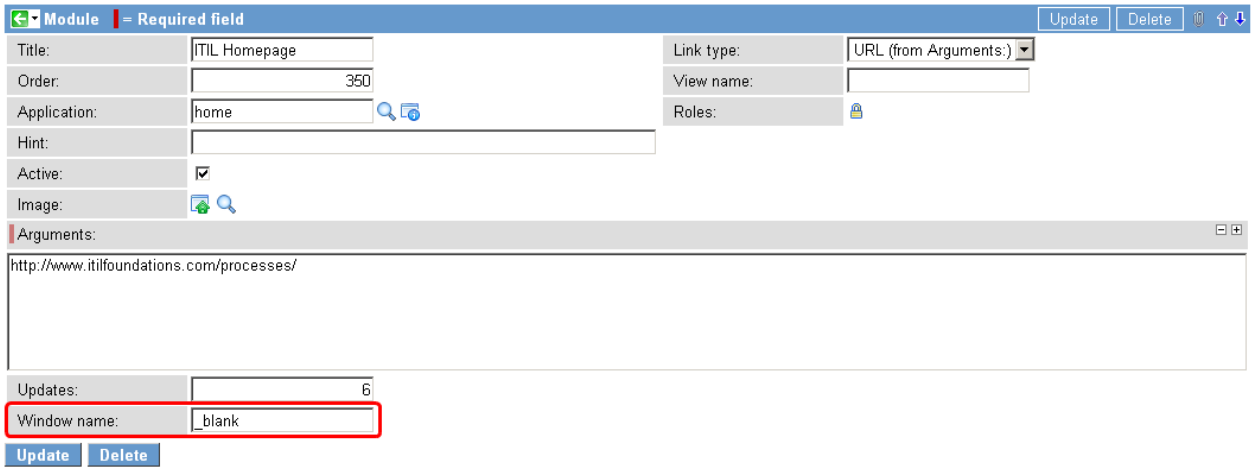
```
{Javascript,N]active=true^assigned_to=javascript:gs.user_id()
```

## Example: URL module that opens in a new window ExternalLink2.png

To configure a module to open an external URL in a new browser window:

1. Select **URL (from Arguments)** from the **Link type** list.
2. Add the complete address to the **Arguments** field.
3. Select the display icon for the module in the **Image** field.
4. If the **Window name** field is not displayed, personalize the form and add this field.
5. Type **_blank** in the **Window name** field.

   If this field is empty, ServiceNow opens the page within the main instance frame (the default behavior).




## Example: URL module that opens an incident list with a custom filter, sort, and *groupby*

A module with a Link type of "URL (from Arguments)" can be configured to create a link to a custom URL.

The best way to figure out what your arguments(URL) should be for this type of module is to go directly to that list and define your filter, sort, and/or *groupby* conditions. These will then be reflected in the browser address bar (since the list isn't being displayed in a frame).

-Navigate to https://demo.service-now.com/incident_list.do
-Define your groupby -Define any filter (This example uses 'active=true' with an ascending sort on 'priority' and a descending sort on 'opened_on'.)

You'll end up with a URL that you can copy from your address bar like this...

```
incident_list.do?sysparm_query=active=true^EQ^ORDERBYpriority^ORDERBYDESCopened_at^GROUPBYassignment_group
```

**Note:** *It is not advised (or necessary) to copy the* [1] *portion of the URL.*

When you have the URL arguments for your module, you can modify your module to be a type of **URL (from Arguments)** and put the piece from your URL into the arguments field. After saving your module and refreshing the left navigation pane, your module should work using the arguments you provided.

## References

[1] http://instance_name.service-now.com/

# Activating ServiceNow Plugins

| Plugins |
| --- |
| • Activating ServiceNow Plugins |
| **Related Topics** |
| • Upgrades and the Release Cycle |

## Overview

Plugins provide additional optional functionality that administrators can activate within a ServiceNow instance. For a list of plugins, see List of Plugins.

## Activating Published Plugins

Most plugins are *published*, meaning that administrators in the global domain can activate any published plugin.

**Instructions for activating a plugin.**

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

   If the plugin depends on other plugins, these plugins and their activation status are listed. Fall 2010 Stable 3.
4. Click **Activate**.

## Requesting Plugins

Some plugins are available only by request. There are two primary reasons why a plugin is only available by request:

- **New Plugins** - When a new plugin is added to the platform, in some cases ServiceNow may want to work directly with the customer in order to implement the plugin.
- **Operational Considerations** - Some plugins may have operational considerations that are only appropriate for certain deployments.

To activate the plugin, make a request to ServiceNow Technical Support using the **Request Plugin Activation** module from HI [1].

## References

[1] https://hi.service-now.com/nav_to.do?uri=com.glideapp.servicecatalog_cat_item_view.do?sysparm_id=891f088e465667e234a3cb52ffa1d299

# Personalizing Forms

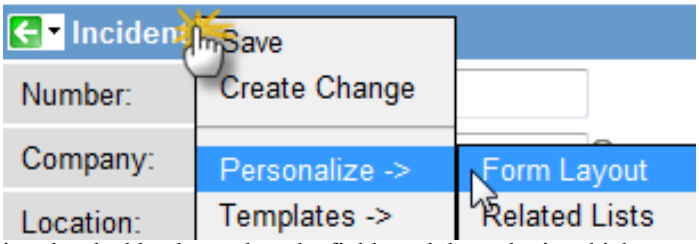
## Overview

Administrators and users with the form_admin role can customize the layout for any form view.

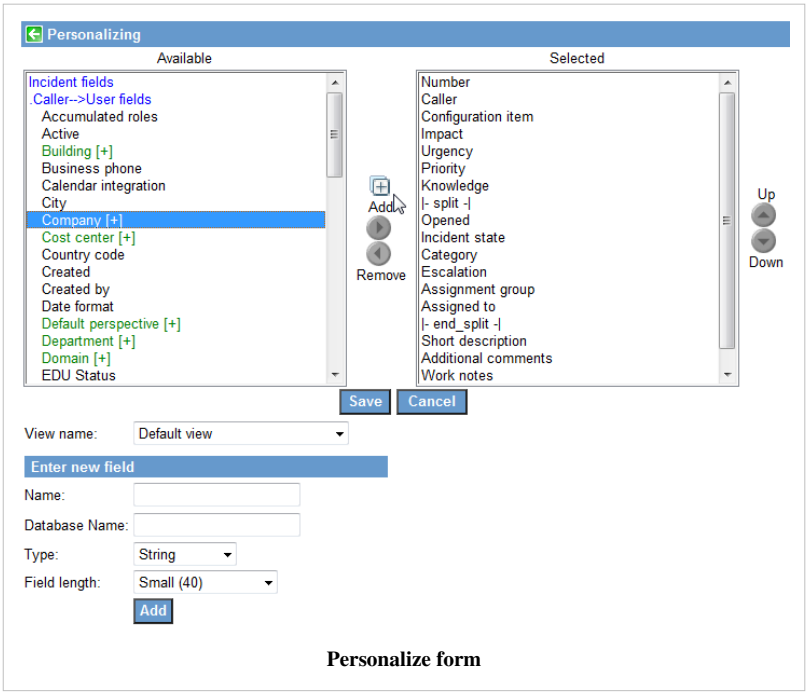
## Personalizing Forms

To personalize a form:

1. Right-click the form header and select **Personalize > Form Layout**.




2. Using the slushbucket, select the fields and the order in which you wish them to appear.
   - Available items that appear in green followed by a plus (+) sign represent related tables. To access fields on these tables, use dot-walking.
3. Click **Save**.
   - Note that although the same field may be added to more than one section on a form, this causes inconsistent behavior and is not recommended.




**Personalize form**

## Configuring Insert for Task Records

The "Save", "Insert", and "Insert and Stay" options are disabled by default for task records (e.g., Incidents, Change Requests).

To configure these options for task records:

1. Navigate to **System Properties > UI Properties**.
2. Locate the property **Allow the use of the "Insert" and "Insert and Stay" options on task derived tables.** (*glide.ui.task.insert* )
3. Select the check box to enable or clear the check box to disable (default) the options for tasks.

# Adding a Related List

Related lists display records in another table that have a relationship with the current record. Administrators can configure related lists to appear on forms and in hierarchical lists.

To add a related list to a form:

1. Open the form.
2. Right-click the header and select **Personalize > Related Lists**.
3. Use the slushbucket to select the related lists that display on the form.

   Related lists appear at the bottom of the form.

# Customer Updates Indicator

The **customer updates indicator** () may appear on the header of forms that have customer updates. Customer updates are changes that are tracked by update sets, such as scripts, service catalog items, and other configuration tables. For the full list, see List of Tracked Customizations.

Clicking the customer updates indicator opens the update set records for the item.

This feature is available with the Fall 2010 Stable 2 release.

## Configuration

Administrators can configure this indicator to appear for all or for specific administrators using the **owned_by_indicator.form** user preference.

To configure the preference, navigate to **User Administration > User Preferences**.

- To enable it for all administrators, ensure the **owned_by_indicator.form** preference is set to *true*.
- To enable it for an individual administrator only, leave the **owned_by_indicator.form** preference set to *false* (default) and then create a new user preference with the following values:
  - **Name**: *owned_by_indicator.form*
  - **User**: <*administrator for which to enable the preference*>
  - **Value**: *true*

# Tutorials

The following video demonstrates personalizing forms. For more e-Learning videos, see Tutorials.

Personalizing Forms Video [1]

## References

[1] http://community.service-now.com/videos/personalize-form-video

# Customizing Homepages

## Overview

Every one of your users can have their own customized homepages that they see when they log in. The information in this wiki page assumes that your organization's ServiceNow administrator has the "Home" application active in your instance. If not, then your default page may be something different.

**Warning:** Because homepages are dynamically generated on each view, loading the homepage with too many gauges may cause performance issues throughout the instance.

Note: The Service Catalog ordering page is essentially also a homepage. It shares the same functionality (e.g., it can be protected with roles), and the information below applies to it as well.

ServiceNow supports editor links on your homepage content that, when clicked, take you to the definition for the particular piece of content you clicked on.

Since the content on your homepage comes from a variety of sources, the target of the editor link is also going to vary based on what you click on. For example:

1. If you click on a graph or chart, the system will take you to the gauge or report that generated that content.
2. If you click on an application module, the system will take you to the application in question.
3. If you click on a catalog section, the system will take you to the catalog category that rendered the content in question.
4. If you click on the scrolling news widget, the system will take you to the "News" knowledge category, since that is where the contents of the scroller are loaded from.

## Viewing your Homepage

When a user logs into ServiceNow, they'll be taken immediately to their homepage. If the user has a personalized homepage, that's what they'll see. If they don't, they'll see the homepage for their role (admin, itil, etc.). If their role has no homepage, they'll see the default blank "My home" page that they can customize. To default a different homepage based on role change the order.

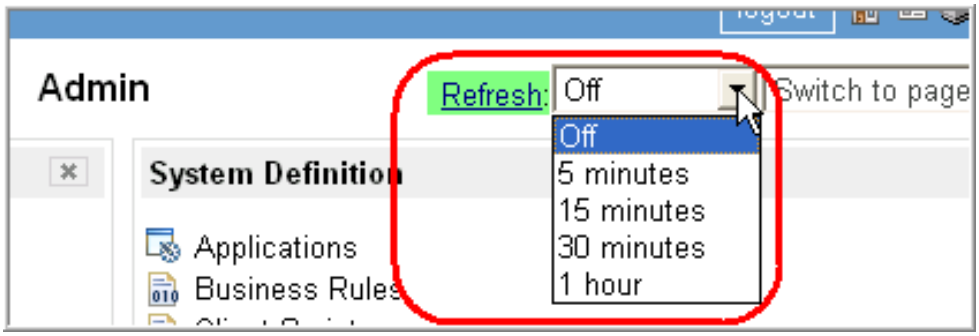
- A lower Order value means it will be preferred before the others of equivalent role
- If no Order value is present, the pages get preference alphabetically

**Note:** *A user sees the homepage with the lowest **Order** value (of the pages they have roles to see) when they log in.*

After you've logged in, you can return to your homepage at any time simply by clicking the icon in the upper-right corner of your browser window near the Print and Help icons.

### Homepage Refresh

Homepage users can specify a refresh time of 5, 15, 30, or 60 minutes, or no refresh. The default is no refresh ("off"). Also, at any time, clicking the "Refresh" link on a homepage will do just that.




### Number of Records Displayed in List gauges

The number of records displayed in a list gauge on your homepage is limited to the number of records displayed in your lists. e.g., if you display 100 records in your lists (this is a user preference set from any list using the "number per page" select box), you will see a max of 100 records in your list gauge.

## Customizing Default Homepages

If you are System Administrator in ServiceNow

From the left navigation pane, select the **Homepage Admin** application, and then select the homepage you want to customize (the one for ESS "Self-Service" users, or the one for ITIL workers, etc.).

Customize the content of the page using the instructions below.

## Customizing your Homepage

### Moving Homepage Items

Repositioning elements on your homepage is as easy as dragging and dropping them. In fact, that's how you do it. Grab the header of a homepage element, and you can click-and-drag it on the page to where you want it positioned.

### Removing Homepage Items

To remove a homepage item, click the [X] on the right side of the item's header.

### Adding Homepage Items

To add an item to your homepage, go to your homepage and click the "Add content" button at the top. A window will open that lets you select from a number of different homepage items you can add.

**Warning:** Each gauge on your homepage is the equivalent of running a report. e.g., a homepage with 10 gauges runs 10 separate reports each time it refreshes. Keep this in mind when adding lots of content to your homepage. If your homepage consistently loads slowly, try removing gauges to determine why.

- **Dashboard Gauges -** lets you select from all available dashboard gauges. Gauges are defined in "System UI -> Guages" and can easily be set to link to an existing report. In this way it is possible to easily add specific content to a homepage.
  - Create a report that contains the data that you would like to be shown

    - You will need to use something like "Assigned to" "is" "javascript:gs.getUserID()" to personalize for the user whom the homepage belongs to
    - Create a gauge that links to your report using "Create Gauge" button.
  - Modify the homepage to include the newly created Gauge.
    - Navigate to the homepage as an admin user by clicking "ESS Homepage"
    - Select add content and attach the gauge (note that they are sorted by the table that the report is for)
- **Gadgets -** offers some informational items for your homepage
- **Knowledge Base -** lets you select a knowledge base category to display
- **Labels -** lets you choose from any label  set up in your instance
- **Scrollers -** offers a news scroller
- **Service Catalog -** lets you select a service catalog category to display
- **System Applications -** lets you display an application and its modules from the navigator

For more information, view Adding Existing Gauges to a Homepage

### Administration

Administrators should see the article Homepage Administration for information on how to administer homepages.

# Service Level Agreements

## Overview

A **Service Level Agreement**, or **SLA**, is a record which defines a set amount of time for a task to reach a certain condition. If the task does not reach the condition by a set amount of time, it is marked **Breached.**

SLAs are used to ensure that a task reaches end conditions within a certain amount of time, such as ensuring that an incident is closed or resolved within a few business days. The success rate can be reported on, and actions can be triggered at different times during the SLA's life-cycle (e.g. notifying the manager when the SLA reaches 75% of its allotted time).

### OLAs and Underpinning Contracts

The Task SLA engine provided by the plugin can also be used to define OLAs or Underpinning Contracts in exactly the same way as SLAs. The only difference between SLAs, OLAs, and Underpinning Contracts is the **Type** field on the Task SLA form. Changing the type field does not change the behavior of the Task SLA.

For an example of an OLA, see Defining an OLA for Catalog Fulfillment.

## New Improvements

### Aspen Release

The property **Compute prior SLA pause time for new, retroactive SLAs (2011 SLA engine only)** is available. When enabled, this property calculates the pause time when a retroactive SLA is attached.

For example: if a retroactive SLA attaches to an incident one hour after its creation, and meets the pause conditions for half an hour, then the elapsed time is half an hour rather than the full hour.

**Note:** *This property is only used with audited tables. Tables which are not audited ignore the pause time before the creation of the record.*

## June 2011 Preview 2

The June 2011 Preview 2 release provides tools for controlling how an SLA Definition's conditions are combined to control the different transitions. New **Condition Rules** provide the ability to script that behavior.

For instances implemented before Preview 2, see Getting Started.

## Fall 2010

**Note**: If the OOB business rules for the SLA plugin have been modified, please contact customer support **prior** to moving to the Fall 2010 release, as the upgrade includes several changes to the SLA engine that may not be able to fully operate with the customized business rules.

Changes made for Fall 2010:

- **Business Rule changes** - all of the SLA business rules are consolidated into a single, async rule called **Process SLAs**. The old Business Rules are deactivated. The incident form will no longer require the SLA calculation to complete before the form is rendered. The SLAs will process in the background within 10 seconds of the form load.
- **Calculation Script changes** - the calculation for the time fields have all been consolidated into a single script include that is now called from everywhere. The calculations now also include a new **business pause time** field, making the calculations more accurate.
- **Scheduled Job changes** - the single scheduled job that processes the SLAs every 2 min has been split up into 6 jobs, processing at closer and closer intervals as the SLAs get closer to breaching, reducing load on the system.

### Checking Update to SLA Business Rules

To check whether the OOB SLA business rules have been modified (which may prevent upgrades), run the following script in **System Definition > Scripts - Background**:

```
var count = 0; var ux = new GlideRecord('sys_update_xml'); ux.addQuery('update_set.name', '!=', 'Baseline'); ux.addQuery('name', 'STARTSWITH', 'sys_script'); ux.addQuery('payload', 'CONTAINS', 'task_sla'); ux.query(); while (ux.next()) {
  count ++;
  gs.log('Conflict found ' + ux.name);
}
gs.log('Total conflicts = ' + count);
```

If no conflicts are returned, the upgrade to Fall 2010 can be performed.

# Video Demo (11 minutes)

Here is a short video showcasing the functionality described below:

# SLA Components

There are four major components that work together to power the Service Level Agreements Plugin:

- **SLA Definition** - the record which defines the conditions that trigger the SLA.
- **Task SLA** - the individual instances of the SLAs associated with particular tasks.
- **SLA Workflow** - the workflow which powers events or actions based on the SLA definition.
- **SLA Automation** - the business rule and scheduled job that automate the SLA.
- **SLA Conditions and Script Include** - *June 2011 Preview 2* - a script include and reference record that can be used to customize the transitions between different SLA states.

## SLA Definition

The **SLA [contract_sla]** table contains the definitions of SLAs, which determine when and how they track time on tasks.

SLAs are defined by three sets of conditions:

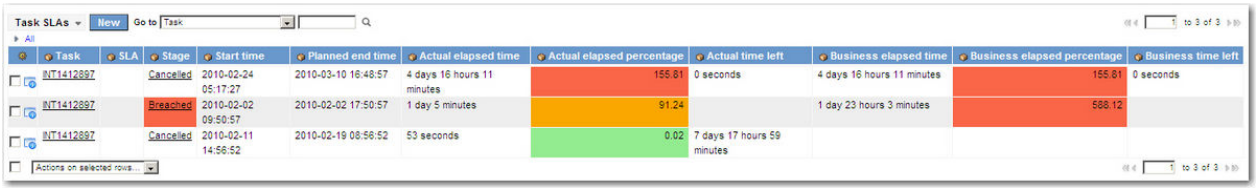
- Start Conditions
- Pause Conditions
- End Conditions

When the start conditions are met, the SLA begins timing towards the defined duration. When the pause conditions are met, the SLA pauses its timing. When the end conditions are met, the SLA stops timing and determines whether the SLA was **Achieved** or **Breached**.

To see the conditions in action, see Demo Walkthrough. For more information on defining SLAs, see Defining an SLA.

## Task SLA

The **Task SLA [task_sla]** table stores each of the individual SLAs attached to particular tasks. These are accessible in a related list on the Task's form.



### Time-Based Fields

The six time-based fields on the Task SLA contain the crucial information powered by the Task SLA:

- **Actual Elapsed Time** - Time between start time and now (minus pause duration).
- **Actual Elapsed Percentage** - Percentage of total SLA that has elapsed (minus pause duration).
- **Actual Time Left** - Time remaining until SLA breach.
- **Business Elapsed Time** - Time within the specified schedule between start time and now (minus pause duration).
- **Business Elapsed Percentage** - Percentage of total SLA that has elapsed within the specified schedule (minus pause duration).
- **Business Time Left** - Time within the schedule remaining until SLA breach.

These fields are dynamically populated by a scheduled job.

## SLA Workflow

The SLA Workflow is a workflow defined in the Graphical Workflow Editor which powers behavior in response to the SLA. This allows email notifications, script actions, and any other workflow activities to be triggered at various stages in the SLA.

For more information, see Creating an SLA Workflow.

## SLA Automation

SLAs are calculated and assessed by a business rule and a scheduled job which run in the background.

### Business Rule

The asynchronous Business Rule **Process SLAs** runs after every task is inserted or modified and evaluates the **Start**, **Pause**, and **End** conditions for the SLA.

### Scheduled Jobs

SLA calculations now get updated based on when they are breached. These happened by the following Schedule Jobs (on the **sys_trigger** table):

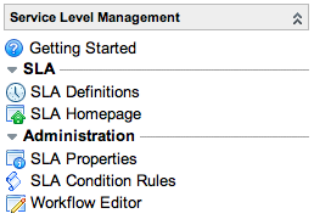
- **SLA update (already breached)** - repeats every day
- **SLA update (breach after 30 days)** - repeats every 5 days
- **SLA update (breach within 1 day)** - repeats every hour
- **SLA update (breach within 1 hour)** - repeats every 10 minutes
- **SLA update (breach within 10 min)** - repeats every 1 minute
- **SLA update (breach within 30 days)** - repeats every day

## SLA Conditions

Introduced with June 2011 Preview 2, the SLA Condition record references a Script Include to provide scriptable condition rules for the transitions between different SLA states. By defining methods as part of an SLAConditionBase sub-class, it is possible to script under what conditions an SLA is marked *Paused*, *Completed*, *Cancelled*, and when the SLA is *Attached* or *Reattached*.

For more information, see Modifying SLA Condition Rules.

# Applications and Modules

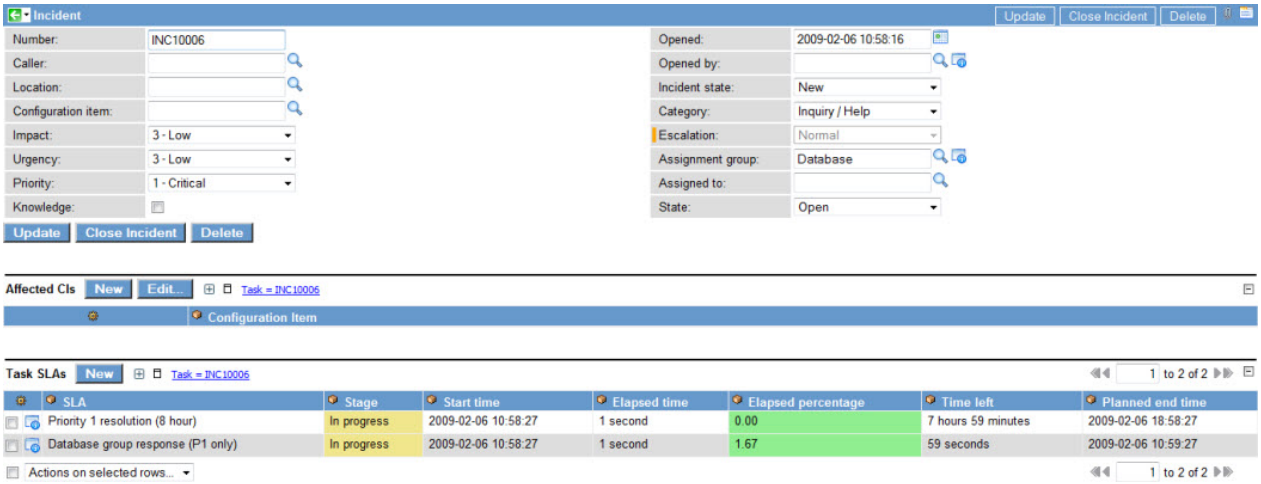



- **Getting Started** - Links to this document on the wiki.
- SLA
  - **SLA Definitions** - The **SLA [contract_sla]** list.
  - **SLA Homepage** - A Service Level Management homepage
- Administration
  - **SLA Properties** - The properties page for Service Levels.
  - **SLA Condition Rules** - The list of Condition Rules for SLAs.
  - **Workflow Editor** - The Graphical Workflow Editor.

# Demo Walkthrough

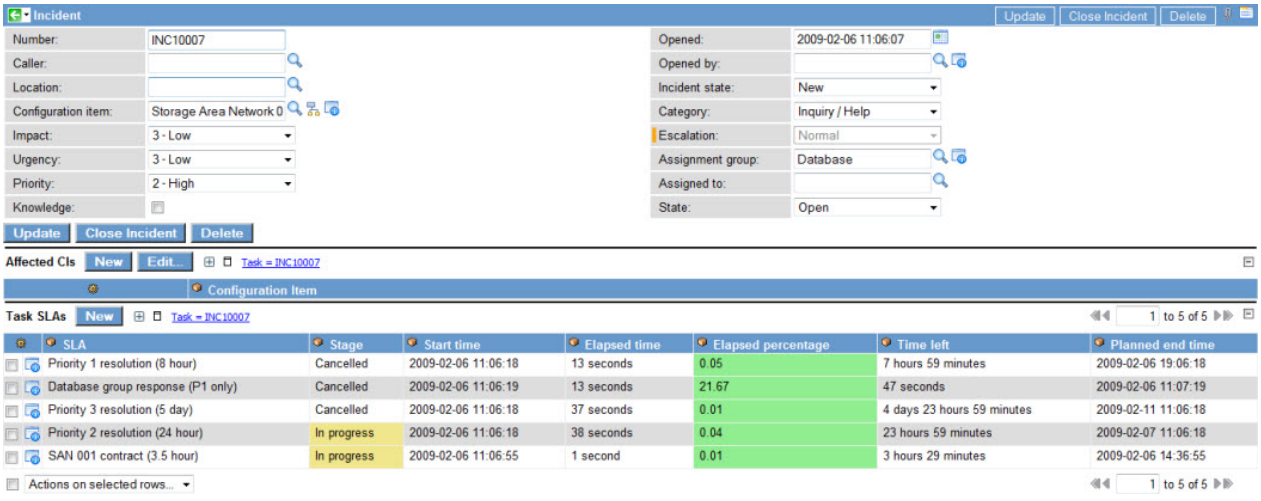
The following walkthrough uses the demo SLAs available with the plugin to illustrate an SLA being achieved, breached, and paused on the **Incident [incident]** table.

1. Navigate to **Incident > Create New**.
2. Set the **Priority** to **1 - Urgent**.
3. Save and then Reload Form.
4. The Related List **Task SLA** should now have two SLA's:



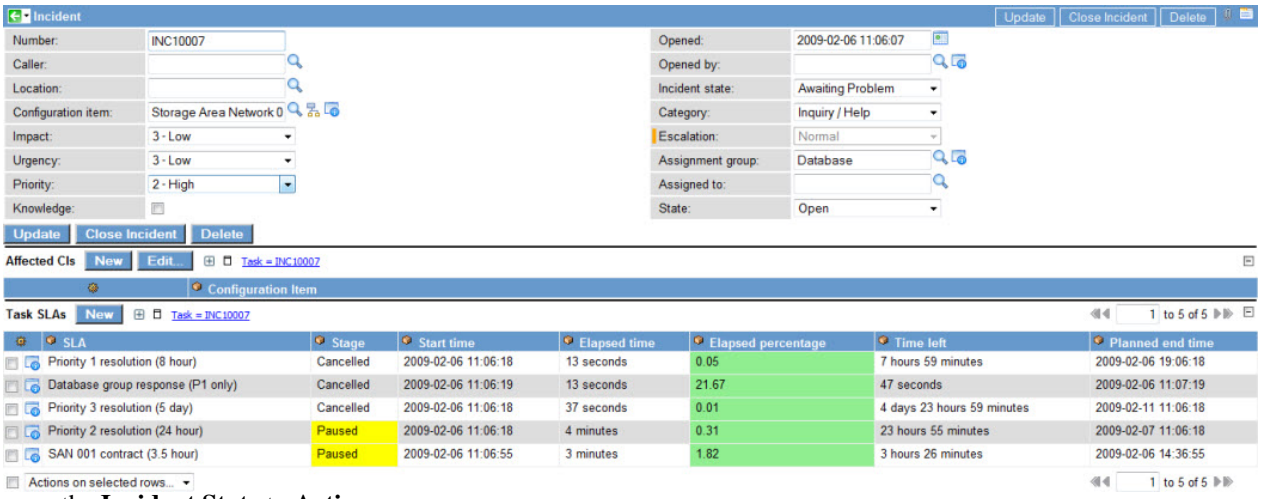

5. Change the priority to **2 - High**.
6. Save and then Reload Form.

   The two Priority 1 SLAs are now marked **Cancelled** and a Priority 2 SLA has been attached, because of the conditions on the SLAs:



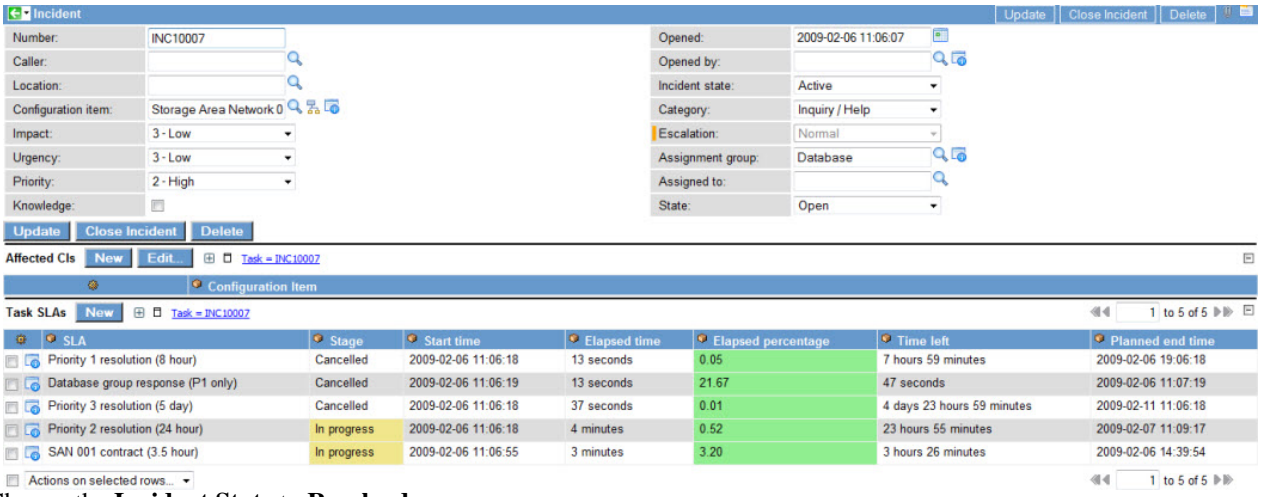
7. Change the **Incident State** to **Awaiting User Info**.
8. Save and then Reload Form.

   **Awaiting User Info** is a **Pause** condition on the Priority 2 SLA, so the SLA will be marked **Paused**:



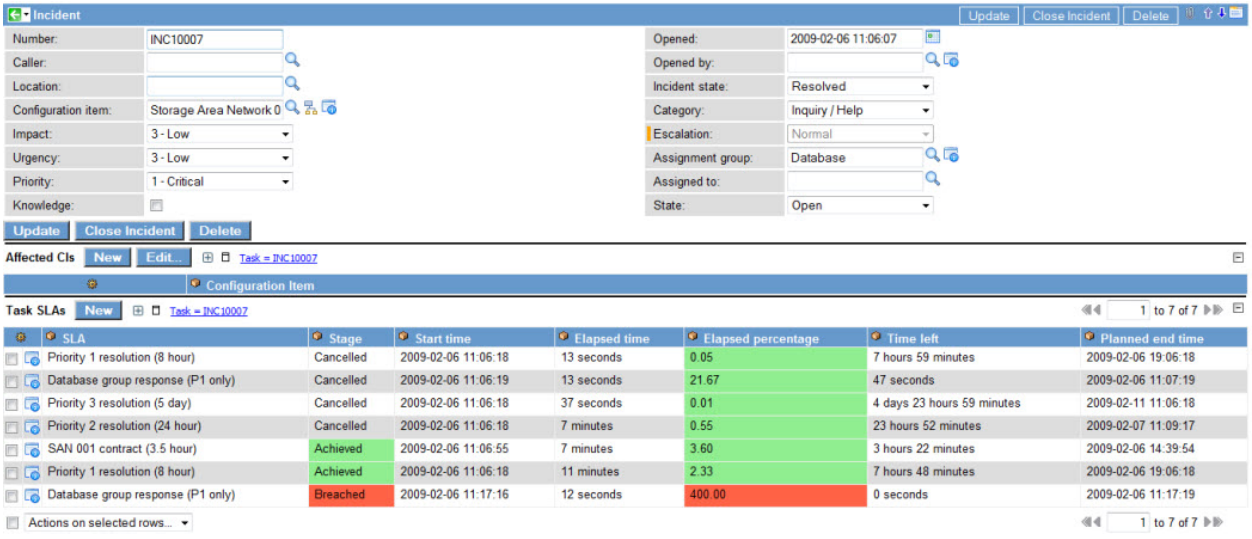

9. Change the **Incident State** to **Active**.
10. Save and then Reload Form.

    Because the Incident is no longer in a pause condition, it will resume timing:




11. Change the **Incident State** to **Resolved**.
12. Save and then Reload Form.

    The SLA will now be marked either **Achieved** or **Breached, depending on how long has elapsed:**



# Installing the Service Level Agreements (SLA) Plugin

## Activating the Plugin

The Service Level Agreements (SLA) Plugin is installed by default for all new instances. Older instances will need to activate the plugin.

**Instructions for activating a plugin.**

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

   If the plugin depends on other plugins, these plugins and their activation status are listed. Fall 2010 Stable 3.
4. Click **Activate**.

## Getting Started

For information on implementing the plugin on instances that have been using the out-of-box SLA engine, see Getting Started with the SLA Plugin.

# Creating an SLA Workflow

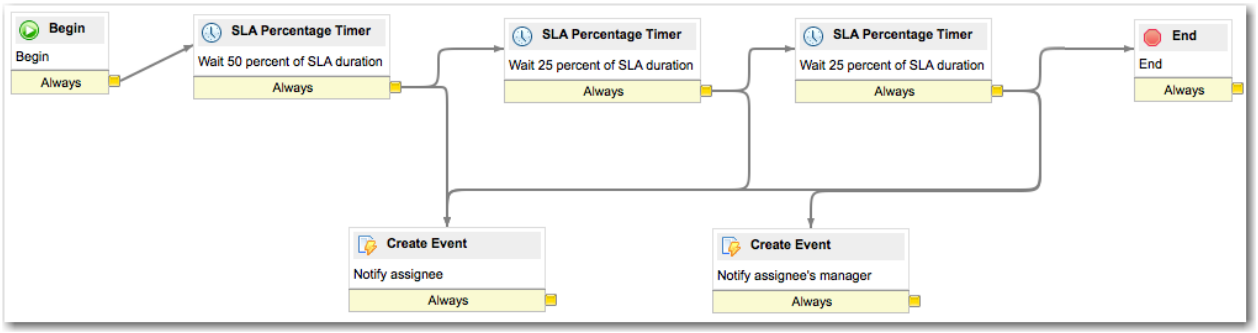
Functionality described here requires the **Admin** role.

Functionality described here requires the **Service Level Agreements (SLA) Plugin** plugin.

## Overview

Once an SLA is defined, it requires a workflow to enforce its definition. The workflow is built using the Graphical Workflow Editor, and is set to run on the **task_sla** table.

The out-of-box workflow that comes with the plugin is **Default SLA Workflow**:



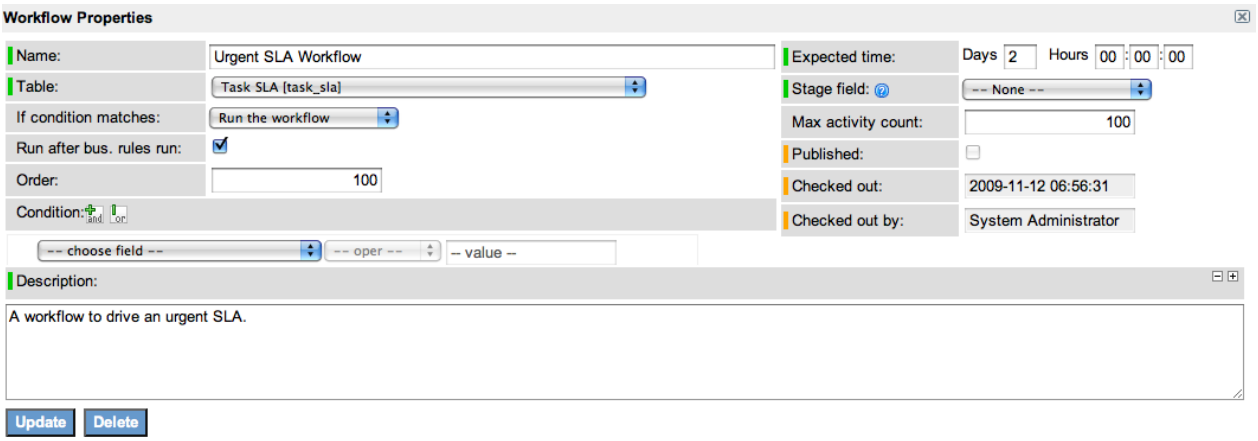

This example creates a new **Urgent SLA Workflow** to be used for a priority one ticket.

# Creating an SLA Workflow

To create an SLA workflow, navigate to **Service Level Management > Workflow Editor**. This launches the Graphical Workflow Editor. Select **New** in the top left, and populate the first form as follows:

- **Name** - **Urgent SLA Workflow**.
- **Table** - **Task SLA [task_sla]**. All SLA Workflows must be on the **Task SLA** table regardless of what table the SLA will be running against.
- **If Condition Matches:** - **--None--**. Because no conditions will be specified since it is set to none, the SLA will not run unless specified by a particular SLA.

The form should now look like this:



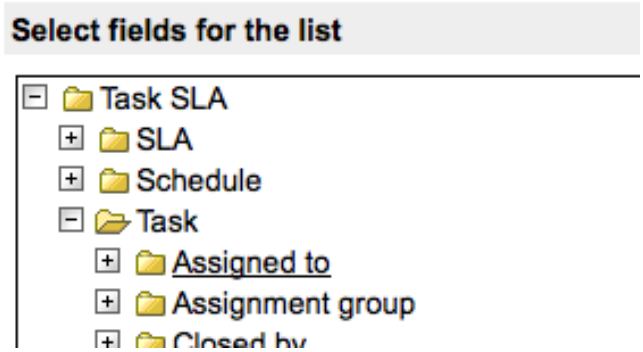

Click **Update**. There is now a workflow with a beginning and end. Now it is time to specify what will happen when the **Urgent SLA Workflow** is triggered.

Suppose the desired process is:

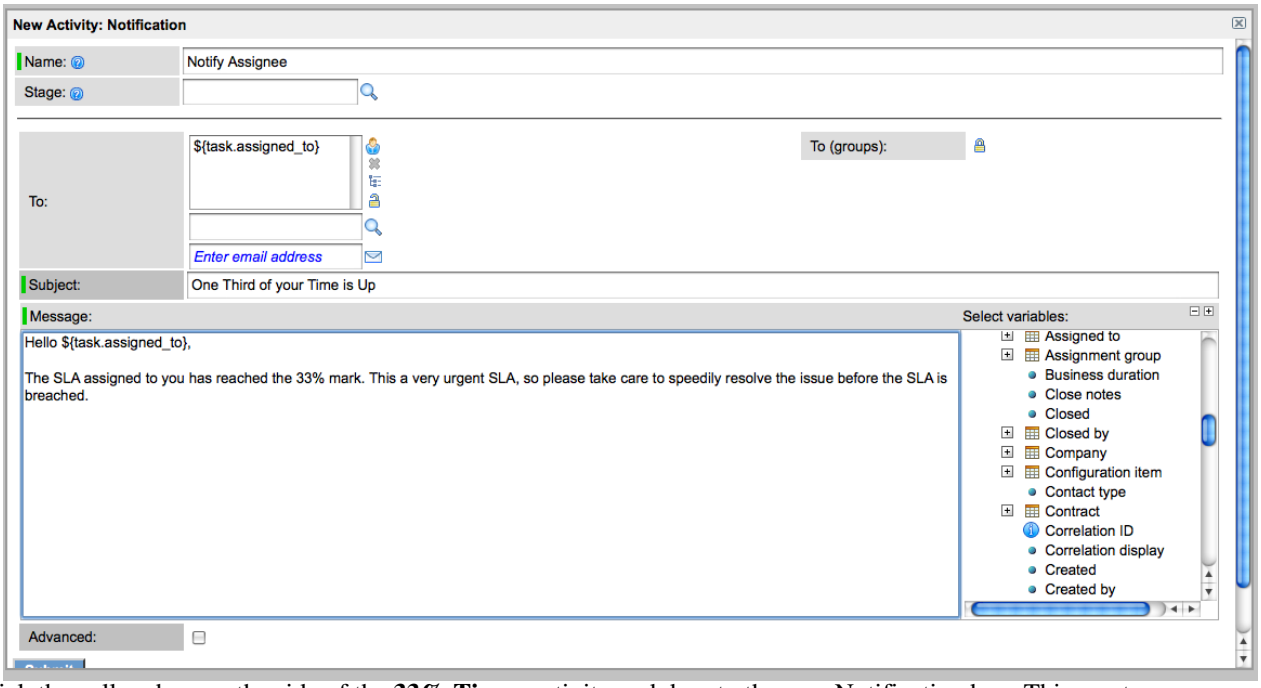
- After 33% of the SLA's duration, alert the user assigned to responding to the task.
- After 66% of the SLA's duration, notify the assignee's manager.
- After 100% of the SLA's duration, notify the Customer Care representative that the SLA has been breached.

To define that process as a workflow:

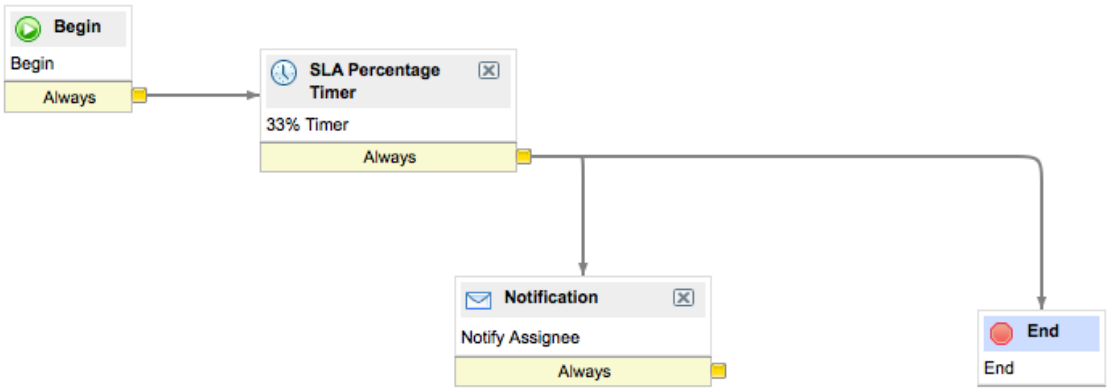
1. Drag the activity **SLA Percentage Timer** onto the arrow between **Begin** and **End**. Name it **33% Timer** and place **33** in the **Percentage** field.
2. Drag the activity **Notification** into the empty space below the **33% Timer** Activity. Name the activity **Notify Assignee**. Click the lock on the **To** Field, select the variable picker (), and select **Task SLA > Task > Assigned To.**




Fill in a subject and email body as necessary. The form should look as follows:

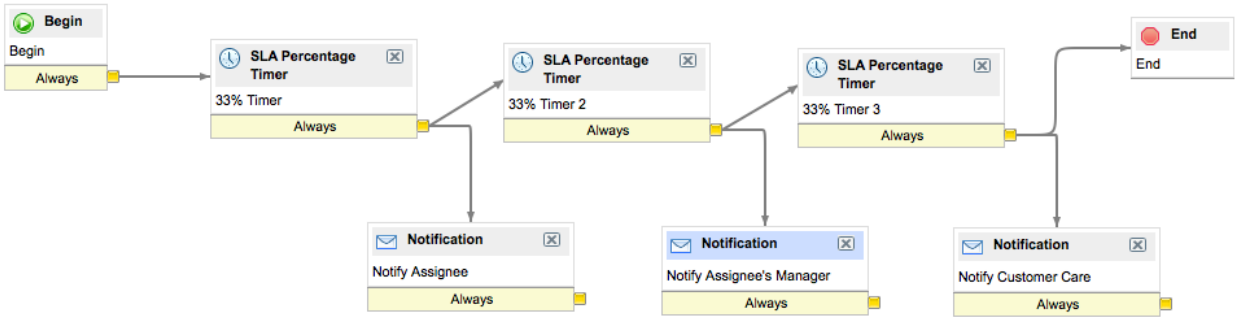



3. Click the yellow box on the side of the **33% Timer** activity and drag to the new Notification box. This creates a second arrow from **33% Timer** to the new Notification:




4. Drag the activity **SLA Percentage Timer** onto the arrow between **33% Timer** and **End**. Name it **33% Timer 2** and place **33** in the **Percentage** field.
5. Drag the activity **Notification** into the empty space below **33% Timer 2** Activity. Name the activity **Notify Assignee's Manager**, unlock the **To** field and select **Task SLA > Task > Assigned To > Manager.** Click Submit. Drag an arrow from **33% Timer 2** to the new Notification.
6. Drag the activity **SLA Percentage Timer** onto the arrow between **33% Timer 2** and **End**. Name it **33% Timer 3** and place **33** in the **Percentage** field.
7. Drag the activity **Notification** into the empty space below **33% Timer 2** Activity. Name the activity **Notify Customer Care**, unlock the **To** field, and type customercare@yourcompany.com into the field marked *Enter Email Address*. Click Submit. Drag an arrow from **33% Timer 3** to the new Notification.

   The workflow should now look like this:




8. Click the Workflow Actions icon and select **Publish**.

The workflow will now be available to any SLA, and can be edited at any time by opening the workflow editor, opening the workflow, and checking it out.

# Service Catalog Administration

## Overview

An administrator has the capability to configure the Service Catalog. Below are a few common ways that administrators can administer the catalog.

## Restricting Items and Categories By Company and Department

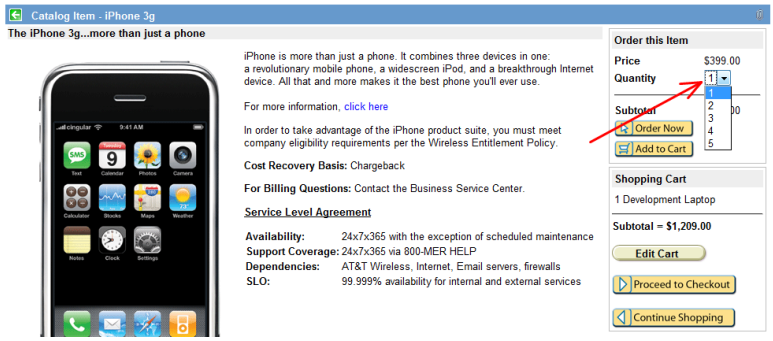
To restrict items and categories by company and department:

1. Navigate to **Service Catalog > Maintain Items** or **Service Catalog > Maintain Categories**.
2. Open an item or category, and then personalize the related lists in the form to add **Available for Department** and **Available for Company**.
3. Click **Update** to save the form layout.
4. Open the item or category you wish to secure and add companies and departments to related lists.

The item or category is available only for the companies and departments listed. If no companies or departments are listed, then the item or category is available to all companies or departments.

## Item Quantity

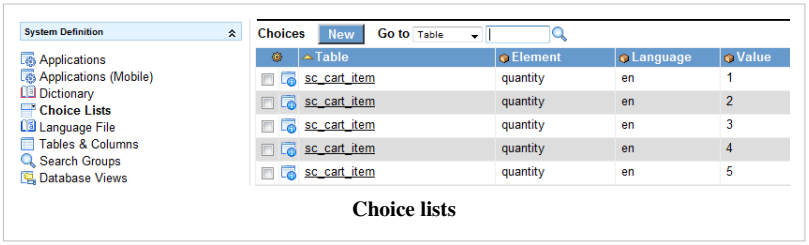
In the Service Catalog, the default quantity choices are 1 to 5. Administrators can configure the quantity selector with additional choices.




**Quantity selector**

To configure the quantity selector:

1. Navigate to **System Definition > Choice Lists**.
2. Search for the table *sc_cart_item* and the element *quantity*. The existing quantity choices appear.
3. Add quantity choices, modeling them after the existing ones.




**Choice lists**

# Service Catalog Variables

## Overview

During the course of executing a workflow or execution plan, the various groups involved in the process might very well have a need to pass information back and forth in a structured form. For example, consider this execution plan for a hypothetical "New PC" catalog item.

1. Procure PC
2. Install Corporate Standard Software
3. Set up email account
4. Deliver PC to requester

Step 1-3 are fairly straightforward, but step 4 requires a piece of information from step 3. After all, when you deliver the PC to an end user, the user needs to know what his email account credentials are. If step 3 and step 4 are executed by different groups of people though, there could be a problem.

That's where variables come in. Variables are structured pieces of information that are defined in a workflow or execution plan. For example, an execution plan for "a New PC" might have these variables defined:

- Email software
- Email account user name
- Email account password

## Variable Location

Variables can be defined either at the workflow or execution plan level or at the catalog item level. A variable defined at a workflow or execution plan level is available for any requests items which use this workflow or execution plan. Conversely a variable defined on a catalog item is available in any request for this item, regardless of workflow or execution plan.

## Associating Variables with Tasks

Each workflow or execution plan can have one or more tasks. When a request is made for a particular item that invokes this workflow or execution plan, still more variables might be created if the catalog item has its own set of variables. The net result can be a sizable number of variables associated with the request item, not all of which should be visible within each and every catalog task.

### Global Variables

By default, a catalog task can see only the Global variables defined in its request item and the workflow or execution plan. You can attach variables to a specific task by navigating to the task and clicking **Edit** on the list of available variables. You may not need to do anything at all, since Service Catalog variables are flagged as Global by default and appear in all catalog tasks, regardless of whether or not you explicitly attached them. If you want to protect a particular variable and/or limit its task visibility, clear the global flag.

# Service Catalog Workflows

Functionality described here requires the **Admin** role.

## Overview

Request fulfillment can be driven by the Graphical Workflow Editor. Using the intuitive, visual editor, the administrator can create reusable workflows to drive complex fulfillment processes. Workflows can generate and assign approvals, generate and assign tasks, or run scripts and subflows.

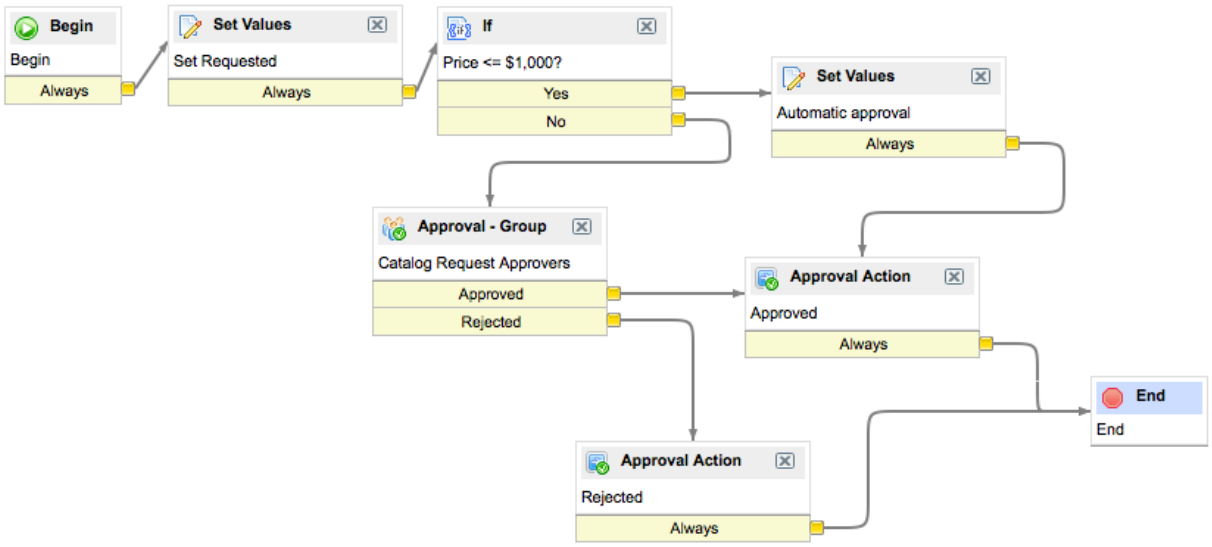
A Service Catalog Workflow is a workflow on the table **sc_request**. It can be attached to particular requests in one of three ways:

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

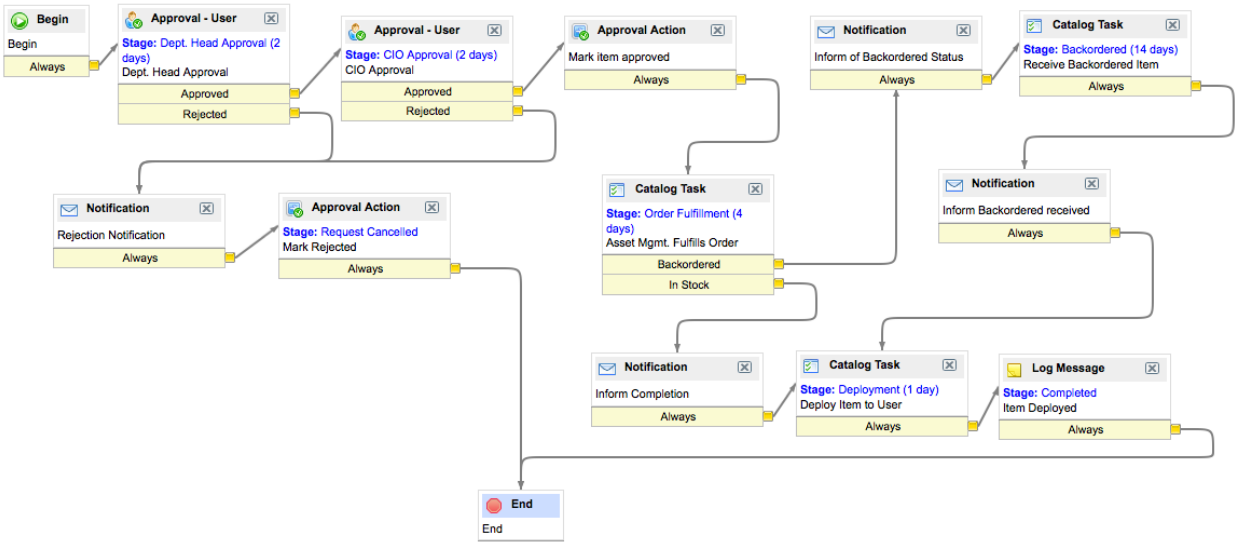
# Out-of-Box Service Catalog Workflows

There are two Service Catalog Workflows provided out-of-box that can be instructive examples of how the workflow engine can work for service catalog. One is a more simple Catalog request, and the other is a more complex example of a Catalog Item request.

## Catalog Request




## Catalog Item Request



# Business Rules

**Note:** This functionality requires a knowledge of **Javascript**.

## Overview

A business rule is a piece of JavaScript configured to run when a record is displayed, inserted, updated, deleted, or when a table is queried. A business rule can be set to run before or after the database action has occurred. In the case of a query, the business rule runs before the database operation, so that the data returned to the user is appropriate to his system privileges (user roles). A typical business rule might execute a script after a user updates an incident or escalates the priority of a change request. Use a business rules to create new events for email notification and script actions.

## Recent Improvements

The following enhancements have been made to Business Rules.

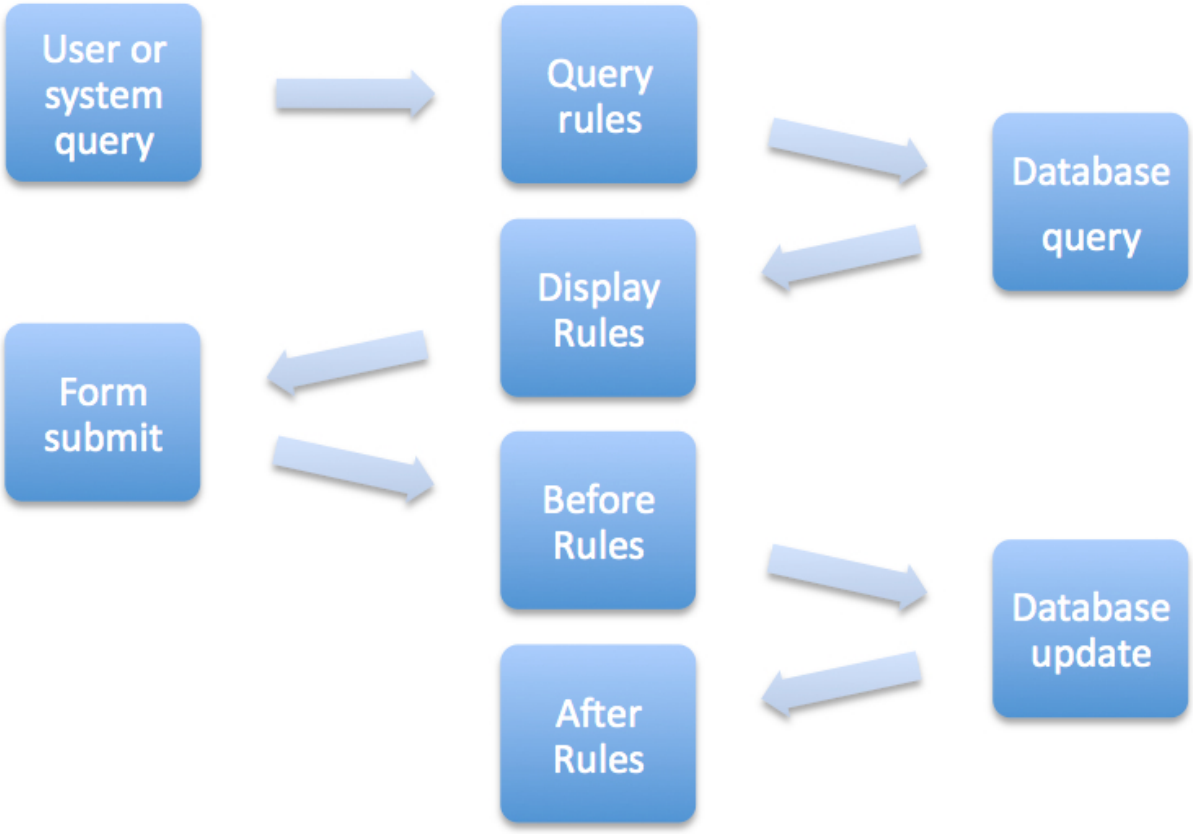
### Fall 2010 Stable 2

Fall 2010 Stable 2 release introduced a new display type of rule. These rules will run just before a form is displayed to the user.

### Aspen Release

In the Aspen release, two Async Delete Warning client scripts use `showFieldMsg()` instead of `addInfoMessage()` when warning against using **Current** in an async business rule.

## Business Rule Process Flow




Note that business rules apply consistently to records regardless of how they are accessed--through forms, lists, or Web Services. This is one major difference between Business Rules and Client Scripts, which only apply when editing through the form.

## Business Rules Form

This example is the business rule used for saving updates made to the *change_phase* table. The rule is configured to execute the JavaScript code after an insert or update is made on the *change_phase* table.

The Business Rules form provides the following fields.

| Field | Input Value |
|---|---|
| Name | A descriptive name for your business rule. |
| Table | Select the appropriate database table for this business rule. |
| Order | Type the sequence in which this business rule should run. If there are multiple rules on a particular activity, then the rules will run in the order specified here, from lowest to highest. |
| Active | Select the check box (*true*) to enable this business rule. |
| When | Select when this business rule should execute: **display**, **'before**, **async**(queued), or **after** the database operation is complete. |
| Insert | Select this checkbox to execute the business rule when a record is inserted into the database. |
| Update | Select this checkbox to execute the business rule when a record is updated. |
| Delete | Select this checkbox to execute the business rule when a record is deleted from the database. |
| Query | Select this checkbox to execute the business rule when a table is queried. |

| | |
|---|---|
| Condition | Create a statement for a condition under which the business rule should execute. By adding the condition statement to this field, you tell ServiceNow to evaluate the condition separately and parse the script only if the condition is true. If you decide to include the condition statement in the script, leave this field blank. |
| Script | Create a script that triggers your new event when the condition you define is true. |

## Comparing Versions

To track changes to this record:

1. Select the check boxes of two versions in the **Versions** related list.

   The current version is the top version in the list.

   **Note**: *This related list is available in the October 2011 release*.
2. Expand the Actions menu at the bottom of the list and select **Compare**.

   A page appears comparing elements of the selected versions. This page does not contain any controls.

# Business Rule Scripting

You create scripts in business rules with JavaScript. The following are the predefined variables that help reference the system:

- **current**: The current record being referenced
- **previous**: The record before any changes were made, available on update and delete operations
- **g_scratchpad**: Fall 2010 Stable 2 - scratchpad object available on display rules, used to pass information to the client to be accessed from client scripts
- **system (or gs)**: References to GlideSystem functions. See GlideSystem.

## Business Rule Variables are Global

System provided variables, such as *current, previous* and "g_scratchpad" are global across all business rules that run for a transaction. User created variables are also globally scoped by default. This means that if a new variable is declared in an order 100 business rule, the business rule that runs next at order 200 will also have access to the variable. This may introduce unexpected behavior.

To prevent such unexpected behavior, *it is recommended that you always wrap your code in a function*. This protects your variables from conflicting with system variables or global variables in other business rules not wrapped in a function.

This is an example of a script that is vulnerable to conflicts with other code. If the variable "gr" is used in other rules the value of the variable may unexpectedly change.

```
var gr = new GlideRecord('incident');
gr.query();
while (gr.next()) {

  //do something

}
```

When this script is wrapped in a function, the variable is only available within the function and will not conflict with other functions using a variable named "gr".

```
myFunction();

function myFunction() {
  var gr = new GlideRecord('incident');
  gr.query();
  while (gr.next()) {
    //do something
  }
}
```

### Aborting a Database Action in a Business Rule

During a *Before* Business Rule script you can cancel, or abort, the current database action by using the current.setAbortAction(true) [1] method. For example, if your Business Rule is executed with a setting of "When: Before" during an Action of "Insert", and you have a condition in the script that, when met, it calls: current.setAbortAction(true), then the new record stored in "current" will not be created in the database.

## Global Business Rules

Business Rules marked as running on table **Global** are loaded and initialized at the beginning of each interaction between a user and the platform. Global business rules ignore the **Condition** field. Therefore, unless the script is wrapped in a function, it will run on every interaction between the user and the platform.

Global Business Rules that have script wrapped in a function can be called upon by any script running elsewhere. Client scripts and business rules can reference any function defined in a global business rule. Global business rules are therefore useful in defining functions to be used repeatedly by a variety of scripts.

## Before vs. After Business Rules and *current.update()*

The function *current.update()* is unnecessary in a business rule that executes before the database activity. Use of this function causes double updates and therefore double events. All changes to the current record should be made in *Before* business rules, which then lets the system update the record. *After* business rules should only react to the update to create events, update related records (like the incidents related to an updated change), etc., and therefore should not use *current.update();*.

## Display Business Rules

Introduced with Fall 2010 Stable 2, display rules are processed when a user requests a record form. The data is read from the database, display rules are executed, then the form is presented to the user. The "current" object is available and represents the record retrieved from the database. It is possible to change values at this point (current.short_description = 'new description'). Any field changes are temporary since they are not yet submitted to the database. To the client, the form values appear to be the values from the database, there is no indication that the values were modified from a display rule. This is a similar concept as calculated fields.

The primary objective of display rules are to utilize a shared scratchpad object, "g_scratchpad", which is also sent to the client as part of the form. This can be useful when you need to build client scripts that require server data that is not typically part of the record being displayed. In most cases this would require a client script making a call back to the server. If the data can be determined prior to the form being displayed, it will be more efficient to provide the data to the client on the initial load. The form scratchpad object is an empty object by default, and only used to store name:value pairs of data.

To populate the form scratchpad with data from a display rule:

```
//from display business rule
g_scratchpad.someName = "someValue";
g_scratchpad.anotherName = "anotherValue";

//if you want the client to have access to record fields not being
displayed on the form
g_scratchpad.created_by = current.sys_created_by;

//these are simple examples, in most cases you'll probably perform some
 other queries to test or get data
```

To access the form scratchpad data from a client script:

```
//from client script
if (g_scratchpad.someName == "someValue") {
  //do something special
}
```

## Determining the operation triggering the Business Rule

Sometimes you will be scripting for a business rule that is triggered on more than one action (eg. Insert, Update, Delete, Query). If you want your business rule script to dynamically branch depending on the action that triggered the event, you can use the "operation()" function. See the sample code below:

```
if( current.operation() == "update" ) {
 current.updates ++;
] if( current.operation() == "insert") {
 current.updates = 0;
}
```

## Example: Lock Accounts That Don't Meet Role or Active Requirements

```
// Lock accounts if bcNetIDStatus != active in LDAP and user does not
// have self-service, itil or admin role
var rls = current.accumulated_roles.toString();
if ( current.u_bcnetidstatus == 'active' &&
   ( rls.indexOf(',itil,')  > 0 ||
     rls.indexOf(',admin,') > 0 ||
     rls.indexOf(',ess,') > 0 ) ) {
  current.locked_out = false;
}
else {
  current.locked_out = true;
}
var gr = new GlideRecord ("sys_user");
```

```
gr.query();
while(gr.next()) {
  gr.update();
  gs.print("updating " + gr.getDisplayValue());
}
```

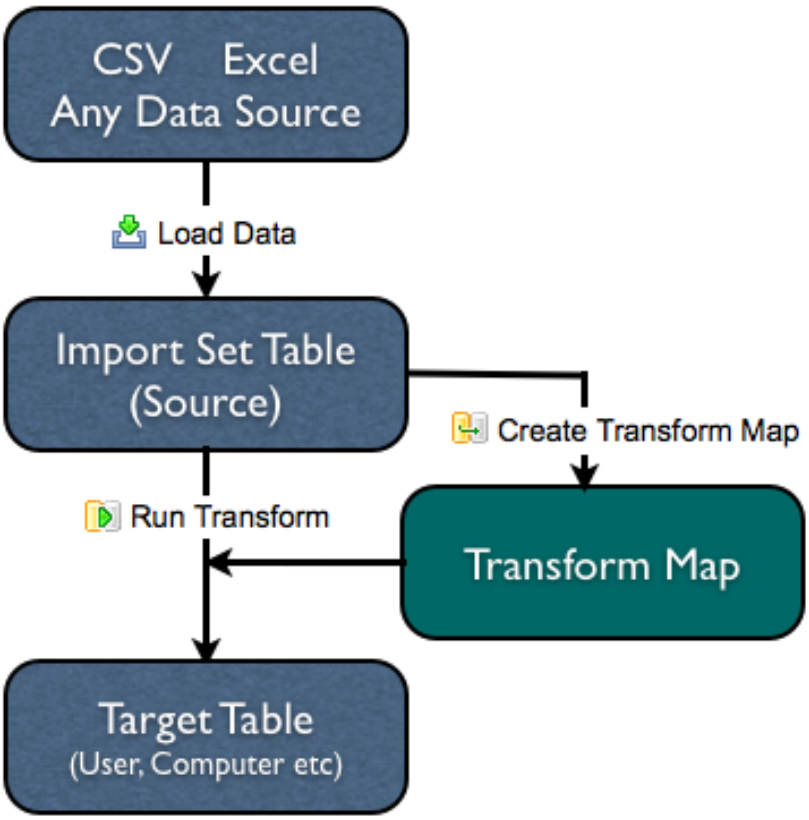
### References

[1] http://wiki.service-now.com/index.php?title=GlideRecord#setAbortAction

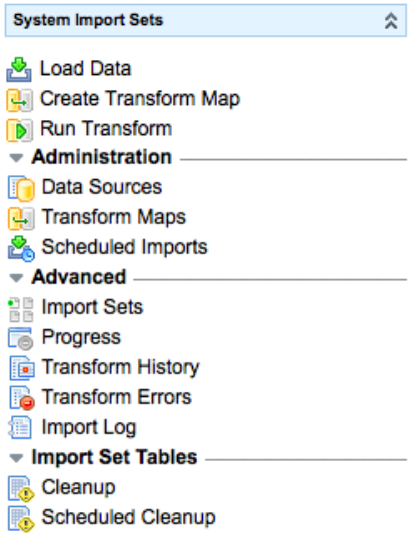
# Importing Data Using Import Sets

> Functionality described here requires the **Admin** role.

## Overview




Importing Data Data Flow



Import Sets Application

This page is a walk-through that will go step by step through the processes involved in importing data into ServiceNow. If you are importing data from an excel spreadsheet and do not wish to manipulate your data in any way then this page should be all you need. If however you wish to import from a networked data source like a JDBC connection or if you would like to verify or manipulate the data that you are importing then you may need to check out some of the additional documentation on Import Sets.

# Creating a New Import Table or Adding to an Existing One

From the left navigation menu, go to: **System Import Sets -> Load Data**

The second step is to create or load an Import Set table by uploading and Excel or CSV file, or from a predefined Data Source. For file imports, the CSV file format is preferred, due to occasional import issues with the way that Excel can alter data format.  The label is used to determine the name for the Import Set table that data will be loaded into. It is also possible to choose an existing Import Set table to use for loading data from the same source, or data that has the same field/column designations. When an existing Import Set table is chosen the table columns are added when the incoming source of data contains fields/columns that do not exist.

It is also necessary to choose a data source. It is possible at this point to define a new data source by browsing for a file to import. Supported formats are XLS and CSV. It is also necessary to specify the parameters of the file that is being imported. For example if the file that is being imported does not have any column headers then it is necessary to set the "Header row number" to 0. From here it is also possible to select an existing data source for use in the import operation.

**Note:** *By default, up to 20 records of the source data are sampled to determine the import set field length. If the field is empty in all the sampled records, the default length of 40 is used. Any data loaded that exceeds the import set table field length is truncated. To change this behavior and enable "dynamic" field resize, set the "com.glide.loader.verify_target_field_size" property to "true".*

# Reviewing the Import Set

Even if no errors were reported by the progress indicator it is still probably a good idea see how your new data was inputted into the new import set. To do so click on the link for "View imported data". There are several fields that will be visible in the table that were not part of the original data source, these are system rows can be used to facilitate scripted import operations. When you are finished reviewing the data, return to the previous page. The next step is to create a transform map to associate with this Import Set by clicking the link to do so.

# Creating a Transform Map

A Transform Map determines how data is mapped onto a ServiceNow table.

First it is necessary to specify a destination table and select a method of mapping.Under the right circumstances where field names in the Import Set and the destination table are identical or nearly identical this is a simple as clicking the "Auto map from matching fields" button. If any fields are not mapped by the auto matching utility then it is also possible to specify mapping relationships using a simple drag and drop interface by clicking the "Mapping assist" button. When using this utility it is possible to match one source field to multiple destination fields.

# Run Import

From a Transform Map, you can click Run Import to do just that. When the import is done, you'll see a link to go straight to the target table containing your imported records. The amount of time that it takes to run an import varies depending on the number of record to be imported and may take as long as several hours for very large import operations (tens of thousands of records).

At this point, you may click on the link View the imported data to see the loaded import set table, Create transform map to create a new transform map to transform the data in the import set table to its target table, or Run import to execute an existing transform map for the loaded data.

Here is the result of viewing the import set data (you may also click on the application module that is automatically created when the import set is created):

Three things to note at this point:

- The spreadsheet was imported, and a new table was created to hold the data.
- Within that table, the imported records are designated with their own "Set" value (in this case, "ISET10004").
- A new module was created in the System Import Sets application for the new table.

# Cancelling an Import Set

To stop a scheduled import that is taking too much time:

1. Navigate to *System Import Sets* > *Advanced* > *Progress*.
2. Open the Progress Worker that must be canceled.
3. Click the **Cancel job** Related Link.

# Posting CSV or Excel files directly to an Import Set

By doing an HTTPs post directly to the **sys_import.do** target, you can dynamically upload a CSV or Excel file into an import set table. The required parameter on the URL is **sysparm_import_set_tablename**, you must specify a name that matches an existing import set table name, if it does not exist, pre-create it by doing a manual import. If the **sysparm_transform_after_load=true** parameter is specified on the *--url* argument to the script, the CSV transform will execute immediately, if a transform map exists.

**Note:** *The ability to run transforms synchronously is new to June 2011 Preview 2*

## Perl Example

The following is an example of using perl to post a CSV file with basic auth credentials.

```
# file: uploadafile.pl
# call me like this:
# uploadafile.pl
--url="https://instance.service-now.com/sys_import.do?sysparm_import_set_tablename=dloo_t
 --uploadfile=/Users/davidloo/Desktop/test_files/test_users.csv
#
# the "sysparm_transform_after_load=true" parameter instructs the
import set to transform immediately after loading
#
use strict;
use LWP::UserAgent;
use HTTP::Request::Common;
use Getopt::Long;
use File::Basename;

my ( $o_url, $o_fqn );
GetOptions(
    "url=s"        => \$o_url,
    "uploadfile=s" => \$o_fqn,
);

# mandatory arguments: url
&usage unless ( $o_url && $o_fqn );

my $url   = $o_url;
my $fname = $o_fqn;

# put timeouts, proxy etc into the useragent if needed
my $ua  = LWP::UserAgent->new();

# setup basic authentication credentials
$ua->credentials(
  'demo.service-now.com:443',
  'ServiceNow',
  'admin' => 'admin'
);

my $req = POST $url, Content_Type => 'form-data',
        Content      => [
                submit => 1,
                upfile =>;  [ $fname ]
        ];
my $response = $ua->request($req);
```

```
if ($response->is_success()) {
    print "OK: ", $response->content;
} else {
    print $response->as_string;
}

exit;

sub usage {
        printf "usage: %s --url=%s --uploadfile=%s\\n",
                basename($0),'https://....','c:/data/test.csv';
        exit
}
```

## Java Example

The following is a Java example using Apache's HttpClient class.

```
HttpClient httpclient = new HttpClient();

PostMethod post = new

PostMethod("https://instance-name.service-now.com/sys_import.do?sysparm_import_set_tablename=u_test_upload&sysparm_transform_after_load=true");


try {

  Credentials defaultcreds = new UsernamePasswordCredentials("admin",

"admin");

  httpclient.getState().setCredentials(AuthScope.ANY, defaultcreds); //

 Prepare HTTP post


  File targetFile = new

File("/Users/davidloo/Desktop/test_files/nodeinfo2736820198834983863.csv");

  Part[] parts = { new FilePart(targetFile.getName(), targetFile) };


  post.setRequestEntity(new MultipartRequestEntity(parts,

post.getParams()));


  int result = httpclient.executeMethod(post);


  // Display status code

  System.out.println("Response status code: " + result);


  // Display response

  System.out.println("Response body: "+post.getResponseBodyAsString());

} catch(Exception e) {

  System.err.println(e.getMessage());

} finally {

  // Release current connection to the connection pool
```

```
  // once you are done

  post.releaseConnection();

}
```

# System Update Sets

## Overview

An **update set** is a group of customizations that can be moved from one instance to another. This feature allows administrators to group a series of changes into a named set and then move them as a unit to other systems. In most cases, update sets allow customizations to be developed in a development instance, moved to a test instance, and then applied to a production instance.

Before using System Update Sets, review the Getting Started with Update Sets guide.

## Recent Improvements

### Aspen

The following enhancements have been added as of the Aspen release:

- You can now compare versions and revert changes to objects on a table with the update_synch attribute (such as business rules). See Comparing Versions. The Versions [sys_update_version] table has been added to support this feature.
- The preview function now detects a **Problem** if an updated object does not exist on the local system or in the current update set. See Resolving Problems. The Update Problems [sys_update_preview_problem] table has been added to support this feature.
- The **BaseLine** update set has been removed.
- A new access control rule restricts the ability to delete update sets.
- A warning message now appears when a Customer Update (sys_update_xml) entry is deleted.
- Customer updates (sys_update_xml records) with no Update Set value are no longer ignored during an upgrade.
- Reports are available for tracking customer updates. See Reporting on Updates.
- A UI Policy has been added to make **sys_m2m** (many-to-many) fields mandatory.

## Developing Customizations with Update Sets

A common process for developing customizations with update sets is:

1. An administrator creates an update set on the development instance.
2. The administrator makes customizations and changes on the development instance.
3. The update set is marked **Complete**.
4. The administrator logs in to the test instance and retrieves the completed update set from the development instance.
5. The update set is applied to the test instance, and customizations are tested thoroughly.
6. The administrator logs in to the production instance and retrieves the completed update set from the development instance.
7. The update set is applied to production.

To learn more, see Using Update Sets.

# Understanding Update Sets

Each update set is stored in the Update Set [sys_update_set] table, and the customizations that are associated with the update set (entries in the Customer Update [sys_update_xml] table) appear as a related list on the update set record.

When a tracked object is customized, a corresponding record is added in the Customer Updates [sys_update_xml] table and is associated with the user's current update set. A corresponding record is also added to the Versions [sys_update_version] table. Administrators can compare two versions and revert to a specific version of an object (available with the Aspen release).

**Note:** *Customer Updates [sys_update_xml] records should not be modified directly.*

## System-Maintained Update Sets

The **Default** update set provides system functionality and should not be changed, deleted, or moved between systems. Use this update set to make changes to an instance without adding the changes to any user-created update sets.

- If a user does not have a current update set selected, changes are tracked in the **Default** update set.
- If the **Default** update set is marked **Complete**, the system creates another update set named **Default1** and uses it as the default update set.

**Click the plus for versions prior to October2011**

The following update sets provide system functionality and should not be changed, deleted, or moved between systems:

- **Default**: used when a user would like to make changes to an instance without adding the changes to any user-created update sets. If a user does not have a current update set selected, the changes are tracked in this update set.

  If the set is marked **Complete**, the system creates another update set named **Default1** and uses it as the default update set.
- **BaseLine**: stores the contents of items before they were changed for the first time. **BaseLine** is used if an installed update set is removed or backed out. To remove or back out the changes made by an update set, the system uses the **sys_update_log** table. For more information, see Backing out Update Sets.

Update Sets New Go to Name 1 to 2 of 2

| Name | State | Installed from | Created | Created by |
|---|---|---|---|---|
| BaseLine | In progress | | 2008-04-25 16:37:36 | glide.maint |
| Default | In progress | | 2008-04-25 16:44:29 | glide.maint |

# Determining Which Customizations are Tracked

Customizations are tracked by update sets only for some objects, using one of the following methods:

- The **update_synch** attribute on the table
- Special handlers for changes that require updates to multiple tables
- Manual unloading for homepages

Update sets cannot be used to transfer other system changes or data. In general, update sets capture configuration information but not task or process data. For example, update sets track service catalog item definitions and related configuration data like variables and variable choices. However, if you test the service catalog by placing orders, the orders (requests, items, catalog tasks) are not tracked by update sets.

## The update_synch Attribute

To see a list of tables on which customizations are tracked, navigate to **System Definition > Dictionary** and filter on **attributes CONTAINS update_synch**.

A default rule blocks the use of the **update_synch** attribute on a table for which it is not predefined to avoid the following issues:

- Some core tables require special update handling because they represent information on multiple tables. When the **update_synch** attribute is added to these tables, duplicate update records are created, causing major conflicts that are difficult to troubleshoot and repair.
- Using the **update_synch** attribute to migrate data records between instances can cause performance issues, as it is not intended for this purpose. To migrate data, use an instance-to-instance import.

**Warning:** Do not add the *update_synch* attribute to a dictionary record. When improperly used, this attribute can cause major performance issues or cause the instance to become unavailable.

## Special Handlers

Some changes require special handlers because they represent information on multiple tables. These changes are packaged into one update set entry so that all records are properly updated when the customization is applied. The following changes are tracked with special handlers:

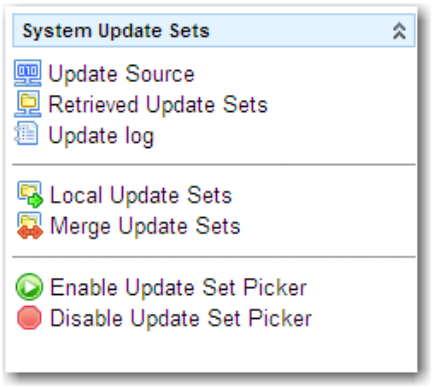
- Workflows
- Form sections
- Lists
- Related lists
- Choice lists
- Content pages (in Content Management)
- Database changes
- System dictionary entries
- Field labels

## Using Homepages with Update Sets

Homepages are not added to update sets by default. To add a homepage to the current update set:

1. Navigate to **Homepage Admin > Pages**.
2. Right-click a homepage record.
3. Select **Unload Portal Page**.

# Application and Modules



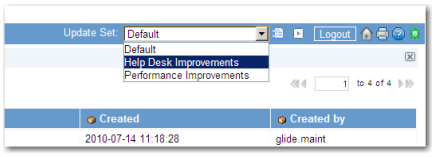

System Update Sets application

The following modules are included in the System Update Sets application:

- **Update Source**: defines the instances from which to retrieve update sets. See Transferring Update Sets.
- **Retrieved Update Sets**: lists the update sets that have been retrieved from update sources.
- **Update Log**: shows the history of applying update sets.
- **Local Update Sets**: lists the update sets created on the instance. See Creating Update Sets.
- **Merge Update Sets**: provides the ability to merge update sets.
- **Enable Update Set Picker**: enables the update set picker in the navigation header.
- **Disable Update Set Picker**: disables the update set picker.

# Granting Access to the Update Set Picker

Use the Update Set Picker to select the update set in which you track customizations.




**Update Set Picker**

To remove the Update Set Picker, navigate to **System Update Sets > Disable Update Set Picker.** To restore it, navigate to **System Update Sets > Enable Update Set Picker**.

By default, only administrators can use the update set picker. To grant access to additional users:

1. Navigate to **sys_properties.list**
2. Create a new property:
   - **Name**: enter **glide.ui.update_set_picker.role**
   - **Value**: enter the role that has access to the update set picker (in addition to administrators).

     **Note**: This property accepts only one role. To give access to multiple roles, create a new role (example, update_set_picker) and include that role in the roles that need access.
3. Click **Submit**.

# Introduction to Knowledge Base

## Overview

The knowledge base is a tool that allows you to store and publish any information, such as desktop support information, company/department processes or procedures, or documentation on internally developed applications. Administrators or those with the knowledge role can add articles to the Knowledge Base. For more on adding content, see Adding Content to the Knowledge Base.

Knowledge articles can be created from scratch or generated from a task (such as an incident or problem). Knowledge base articles can also be accessed from any form that contains the knowledge base icon:  .

## Knowledge Base Homepage

The knowledge base homepage is populated with articles that are organized by topic and category. The most recent articles of a given category appear in the category's gauge with a date-stamp on the right. Click the article name to open it. Click **View all items** to open a list of all articles in the category.

Refine your searches according to topic or category by clicking **Advanced search** next to the search bar in the top left.

## Knowledge Base List

When you click the **View all items** link for a category, the knowledge base list displays the articles in that category. Only published articles whose **Valid to** date has not passed appear in the this list. Unlike a record list, you cannot use the list editor to alter the entries in the knowledge base list. Administrators can access and modify the knowledge base records by navigating to **Knowledge Base > Edit**.

## Knowledge Base Article

Selecting an article name opens the article followed by some data about the article. Leave feedback by rating it under **Was this helpful?** or by expanding **Feedback** at the bottom of the article and submitting a comment. Administrators see an "Edit Article" button that opens the form for the knowledge base article.

<br

# Using Access Control Rules

## Overview

All security in the contextual security manager is dictated by access control rules. You can get a list of all these rules under *System Security > Access Control* in your navigator.

An access rule consists of two core things, a description of the entity and operation being secured, and a description of the rights required to access it.

An entity/operation pair is defined by three fields:

- **Type**: Type of entity being secured (e.g. *record)*
- **Operation**: Operation being secured (e.g. *read)*
- **Name**: Unique identifier describing the object (may include wildcards. * = wildcard). The simplest identifier is one that fully identifies an item.

## Securing Access

ACL rules can be defined in three ways:

- Roles
- Conditional Expressions
- Scripts

**Note:** *Out-of-box, there is a rule for "*.*" on the create operation which defaults the create operation to whatever the ACL rule is for the write operation. If a write operation ACL is defined on a table but a create ACL is not, it will default to the write operation's ACL. To override, simply create a create operation ACL*

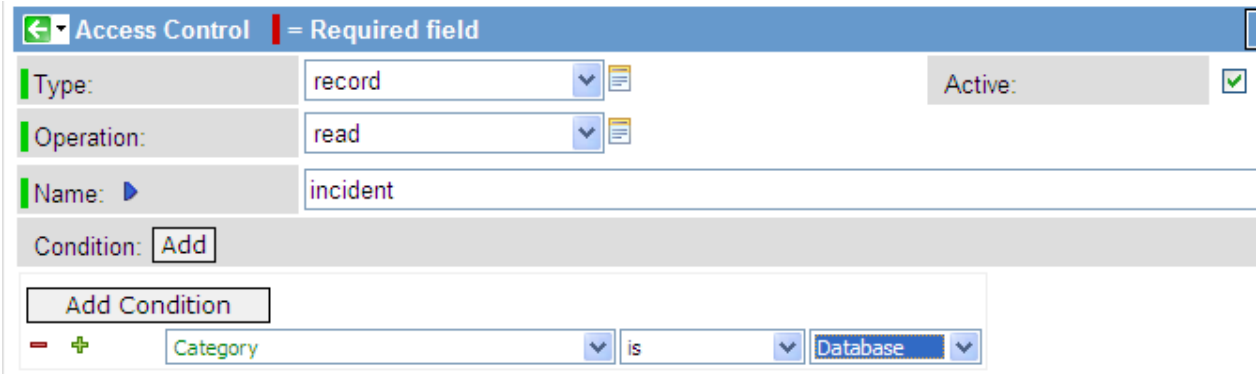
### Securing via Roles

At the bottom of your Access Control Rule there's a related list of roles. If you put one or more roles there, then only users with at least one of those roles are allowed to perform the requested operation.

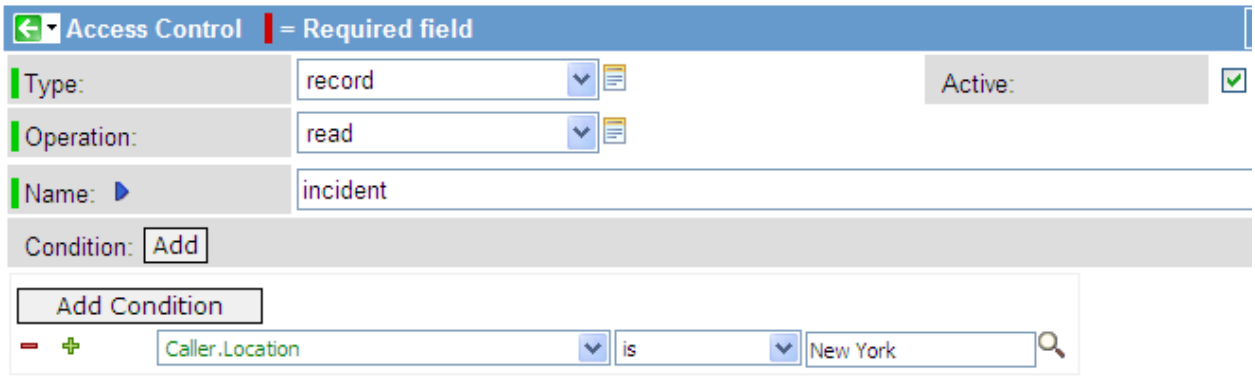
### Security via Conditional Expressions

In the middle of your Access Control Rule, you'll see a condition widget where you can add conditional expressions to your ACL. For example, you might have a conditional expression that says "category is Database" if you want this rule to evaluate to true only for database records.

#### Sample Condition 1: Allow only if category is Database



### Sample Condition 2: Allow only if caller is from New York




## Security via Script

You also have an opportunity to apply security based on user defined script. Your script has access to the current record and has responsibility for setting a global "answer" variable to allow, or deny, access to the requested resource/operation. Script is evaluated *in addition to* other conditions you set on the rule. All must evaluate to true.

**Sample Script 1: Allow only if assigned to me**

```
answer = gs.getUserID() == current.assigned_to;
```

**Sample Script 2: Allow only if I'm a member of the current assignment group**

```
answer = gs.getUser().isMemberOf(current.assignment_group);
```

**Sample Script 3: Allow only if I'm the group manager of the current assignment group**

```
answer = gs.hasRoleInGroup('group_manager', current.assignment_group);
```

# Securing UI Pages with Access Control Rules

To restrict access to a specific UI Page, create an ACL with a Type of "ui_page", an Operation of "read", and a Name value equal to the name of a UI Page.

For an example of how to limit access to Live Feed using this type of rule, see Limiting Live Feed Access by Role.

# Rule Levels

## Field Level Rules

Field level rules apply security to a particular field in a particular table. For example:

incident.number -- this rule applies to the **number** field on the **incident** table

You can also use a wildcard if you want. For example:

incident.* -- this rule applies to **all** fields on the **incident** table **except** those with explicit rules

Thus, given the two rules above, incident.number would use its explicit rule, but another field, say, incident.short_description would use incident.*

## Row Level Rules

Row level rules apply security to an entire row. For example:

incident -- this rule applies to all rows in the incident table

## Precedence between Row and Field Level Rules

What happens if a row level rule and a field level rule are in conflict? Perhaps my row level field indicates that I shouldn't be able to write to a particular row, but the field level rule indicates I do have write access?

In a nutshell, *both* rules must be met before an operation is allowed.

So, given a row level rule on incident, and a field level rule on incident.number, access to the number field would be allowed only if both rules evaluated to true.

## Rule Search Order

The system is aware of our object hierarchy when it tries to identify a security rule to apply to a particular entity. The search order for a field level rule is:

1. explicit rule on self
2. explicit rule on field in parent
3. ... until parent doesn't contain field
4. wildcard rule on self
5. wildcard rule on field in parent
6. ... until parent doesn't contain field

Example: Given incident.number

Search is:

1. incident.number
2. task.number
3. *.number
4. incident.*
5. task.*
6. *.*

## Multiple Rules at the Same Level

What if the system, for example, finds two rules for incident.number?

The system will evaluate both rules and if **either** is true, then the requested access is allowed.

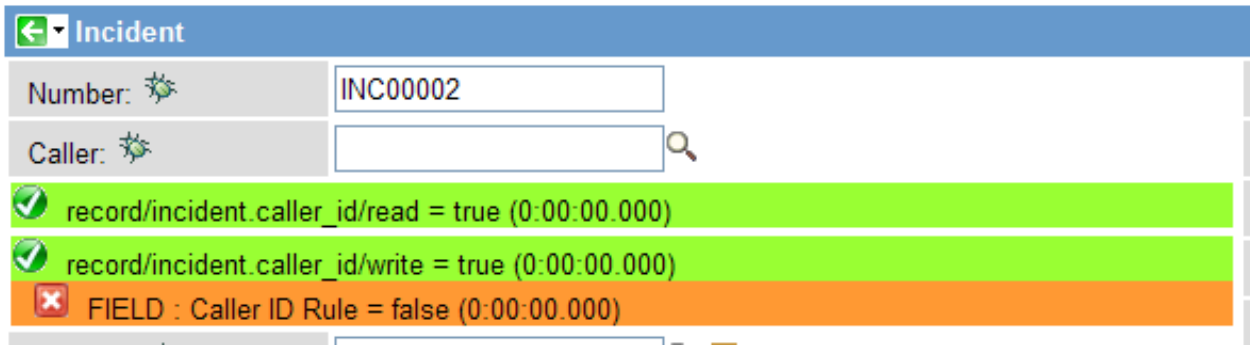
# Debugging

To turn on rule debugging, System Security > Debug Security Rules. Once you do so, you'll get two sets of debugging information.

**Note:** *Impersonation can simplify debugging ACL rules. First turn on ACL debugging, then impersonate another user to see the system from that user's perspective would have seen it.*

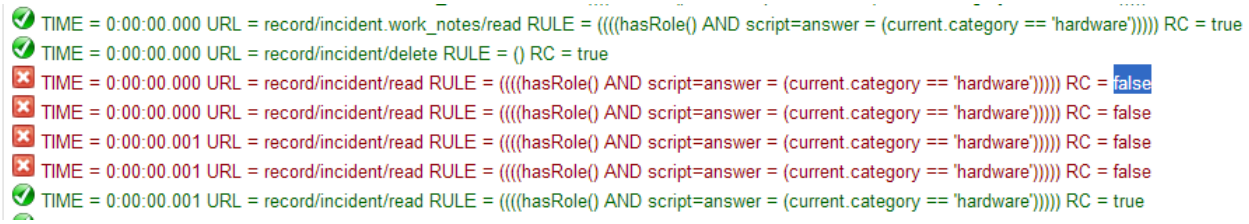
## Field Level Debugging

Each field with an ACL rule on it will now have a small "bug" icon (see below). Clicking that icon will expand a set of ACL rules for that particular field and their evaluation results.




## Rule Output

At the bottom of your form (both lists and edit forms) you'll get the low level output from the rule engine. While this might not be as granular as the field level information, it does show up on both lists and edit forms which can often help.




## Troubleshooting

Q. Why can my users see a specific field in a list but not on a form? A. You may have a security rule on a field that lets a user see data in the field SOMETIMES (based on some characteristic of the record). In this case, the field may appear in a list of records, though it will appear empty where the user cannot see the value. Check your ACLs carefully, and use debugging.

# Admin Overrides

Out-of-box, the **admin** role is exempt from ACL rules. To disable this override on a particular ACL rule, clear the **Admin overrides** checkbox on the ACL.

# Security

## Overview

Security is built into all levels of ServiceNow functionality. Implement the security features that are most appropriate for your organization, from managing failed logins and encrypted password protection, to roles and access control rules, to audit logs of user interactions.

### Login Security

Prevent unauthorized access to the ServiceNow platform.

### General Security

Establish roles, property settings, access control settings, and more to assure ServiceNow security.

### Auditing and System Logs

Control which interactions with ServiceNow are tracked and logged, and view the logs.

### Encryption

Apply encryption to fields, tables, and attachments to limit access to users with the proper credentials.

### Company and Domain Separation

Manage data access by company or domain to assure that only appropriate information, processes, and options are available to users.

### Contextual Security

Protect information by controlling read/write/create/delete authorization based on record contents and object hierarchy.

# Chat

## Overview

Chat provides real-time communication via instant messaging between users in a ServiceNow instance. Features include:

- One-to-one chats (instant messaging) between users.
- Chat rooms for conversations with multiple users. Chat rooms may be *Public* (any user can join) or *Private* (only invited users can join).
- Chat rooms linked to task records. Users can work together to solve issues, and conversation history can be shared by everyone who needs to reference it.
- Help Desk Chat. End users can access live support via instant messaging. Service Desk staff can resolve basic issues in real-time or create incidents directly from chat requests for more extensive issues.

This article helps administrators install and set up the Chat plugin, which is available as of the Winter 2011 release.

**Note:** *Depending on your upgrade cycle, your system may require an update to Winter 2011 to use the Chat plugin. If your system is running the Winter 2011 release, access the self service portal* [1] *to request this update before installing the plugin. To learn more, see Requesting an Upgrade.*

# Application & Modules

The **Social IT** application is added to the application navigator, with modules for using and administering instant messaging services.

For all users:

- **Chat**: Opens the chat desktop.
  - All users can have a conversation with one or multiple co-workers and view status updates. To learn more, see Using Chat.
  - Service Desk staff can also provide live support to users via instant messaging. To learn more, see Using Help Desk Chat.

For administrators and users with the chat_admin role:

- **Queues**: Opens the Chat Queues table. Define groups and schedules for providing support via instant message. To learn more, see Setting Up Chat Queues for Help Desk Chat.
- **Actions**: Opens the Chat Actions table. Define additional chat window menu items. To learn more, see Adding Actions to the Chat Window Menu.
- **Properties**: Opens the chat properties page (see Properties).

# Enhancements

## June 2011 Preview 1

The following enhancements have been added as of the June 2011 Preview 1 release:

- Users can now control audio and chat window display options from the **Chat Preferences** window. To learn more, see Setting Your Chat Preferences.
- A context (right-click) menu has been added to the **Online Users** list. Users can right-click a user name and select **Send Message** or **Add To Friend List**.
- A context menu has been added to the **Show All Rooms** list. Users can right-click a room name and select **Join Room**.
- **Send Message** has been added to the **My Friends** context menu.
- The chat room description now appears under the room name in the **Show All Rooms** list.
- Various improvements to friend status, layout, and general stability are included. To learn more, see June 2011 Preview 1 Fixes.

## June 2011 Preview 2

The following enhancements have been added as of the June 2011 Preview 2 release:

- The My Stuff window has been redesigned for better usability and is now called the buddy list. Also, My Friends is now Contacts and My Rooms list is now Rooms.
- Users can now change their profile picture in their buddy list. To learn more, see Updating Your Profile.
- Users can now change their status in their buddy list. Previously, users updated their status in the chat menu.
- Icons now appear beside a chat room name to indicate when a room is private () or a password is required ().
- Invitations now display the room name, description, and user who sent the invitation. Users can now click **Join** or **Decline**.
- For chat rooms, the member list is now organized into sections by active participants and invited users. Users can click the section name to collapse or expand it.
- Contacts are now organized alphabetically and into sections by status (*Online* and *Offline*). Users can click the section name to expand or collapse it. Previously, contacts were organized in the order they were added.

- Context (right-click) menus have been added to the **Contacts** and **Rooms** section headers. These menus provide the functions that were previously available as links (**Add Contact** and **Show Online Users** on Contacts; **Create Room** and **Show Rooms** on Rooms). The functions are also available from a new toolbar menu ().
- Internationalization (i18n) support has been added for system messages and UI elements in the chat desktop.
- Various improvements and fixes are included for audio notifications, system messages, window focus, IE support, room list, layout, and general stability.

## June 2011 Preview 3

The following enhancements have been added as of the June 2011 Preview 3 release:

- Users can now specify a **Room avatar** (picture) for chat rooms. The room avatar appears beside the room name in the buddy list and in the list of all rooms.

## Aspen Release

The following enhancements have been added in the Aspen release:

- The syntax for adding a custom help desk chat link to the ESS portal has changed.
- The buddy list is now called the favorites list and Contacts is now called Users.
- Chat room participants can now add the room as a favorite. The favorites list (formerly buddy list) shows favorite rooms.
- When creating chat rooms, users can now specify whether the room is public or hidden, temporary or saved, and members-only or open to everyone. Users without an invitation can now see public rooms only.
- Chat rooms for tasks are improved. If rooms exist already, a user-friendly list appears. When creating a new room, standard options are available.
- Users can now invite another user to any chat. One-to-one chats are automatically converted into temporary chat rooms when another user is invited.
- Additional emoticons (smileys) are now available.
- Users can now view and add to favorites only the users that they can see elsewhere in the system.
- Chat history now saves the most recent 350 messages.
- Chat preferences are now available on the chat window menu (). To improve usability, the chat desktop header has been removed.

# Activating Chat

Before activating the Chat plugin, consider the installed components, dependencies, and impact.

- **Installed Components**: Includes tables, a field, business rules, a script include, an application, a user role, properties, an event, and an email notification. For more details, review the components that are Installed with Chat.
- **Dependencies** (installed automatically): Social IT Infrastructure
- **Impact**: The plugin installs new features; it does not overwrite or impact current configurations. It has minimal impact on the system. However, when the system is configured to use short polling (see Properties) and the client is in debug mode, users may experience a performance impact. Polling also keeps the session alive when the chat desktop is open (see Modifying Session Timeout).

**Instructions for activating a plugin.**

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

   If the plugin depends on other plugins, these plugins and their activation status are listed. Fall 2010 Stable 3.
4. Click **Activate**.

# Securing Chat

## Accessing Chat Tables

By default, administrators and users with the chat_admin role can view all chat tables and can modify the following chat tables:

- chat_action: Defines additional chat window menu items (see Adding Actions to the Chat Window Menu).
- chat_queue: Defines groups and schedules for providing support via instant message (see Setting Up Chat Queues for Help Desk Chat).
- chat_room: Can be modified if necessary (such as to change a chat room from *Public* to *Private*), but should almost always be managed by system functionality.

Extending or modifying data in any other chat table is **not** recommended.

## Restricting Chat Room Creation

To define the access rights for creating chat rooms:

1. Navigate to **Social IT > Chat Administration > Properties**.
2. Locate the property *Comma-separated list of roles that are allowed to create chat rooms.*
3. Enter user roles. A user must have one of the specified roles to create a chat room.

## Restricting Access to Chat Activity on Task Records

Administrators can add the **Chat Activity** field to a task record (see Configuring Chat Activity on Task Records). Read access to the messages displayed in this field is handled by the security operation *chat_messages_read* on the *chat_room* table.

By default, a user can read the chat messages for a room if:

- The room is public
- The user is a member of the room

To change chat room message read access, modify the access control rule. To learn more, see Using Access Control Rules.

# Configuring Chat Activity on Task Records

Chat rooms can be linked to any task record in the following ways:

- Create a room from the record (see Creating Chat Rooms for Tasks).
- Create a record from a chat room (such as creating an incident from a Help Desk Chat, see Providing Support (Service Desk Staff)).

Administrators can add the chat messages as a journal field by personalizing the task record form to add the **Chat Activity** field. The maximum number of chat messages that are displayed by this field is 1000.

## References

[1] https://hi.service-now.com/

# Article Sources and Contributors

**Training:ServiceNow Certification** *Source*: http://wiki.service-now.com/index.php?oldid=91101 *Contributors*: Michelle.Corona, Rachel.sienko

**Administering Applications and Modules** *Source*: http://wiki.service-now.com/index.php?oldid=97940 *Contributors*: CapaJC, Cheryl.dolan, G.yedwab, Guy.yedwab, Jared.laethem, Joseph.messerschmidt, Mark.stanger, Rachel.sienko, Suzannes, Swood, Vhearne

**Activating ServiceNow Plugins** *Source*: http://wiki.service-now.com/index.php?oldid=91850 *Contributors*: CapaJC, Cheryl.dolan, G.yedwab, Guy.yedwab, Jerrod.bennett, John.roberts, Joseph.messerschmidt, Kdooren, Mark.stanger, Rachel.sienko, Rob.woodbyrne, Swood, Vhearne

**Personalizing Forms** *Source*: http://wiki.service-now.com/index.php?oldid=96670 *Contributors*: CapaJC, G.yedwab, Guy.yedwab, Joe.Westrich, Joseph.messerschmidt, Rachel.sienko, Swood, Vhearne

**Customizing Homepages** *Source*: http://wiki.service-now.com/index.php?oldid=80181 *Contributors*: Bow, CapaJC, Davida.hughes, Dhorner, Dkearney, G.yedwab, Guy.yedwab, Jared.laethem, Jerrod.bennett, Joe.Westrich, Joseph.messerschmidt, Mark.stanger, Myla.jordan, Neola, Pat.Casey, Rachel.sienko, Rob.woodbyrne, Swood, Vhearne

**Service Level Agreements** *Source*: http://wiki.service-now.com/index.php?oldid=95905 *Contributors*: Cheryl.dolan, Davida.hughes, G.yedwab, Guy.yedwab, Joseph.messerschmidt, Rachel.sienko, Steve.wood

**Creating an SLA Workflow** *Source*: http://wiki.service-now.com/index.php?oldid=75510 *Contributors*: Davida.hughes, G.yedwab, Guy.yedwab, Nick.roberts, Rachel.sienko, Swood, Voytek.blonski

**Service Catalog Administration** *Source*: http://wiki.service-now.com/index.php?oldid=73580 *Contributors*: CapaJC, Guy.yedwab, Pat.Casey, Rachel.sienko, Swood, Vhearne

**Service Catalog Variables** *Source*: http://wiki.service-now.com/index.php?oldid=91911 *Contributors*: CapaJC, Cheryl.dolan, Guy.yedwab, Pat.Casey, Rachel.sienko, Rob.woodbyrne, Vhearne

**Service Catalog Workflows** *Source*: http://wiki.service-now.com/index.php?oldid=73602 *Contributors*: G.yedwab, Guy.yedwab, Rachel.sienko, Swood

**Business Rules** *Source*: http://wiki.service-now.com/index.php?oldid=97923 *Contributors*: Cheryl.dolan, G.yedwab, Guy.yedwab, John.andersen, John.roberts, Joseph.messerschmidt, Michelle.Corona, Neola, Rachel.sienko, Steve.wood, Suzannes

**Importing Data Using Import Sets** *Source*: http://wiki.service-now.com/index.php?oldid=83389 *Contributors*: Ajandersen, Cheryl.dolan, David Loo, G.yedwab, Guy.yedwab, Jared.laethem, Joseph.messerschmidt, Michelle.Corona, Pinnacle ls, Rob.woodbyrne, Steve.wood, Swood, Vhearne

**System Update Sets** *Source*: http://wiki.service-now.com/index.php?oldid=98397 *Contributors*: Annmarie, Brozi, CapaJC, G.yedwab, Guy.yedwab, Jared.laethem, Jerrod.bennett, Jessi.graves, Joseph.messerschmidt, Rachel.sienko, Steve.wood, Suzannes, Swood, Vhearne

**Introduction to Knowledge Base** *Source*: http://wiki.service-now.com/index.php?oldid=94829 *Contributors*: Cheryl.dolan, Davida.hughes, Guy.yedwab, Neola, Rachel.sienko

**Using Access Control Rules** *Source*: http://wiki.service-now.com/index.php?oldid=97521 *Contributors*: CapaJC, G.yedwab, Guy.yedwab, Neola, Rachel.sienko, Steve.wood, Suzannes, Swood

**Security** *Source*: http://wiki.service-now.com/index.php?oldid=97950 *Contributors*: Cheryl.dolan, Joe.Westrich, Joseph.messerschmidt, Steve.wood

**Chat** *Source*: http://wiki.service-now.com/index.php?oldid=95864 *Contributors*: Cheryl.dolan, Joe.Westrich, Joseph.messerschmidt, Mark.johnson, Rachel.sienko, Wallymarx

# Image Sources, Licenses and Contributors

**Image:Application.png** *Source*: http://wiki.service-now.com/index.php?title=File:Application.png *License*: unknown *Contributors*: CapaJC, Guy.yedwab
**Image:Update.png** *Source*: http://wiki.service-now.com/index.php?title=File:Update.png *License*: unknown *Contributors*: CapaJC
**Image:ApplicationCategory1.png** *Source*: http://wiki.service-now.com/index.php?title=File:ApplicationCategory1.png *License*: unknown *Contributors*: CapaJC
**Image:Module.png** *Source*: http://wiki.service-now.com/index.php?title=File:Module.png *License*: unknown *Contributors*: CapaJC, Guy.yedwab
**Image:ExternalLink1.png** *Source*: http://wiki.service-now.com/index.php?title=File:ExternalLink1.png *License*: unknown *Contributors*: CapaJC, Guy.yedwab, Swood
**Image:Warning.gif** *Source*: http://wiki.service-now.com/index.php?title=File:Warning.gif *License*: unknown *Contributors*: CapaJC
**Image:PersonalizeFormLayout.png** *Source*: http://wiki.service-now.com/index.php?title=File:PersonalizeFormLayout.png *License*: unknown *Contributors*: CapaJC
**Image:SlushBucketView.png** *Source*: http://wiki.service-now.com/index.php?title=File:SlushBucketView.png *License*: unknown *Contributors*: CapaJC
**Image:Home.png** *Source*: http://wiki.service-now.com/index.php?title=File:Home.png *License*: unknown *Contributors*: CapaJC
**Image:HomePageRefresh.png** *Source*: http://wiki.service-now.com/index.php?title=File:HomePageRefresh.png *License*: unknown *Contributors*: CapaJC
**Image:TaskSLA.jpg** *Source*: http://wiki.service-now.com/index.php?title=File:TaskSLA.jpg *License*: unknown *Contributors*: G.yedwab, Kevin.leake
**Image:SLA-application-j11p2.png** *Source*: http://wiki.service-now.com/index.php?title=File:SLA-application-j11p2.png *License*: unknown *Contributors*: Guy.yedwab
**Image:IncidentInitial.jpg** *Source*: http://wiki.service-now.com/index.php?title=File:IncidentInitial.jpg *License*: unknown *Contributors*: Kevin.leake
**Image:IncidentMid.jpg** *Source*: http://wiki.service-now.com/index.php?title=File:IncidentMid.jpg *License*: unknown *Contributors*: Kenny.gabel, Kevin.leake
**Image:IncidentPause.jpg** *Source*: http://wiki.service-now.com/index.php?title=File:IncidentPause.jpg *License*: unknown *Contributors*: Kenny.gabel, Kevin.leake
**Image:IncidentAfterPause.jpg** *Source*: http://wiki.service-now.com/index.php?title=File:IncidentAfterPause.jpg *License*: unknown *Contributors*: Kenny.gabel, Kevin.leake
**Image:IncidentBreached.jpg** *Source*: http://wiki.service-now.com/index.php?title=File:IncidentBreached.jpg *License*: unknown *Contributors*: Kevin.leake
**Image:SLA Workflow.png** *Source*: http://wiki.service-now.com/index.php?title=File:SLA_Workflow.png *License*: unknown *Contributors*: G.yedwab, Guy.yedwab
**Image:Workflow Properties.png** *Source*: http://wiki.service-now.com/index.php?title=File:Workflow_Properties.png *License*: unknown *Contributors*: Guy.yedwab
**Image:Variablepicker2.png** *Source*: http://wiki.service-now.com/index.php?title=File:Variablepicker2.png *License*: unknown *Contributors*: Guy.yedwab
**Image:Notification.png** *Source*: http://wiki.service-now.com/index.php?title=File:Notification.png *License*: unknown *Contributors*: Guy.yedwab
**Image:SLA Workflow2.png** *Source*: http://wiki.service-now.com/index.php?title=File:SLA_Workflow2.png *License*: unknown *Contributors*: Guy.yedwab
**Image:SLA workflow3.png** *Source*: http://wiki.service-now.com/index.php?title=File:SLA_workflow3.png *License*: unknown *Contributors*: Guy.yedwab
**Image:Workflow Actions.png** *Source*: http://wiki.service-now.com/index.php?title=File:Workflow_Actions.png *License*: unknown *Contributors*: Guy.yedwab
**Image:CatalogQuantity2.png** *Source*: http://wiki.service-now.com/index.php?title=File:CatalogQuantity2.png *License*: unknown *Contributors*: CapaJC, Swood
**Image:CatalogQuantity1.png** *Source*: http://wiki.service-now.com/index.php?title=File:CatalogQuantity1.png *License*: unknown *Contributors*: CapaJC, Guy.yedwab, Swood
**Image:SCWorkflow1.png** *Source*: http://wiki.service-now.com/index.php?title=File:SCWorkflow1.png *License*: unknown *Contributors*: Guy.yedwab
**Image:SCWorkflow2.png** *Source*: http://wiki.service-now.com/index.php?title=File:SCWorkflow2.png *License*: unknown *Contributors*: Guy.yedwab
**Image:Business_rules_flow.jpg** *Source*: http://wiki.service-now.com/index.php?title=File:Business_rules_flow.jpg *License*: unknown *Contributors*: John.roberts
**Image:Import set overview.png** *Source*: http://wiki.service-now.com/index.php?title=File:Import_set_overview.png *License*: unknown *Contributors*: David Loo
**Image:import_set_app.png** *Source*: http://wiki.service-now.com/index.php?title=File:Import_set_app.png *License*: unknown *Contributors*: David Loo
**Image:Us1.png** *Source*: http://wiki.service-now.com/index.php?title=File:Us1.png *License*: unknown *Contributors*: CapaJC, Don.Goodliffe
**Image:updatesetapp.png** *Source*: http://wiki.service-now.com/index.php?title=File:Updatesetapp.png *License*: unknown *Contributors*: G.yedwab
**Image:Update set picker.png** *Source*: http://wiki.service-now.com/index.php?title=File:Update_set_picker.png *License*: unknown *Contributors*: G.yedwab
**Image:Kbicon.png** *Source*: http://wiki.service-now.com/index.php?title=File:Kbicon.png *License*: unknown *Contributors*: Rob.woodbyrne
**Image:Sec cat db.png** *Source*: http://wiki.service-now.com/index.php?title=File:Sec_cat_db.png *License*: unknown *Contributors*: Pat.Casey
**Image:Sec caller nyc.png** *Source*: http://wiki.service-now.com/index.php?title=File:Sec_caller_nyc.png *License*: unknown *Contributors*: Pat.Casey
**Image:Acl debug.png** *Source*: http://wiki.service-now.com/index.php?title=File:Acl_debug.png *License*: unknown *Contributors*: Pat.Casey
**Image:Acl debug list.png** *Source*: http://wiki.service-now.com/index.php?title=File:Acl_debug_list.png *License*: unknown *Contributors*: Pat.Casey