

# Using the User Administration Application

---

# Introduction

---

## User Administration

---

### Overview

Manage the individuals who can access ServiceNow by defining them as users in the system and assigning them to groups. Use the session control options to terminate ServiceNow sessions, for example when system maintenance is required. Create roles that provide selective access to ServiceNow functionality, then assign the roles to groups when all associated users need to access that functionality, or to individual users.

#### Users and Groups

Keep the focus on people through effective user management.

#### Roles

Simplify permissions and security by assigning roles.

#### User Sessions

Customize session rules and access user sessions directly

#### Group On-Call Rotation

Ensure optimum availability of on-call personnel

---

# Functions

## Managing User Sessions

### Overview

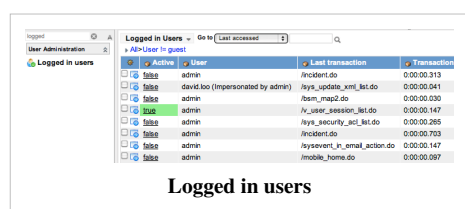
ServiceNow provides the ability to view and terminate individual user sessions. You may want to use this feature for locking out a user and terminating his session immediately, or terminating all user sessions during system maintenance. Terminating a user session has the result of logging that user out on the next transaction (usually the next browser click).

Note that however many windows a user has open, it is considered one session. If the user has two separate browsers open (for example, IE and Firefox), this is considered two separate sessions. To change the session timeout, see [Modifying Session Timeout](#).

To access these functions, select **User Administration > Logged in users** from the left-navigation pane. If you don't see this module, contact Technical Support to request activation of the **User Session Management** plugin.

Notes:

- You can only see users that are logged into the same application node as you.
- If the **Active** field value is **false**, the user is logged in but not currently running a transaction. Thus, most users appear as inactive at any given time. Also, the current user always appears as active because you run a transaction to view the list.



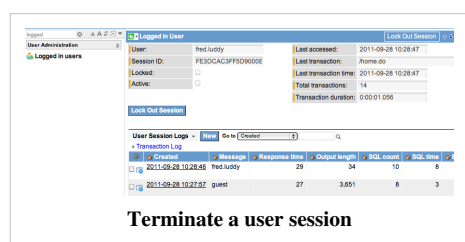
### Terminating a Specific User Session

To terminate a specific user session:

- From the left navigation pane, select **User Administration > Logged in users**.

- Select the session you want to end.
- Click **Lock Out Session**.

The session is terminated, and the user is redirected to the login page at the next attempted transaction. The user is not "locked out". Multiple user sessions may be associated with one user; terminating a user session only affects the specific session.

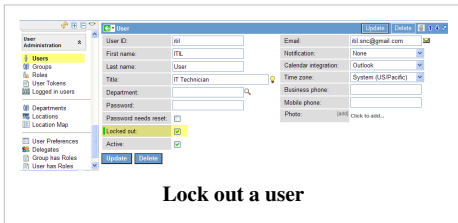


### Locking out a User

A locked out user cannot log in to an instance. Since the Berlin Release, locked out users cannot send inbound emails to the instance. To lock a user out of your instance and terminate all of their active sessions:

- From the left navigation pane, select **User Administration > Users**, and select the user from the list.

2. Select the **Locked Out** check box, and update the record.



The screenshot shows the 'User Administration' interface. On the left, a navigation pane lists 'Users', 'Groups', 'Roles', 'User Tokens', 'Logged in users', 'Departments', 'Locations', 'Location Map', 'User Preferences', 'Delegates', 'Group has Roles', and 'User has Roles'. The 'Users' section is selected. The main area shows a form for editing a user. The 'Locked out' checkbox is checked, and the 'Active' checkbox is unchecked. The 'Update' button is visible at the bottom right of the form.

**Lock out a user**

## Marking a User Inactive

An inactive user:

1. Does not show up in lists of users
2. Does not show up in the magnifying glass on reference fields

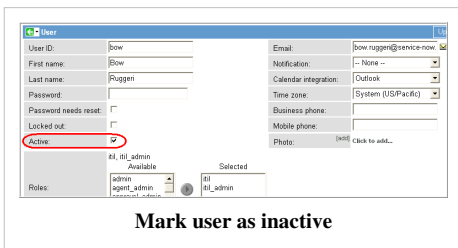
3. Does not show up in the drop-down list of users you get when you type into a reference field

An admin can see all users, regardless of state.

Making a user inactive does not lock out the user. A business rule (called **Lock Out Inactive Users** on the sys\_user table) sets the **Locked Out** flag to **true** whenever a user is flagged as inactive, but it is possible to have a user who is inactive but can still log in to the system.

To mark a user as inactive:

1. From the left navigation pane, select **User Administration > Users**, and select the **user** from the list.
2. Uncheck the **Active** checkbox, and update the record.



The screenshot shows the 'User Administration' interface. On the left, a navigation pane lists 'Users', 'Groups', 'Roles', 'User Tokens', 'Logged in users', 'Departments', 'Locations', 'Location Map', 'User Preferences', 'Delegates', 'Group has Roles', and 'User has Roles'. The 'Users' section is selected. The main area shows a form for editing a user. The 'Active' checkbox is unchecked, and the 'Locked out' checkbox is checked. The 'Update' button is visible at the bottom right of the form.

**Mark user as inactive**

## Enhancements

### Dublin

Administrators can add the following properties to the System Properties table:

- `glide.security.csrf.handle.ajax.timeout`
- `glide.security.auto.resubmit.ajax`
- `glide.ui.auto_req.extend.session`

See [Modifying Session Timeout](#) for an explanation of what these properties do.

# Creating Roles

---

## Overview

A role is a category that can be assigned to a group or user, and can be granted access to particular parts of the system. Once access has been granted to a role, all of the groups or users assigned to that role are granted the same access. Roles can also contain other roles, and any access granted to one role will be granted to any role that contains it.

For a complete list of the roles included with ServiceNow, see [Base System Roles](#).

## Creating Roles

1. Navigate to **User Administration > Role**.
2. Click **New**.
3. Give the role a unique, descriptive name and a brief description.
4. Click **Submit**.

The new role appears on the Roles list. It does not have access to any applications or modules until you add other roles to it or add the new role to the appropriate applications and modules.

5. To add other existing roles to the new role, open the role in form view and click **Edit** in the **Contains Roles** Related List.

Use the slushbucket to add the appropriate existing roles to the new role and click **Save**. Users who are assigned the new role automatically inherit access to the same applications and modules as the existing roles added here.

6. To create a role to add to the new role, click **New** in the **Contains Roles** Related List.
7. To give the role access to additional applications or modules:
  1. Navigate to **System Definition > Applications** or **System Definition > Modules**.
  2. Click the appropriate application or module to open it in form view.
  3. Click the lock to open the **Roles** field.
  4. Use the slushbucket to add the desired roles to the application or module.
  5. Click the lock to close the **Roles** field, then save your changes.

# Counting Licensed Users

## Overview

Most ServiceNow products are licensed per user. As of October 1, 2013, there are three user types. Contact your ServiceNow account manager for help reporting on license usage by user type.

- **Requesters:** can submit requests and manage their own requests, access public pages, take surveys, and use live feed and chat. Requesters are typically end users who access the instance through an employee self-service portal. Requesters have no associated roles.
- **Approvers:** can perform all requester actions and view or modify requests directed to the approver. Approvers have the `approver_user` role, but no other roles.
- **Fulfillers:** can access all functionality based on assigned roles. Fulfillers have one or more roles other than the `approver_user` role.



**Note:** The *Licensed Users* module displays a list of all users with any role. It does not differentiate between approvers and fulfillers. Do not use this module to determine the number of licensed users on your instance. For customers whose contracts have not been converted to the October 2013 pricing model, this module can be used to distinguish process users from end users.

# Adding a New Department

## Overview

Departments are another way to categorize users, groups, and assets.

## Adding a New Department

1. From the left navigation pane, select **User Administration --> Departments**.
2. Click **New** to add a new department
3. When done, click **Submit**

Department	
Name:	<input type="text"/>
ID:	<input type="text"/>
Department Head:	<input type="text"/>
Primary contact:	<input type="text"/>
Description: <input type="text"/>	
<input type="button" value="Submit"/>	

# Impersonating a User

---

## Overview

Administrators can impersonate other users for testing purposes. When impersonating another user, the administrator has access to exactly what that user would have access to in the system, including the same menus and modules. ServiceNow records anything the administrator does while impersonating another user as having been done by that user.

Use this feature to test what different users can do in the system and to perform actions for them in their stead.

## Useful Logins

Several different logins are recommended to test the system:

- An **admin** account to do work.
- An **itil** (or similar) login to test as a technician.
- An **ess** login to test as an end user.

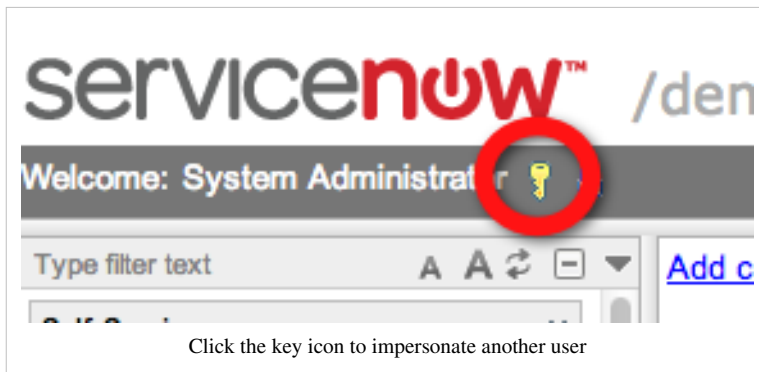
More logins may be required to adequately test the system.



**Note:** Impersonating a user who has been locked out or is inactive will force the administrator out of the system, just as a locked-out or inactive user would be.

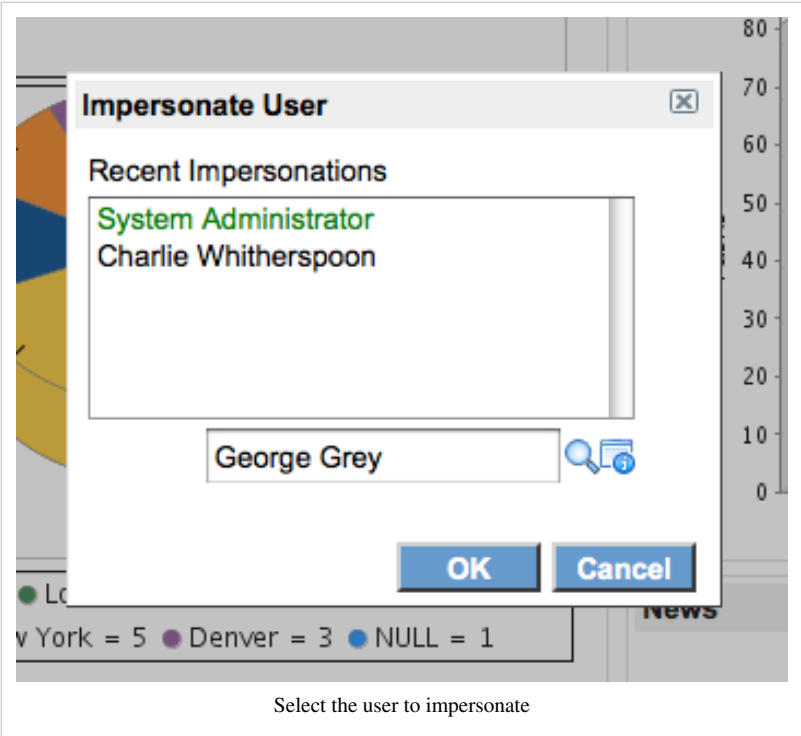
## Impersonating A User

1. Click the impersonate icon. (A dialog box appears)



Click the key icon to impersonate another user

2. Select the user from the **Recent Impersonations** list, click the lookup icon and select the user's name from the full list, or type the user's name.



3. Click **OK**.

## Impersonating a User on a Mobile Phone

The impersonation icon is not visible in the mobile view of the platform, and impersonating is not supported for mobile phones. For most mobile phones, however, it is possible to impersonate a user by switching to standard view, performing the impersonation (see above), and switching back to mobile view. Some mobile devices may have problems rendering the Impersonation dialog.

## Invoking or Modifying the Impersonate Button

The Impersonate button and its effects are contained in a UI Macro called `impersonate_button`. Modifying the `impersonate_button` is not recommended.

## Enable/disable the Impersonate Button

The impersonation capability can be enabled/disabled with the `glide.ui.impersonate_button.enable` UI Property, "Enable impersonation button in banner line".

## Logging

Impersonations are logged in the System Log. Logging can be enabled/disabled with the `glide.sys.log_impersonation` property.

Log	New	Go to	Created			1 to 2 of 2
	Created	Level	Message	Source		
	2009-12-16 16:11:21	Information	Impersonation end: James Capaldo (james.capaldo)	Impersonate		
	2009-12-16 16:09:36	Information	Impersonation start: Beth Anglin (beth.anglin) by: James Capaldo (james.capaldo)	Impersonate		

## Forcing Logout

In some cases, impersonating a user might cause an issue that makes it difficult to switch back (e.g. if in a test environment, the user is being presented with a broken page). To return to the user, go to `http://instance.service-now.com/logout.do` and log back in.



# Skills Management

---

## Overview

The Skills Management plugin enables an administrator to assign configured competencies, called *skills*, to groups or individual users. These skills can then be used to determine who can be assigned to particular tasks.

Skills can contain other skills. Any access granted to a parent skill will be granted to any skill that it contains. Once a skill is assigned to a group, all members of the group automatically inherit that skill and any others contained within it. The skills mechanism is similar to ServiceNow role management.

## Enhancements

### Calgary

The following enhancement is added in the Calgary release:

- Skills can now be related to models. This is especially useful in the Work Management application.

## Activating the Plugin

The plugin is automatically activated when the following applications are activated:

- Field Service Management (Berlin and earlier releases)
- Work Management (Calgary release)
- Project Management v2 Plugin

Administrators can also activate the Skills plugin manually.

**Click the plus to expand instructions for activating a plugin.**

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

If the plugin depends on other plugins, these plugins and their activation status are listed.

3. [Optional] Select the **Load demo data** check box.

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

4. Click **Activate**.

## Creating Skills

1. Navigate to **Skills > Skills**.
2. Click **New**.
3. Enter a unique, descriptive **Name**.
4. Enter a **Description** of the skill.
5. Click **Submit**.
6. Reopen the Skill record.
7. [Optional] Use the **Contains Skills** related list to add sub-skills.
8. [Optional] Use the **Models** related list to add any models that should be associated with the skill (Calgary release).

The **Users** related list contains users (based on their User record or the groups they belong to) who have the skill and sub-skills named in this record.

**Skill** [Update] [Delete]

Name: Mac OS

Active: ☒

Description:

[Update] [Delete]

**Contains Skills** [New] [Edit...] Go to: Name [ ] [Q] 1 to 1 of 1

▶ Skill = Mac OS

Name	Contains
Mac OS - Mac OS Installation/Setting	Mac OS Installation/Setting

Actions on selected rows... 1 to 1 of 1

**Users** [New] [Edit...] Go to: Active [ ] [Q] 1 to 4 of 4

▶ Skill = Mac OS

Active	Included in skill	Included in skill instance	Inherited	Inherited from	User
<input checked="" type="checkbox"/>	true		false		Marisa Smiler
<input checked="" type="checkbox"/>	true		false		Karen Zombo
<input checked="" type="checkbox"/>	true		false		Abel Tuter
<input checked="" type="checkbox"/>	true		false		David Dan

Actions on selected rows... 1 to 4 of 4

**Models** [New] [Edit...] Go to: Model [ ] [Q] 1 to 3 of 3

▶ Skill = Mac OS

Model
Apple MacBook Pro 17"
Apple MacBook Air 13"

## Assigning Skills

Assign skills to individual users or to groups. Members of a group inherit all the skills configured for their group.

## User Skills

If you assign a skill that contains other skills to a user, the user automatically inherits the contained skills. To do the following procedure, you must activate the Work Management plugin.

1. Navigate to **Skills > Users**.
2. Select a user from the list.
3. In the User record, select the **Skills** related list.
4. Click **Edit** and select one or more existing skills from the slushbucket.
5. Click **Save**.

User

Update

Delete

User ID:

karen.zombo

Email:

karen.zombo@example.co

First name:

Karen

Notification:

Enable

Last name:

Zombo

Calendar integration:

Outlook

Title:

IT Technician

Time zone:

System (America/Los\_Angeles)

Department:

IT

Business phone:

Password:

Mobile phone:

Password needs reset:

☐

Photo:

Click to add...

Locked out:

☐

Active:

☒

Web Service Access Only:

☐

Date format:

System (yyyy-MM-dd)

Update

Delete

Related Links

[Notification Preferences](#)

Roles

Groups

Delegates

Skills (12)

Locations Covered (1)

Skills

New

Edit...

Go to

Skill

1 to 12 of 12

User = Karen Zombo

Skill	Inherited
Apple Computers	false
Mac OS Installation/Setting	true
DELL Desktops	true
MacBooks	true
Desktops	false
DELL Laptops	true

## Group Skills

If you assign a skill this skill contains other skills to a group, the group and all its members automatically inherit the contained skills.

1. Navigate to **Skills > Groups**.
2. Select a group from the list.
3. In the Group record, select the **Skills** related list.
4. Click **Edit** and select one or more existing skills from the slushbucket.
5. Click **Save**.

The skill is added to the group and all the group members who are granted this skill are listed at the top of the form.

The screenshot displays the 'Group' configuration window for 'Field Services'. It includes a list of skills being granted to group members, a form for group details (Name, Manager, Type, Group email, Parent), and a table of skills assigned to the group.

**Group Details:**

- Name: Field Services
- Manager: Elvira Blumenthal
- Type: catalog
- Group email: [empty]
- Parent: [empty]

**Skills Table:**

Created	Skill	Inherits
2013-02-01 07:54:26	Canon Printers	true

## Filtering Potential Assignees Based On Skills

In the base system, field service orders (versions prior to Calgary), work management tasks (Calgary release), and project tasks use skills to filter assignments. If a skill is identified in the **Skill** field, only groups or users with the appropriate skill can be assigned to the task.

The Skills Management plugin contains a script include that builds a qualifier based on the assignment group and required skills for the task. For example, the **Assigned To** field on the Project Task record uses the following reference qualifier (using a **dictionary override**):

```
javascript:var util = new SkillsUtils();
util.assignedToRefQual(current);
```

This results in the following:

- If an **Assignment group** is set, the list is filtered on members of that group.
- If **Skills** are set (the **Skills** field may need to be added to the form), the list is filtered on users with all the skills selected.
- If **Assignment group** and **Skills** are both set, the list is filtered on group members with the defined skills.

You can introduce the same behavior to other task tables by using the same reference qualifier.

# Defining Locations

---

## Overview

Location records store the address and contact details for each site in your organization. Use location records to identify where records in the system are situated. For example, you can specify that an email server configuration item (CI) is located in your New York office or that a facilities request to fix a broken thermometer is for your Boston office.

Location records are stored on the Location [cmn\_location] table.

## Location Hierarchy

You can configure location records in a parent-child hierarchy to provide varying levels of detail. This gives you the option of choosing the right level of location detail for different records in the system. For example, the location of an email server CI could be **Second Floor**, which is the lowest-level child record in the following location hierarchy:

- Americas
  - New York
    - New York City
      - New York Datacenter
        - Second Floor

The email business service CI, which is responsible for handling all email for the entire New York region, could be associated with a location higher in the hierarchy, such as **New York**.

## Defining a Location

1. Navigate to **Organization Management > Locations**.
2. Fill in the form fields, as appropriate (see table).
3. Click **Submit**.

Field	Input Value
Name	The name of the location. This is the display value that the location will use when referenced on forms.
Street	The street address of the location, if applicable.
City	The city of the location, if applicable.
Zip / Postal Code	The zip or postal code of the location, if applicable.
Country	The country of the location, if applicable. Countries will be suggested.
Contact	A reference to a user who would be a contact for the location, if applicable.
Phone	The phone number for the location, if applicable.
Fax	The fax number for the location, if applicable.
Parent	A reference to the location record where this location can be found. See above to see how this establishes the location hierarchy.
Latitude	The latitude of the location, if applicable. This field is populated by a business rule. Deactivate this business rule to prevent the system from overwriting any values populated in the field manually.
Longitude	The longitude of the location, if applicable. This field is populated by a business rule. Deactivate this business rule to prevent the system from overwriting any values populated in the field manually.

*Fields which can be added by personalizing the form:*

Company	A reference field to the <b>Company</b> [ <b>cmn_company</b> ] table
Full Name	A read-only, calculated field that assembles the parent hierarchy of the location into a full name.
Stock Room	A boolean field that identifies whether the location is being used as a stock room.
Time Zone	A drop-down field to select the location's time-zone. By default, the location uses the system time zone. For more information, see <a href="#">Using Time Zones</a> .

## Map Location

The latitude and longitude fields are populated by a business rule (**get\_lat\_long**) which queries Google Maps. The more specific the location is, the more accurate the latitude and longitude will be.

Once the latitude and longitude are populated, Map Pages can be defined that display locations in an interactive map. For more information, see [Using Map Pages](#).

# Creating Groups

## Overview

A group is a set of users who share a common purpose. Groups may perform tasks such as approving change requests, resolving incidents, receiving email notifications, or performing work order tasks. Any business rules, assignment rules, system roles, or attributes that refer to the group apply to all group members automatically. Users with the `user_admin` role can create and edit groups.

## Creating Groups

1. Navigate to **User Administration > Groups**.
2. Click **New**.
3. Fill in the form.

To see some of the fields, you may need to personalize the form.

Field	Description
Name	Name of the group.
Manager	Group manager or lead.
Type	<p>Category for this group. For example, a group designated as type <b>catalog</b> is a service catalog group and can also be accessed under the <b>Service Catalog &gt; Catalog Policy &gt; Fulfillment Groups</b> module.</p> <p>You may need to personalize the form to add the <b>Type</b> field. Activating the Work Management plugin (Calgary release) adds the <b>Type</b> field automatically.</p> <p>See also <a href="#">Configuring Group Types for Assignment Groups</a>.</p>
Group email	Group email distribution list or the email address of the group's point of contact, such as the group manager.
Parent	<p>Other group of which this group is a member. If a group has a parent, the child group inherits the roles of the parent group. The members of the child group are considered members of the parent group.</p> <p><b>Note:</b> The <b>Assignment group</b> and <b>Assigned to</b> fields on incidents have special logic that prevents setting the <b>Assigned to</b> field to a user not defined directly in the assignment group. Therefore, only users defined in the assignment group, and not members of the assignment group's child groups, can be entered in the <b>Assigned to</b> field.</p>

Active	Check box that indicates whether the group is active or inactive. Inactive groups still appear in any reference field that already references the group, but are not visible by non-admin users in: <ul style="list-style-type: none"><li>• lists of groups</li><li>• the reference lookup list for reference fields</li><li>• the autocomplete list of groups displayed when you type into a reference field</li></ul>
Exclude manager	Check box that controls whether the group's manager receives email notifications.
Include members	Check box that controls whether the group members receive individual emails when someone sends an email to the <b>Group Email</b> address.
Description	Helpful information about the group.

## Adding Users to Groups

After defining a group, add users to the group.

1. Navigate to **User Administration > Groups**.
2. Click a group **Name**.
3. In the **Group Members** related list, click **Edit**
4. Select one or more names in the **Collection** list.
5. Click **Add**.
6. Click **Submit**.

## Removing Users from Groups

You can remove users from a group at any time.

1. Navigate to **User Administration > Groups**.
  2. Click a group **Name**.
  3. In the **Group Members** related list, select the check box next to a group member name.
  4. From the **Actions on selected rows** menu, select **Delete**.
-

# Associating Users to Groups

---

## Overview

When you add users to ServiceNow, make sure that each user is associated with a group. Consider which fields are mandatory. Full, complete user profiles are the most useful. Use a unique user ID when creating new profiles or updating existing profiles. If all logs are updated by the **admin** user, it becomes difficult to track what was configured and by whom. Consider creating an ITIL-based role for each administrator for these types of tasks. To import large numbers of users at once, consider using import sets.

## Creating a User

1. Navigate to **User Administration > Users**.
2. Click **New**.
3. Enter the user's information (see table).
4. [Optional] Personalize the form to add the **Schedule** field and assign a schedule to the user.
5. Click **Submit**.

The new user record appears at the top of the list.

Field	Input Value
User ID	Create a unique identifier for this user's ServiceNow login user name. Typical examples of user IDs are <b>cwitherspoon</b> and <b>charlie.witherspoon</b> . You cannot create a new user whose User ID duplicates an existing user (Berlin release). If you do import duplicates from an update set, the more recently created names takes the duplicate User ID.
First name	Enter the user's full first name.
Last name	Enter the user's last name.
Title	Enter a title or job description, or select one from the list.
Department	Select the user's department from the list.
Password	Assign a password to the user. This password can be permanent or temporary.
Password needs reset	Select this check box to require the user to change the password during the first login.
Locked out	Select this check box to lock the user out of the instance and terminate all their active sessions.
Active	Select this check box to make this user active. Only the administrator sees inactive user in: <ul style="list-style-type: none"> <li>• Lists of users</li> <li>• The selection list on reference fields (magnifying glass icon)</li> <li>• The auto-complete list that appears when you type into a reference field</li> </ul>
Email	Enter the user's email address. To enter a non-standard email address that does not pass field validation, you must deactivate the validation script first. <ol style="list-style-type: none"> <li>1. Navigate to <b>System Definition &gt; Validation Scripts</b>.</li> <li>2. Select the <b>email</b> record.</li> <li>3. Clear the <b>Active</b> check box and save the change.</li> <li>4. Complete the user profile, including the email address, and update or submit the record.</li> <li>5. Reactivate the email validation script.</li> </ol>
Notification	Select the type of notification to send to this user. The default is <b>Email</b> . If you select <b>None</b> , the user can still receive notifications if he or she subscribes to the notification or is specified as a recipient in the Email Notifications form.  To prevent notification completely, set a condition on the Email Notification form itself that does not deliver the notification if this field is set to <b>None</b> .



Calendar integration	Select <b>Outlook</b> to have this user receive meeting notifications via email directly to the calendar. Otherwise, select <b>None</b> .
Time zone	Select the user's time zone.
Business phone	Enter this user's business phone number.
Mobile phone	Enter this user's mobile phone number.
Photo	Attach a photo of the user, if appropriate.

## Associating the User to a Group

1. Navigate to **User Administration > Groups**.
2. Click the group to which you want to assign the user.
3. In the **Group Members** related list, click **Edit**.
4. Select the user in the **Collection** list, and then click **Add**.
5. Click **Save**.

## Assigning Roles to the User

A user automatically inherits roles from all groups the user belongs to. These roles cannot be deleted from the user's record, only from the group's record. Roles can also be associated directly with the user. If a user has the same role assigned more than once, such as from multiple groups, the role appears multiple times in the **Roles** related list on the user record.

To add roles to a user's record:

1. Navigate to **User Administration > Users**.
2. Open a user's record.
3. In the **Roles** related list, click **Edit**.
4. Select the desired roles in the **Collection** list, and then click **Add**.
5. Click **Save**.

## Allow Users to View Their Profile

Users are able to view their profile by clicking their name in the **Welcome** banner. If your users cannot do this, enable a system property:

1. Navigate to the System Properties [sys\_properties] table.
2. Search for the **glide.ui.welcome.profile\_link** property.
3. Set the value to **true**.

## Enhancements

### Berlin

- The User ID field [sys\_user.user\_name] requires unique values. You cannot create a new user whose User ID duplicates an existing user.

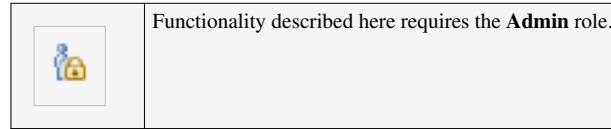
---

# User Security

---

## Granting Access

---



## Overview

The contextual security manager provides incredible flexibility and power to protect information by controlling read/write/create/delete authorization. Key advantages include:

- Contextual Security -- Secure a record based on its contents
- Hierarchical Security -- Can apply security rules to any level in our object hierarchy

## Differences between Contextual Security and Simple Security

Everything you can do with the simple security manager you can also do with the contextual security manager. Likewise, after conversion to the contextual security manager, you should not see any behavior changes in your instance. However, on a go forward basis, the process of security a small number of resources has changed.

## Things that have changed

### Securing Fields and Tables

Under the simple security manager, you could secure fields and tables by adding roles to the appropriate dictionary entry. After installing the contextual security manager, these dictionary roles are no longer tested. Instead the system looks for ACL rules on fields and or tables.



**Note:** After you install the Contextual Security Manager you must secure fields and tables via ACL rules. Even if you personalize the dictionary form and add roles to a dictionary entry, no change in rights will occur.

### Granting Roles to Users

Roles can still be granted to users or groups using the same logic as under the simple security manager. The one noteworthy exception is that the "roles" field on the user record is no longer checked under the contextual security manager (and should be, in fact, removed from your user and group forms upon installation).



**Note:** To add roles to a user or group record under Contextual Security you must add them to the Roles related list instead of to the user or group record itself.

---

## Things that have not changed

### Applications and Modules

Applications and modules both contain lists of roles under which they can be viewed. For example, the System Definition application requires the admin role to be viewed.

Security rights for Applications and Modules are still defined via these role arrays although they may be transitioned to ACLs at some future date.

### Catalog Items and Variables

Both catalog items, and catalog variables contain lists of roles under which they can be viewed.

Security rights for these entities are still defined via these role arrays although they may be transitioned to ACLs at some future date.

### Inheritability of Group Roles

Under the contextual security manager, a group still automatically inherits any role granted to the group.



**Note:** The role's *inherits* flag is set 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. \*.\*

## 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.

## 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.

# Using Access Control Rules

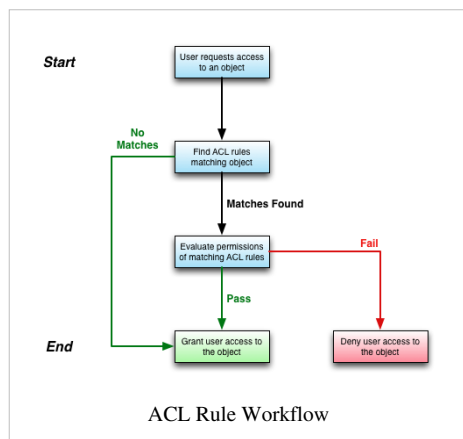
## Overview

ServiceNow uses access control list (ACL) rules, also called access control rules, to control what data users can access and how they can access it. ACL rules require users to pass a set of requirements in order to gain access to particular data. Each ACL rule specifies:

- The **object** being secured
- The **permissions** required to access the object

ServiceNow searches for ACL rules that **match** the object the user wants to access. If there are no matching ACL rules for the object, then the object does not require any additional security checks. By default, ServiceNow provides ACL rules to restrict access to all database and personalization operations.

After finding a matching ACL rule, ServiceNow *evaluates* if the user has the permissions required to access the object. If a user meets the ACL rule permissions, the instance grants the user access to the object. If a user does not meet the ACL rule permissions, the instance denies the user access to the object.



Users with access to the security\_admin role can:

- Create ACL rules to secure new objects
- Update existing ACL rules to grant or deny users access to objects based on their business requirements
- Debug ACL rules to determine why users cannot access certain objects

## Enhancements

## Calgary

The following enhancements are available as of the Calgary release:

- You can create ACL rules for processors.
- You can create ACL rules for client-callable script includes.

## Creating ACL Rules

Create custom ACL rules in order to secure access to new objects or to change the default security behavior. To create new ACL rules, you must elevate privileges to the security\_admin role.

1. Elevate privileges to the security\_admin role.
2. Navigate to **System Security > Access Control (ACL)**.
3. Click **New**.
4. Define the object the ACL rule secures and the permissions required to access the object. See Access Control Fields.
5. Click **Submit**.



**Note:** The *Requires roles* related list is available only after you save the ACL rule.

## Access Control Fields

Access control records use the following fields.

Field	Description
Type	Select what kind of object this ACL rule secures. An object's type determines how the object is named and what operations are available.
Operation	Select the operation this ACL rule secures. Each object type has its own list of operations. An ACL rule can only secure one operation. To secure multiple operations, create a separate ACL rule for each.
Name	Enter the object's record name or select the object's table and field names. The name identifies the object being secured. The more specific the name is, the more specific the ACL rule is. You can use the wildcard character asterisk (*) in place of a record name, table name, or field name to select all objects that match a particular record type, all tables, or all fields. You cannot combine a wildcard character and a text search. For example, inc* is not a valid ACL rule name, but incident.* and *.number are valid ACL rule names.
Active	Select this check box to have ServiceNow enforce this ACL rule.
Admin Overrides	Select this check box to have users with the admin role automatically pass the permissions check for this ACL rule, regardless of what script or role restrictions would apply. Clear this check box if administrators must meet the permissions defined in this ACL rule to gain access to the secured object. Since administrators will always pass role checks (see the description of the <b>Requires role</b> field), use the condition builder or <b>Script</b> field to create a permissions check that administrators must pass.
Description	[Optional] Enter a description of the object or permissions this ACL rule secures.
Condition	Use this condition builder to select the fields and values that must be true for users to access the object.
Script	<p>Enter a custom script describing the permissions required to access the object. The script can use the values of the <b>current</b> and <b>previous</b> global variables as well as system properties. The script must generate a true or false response in one of two ways:</p> <ul style="list-style-type: none"> <li>• return an <b>answer</b> variable set to a value of true or false</li> <li>• evaluate to true or false</li> </ul> <p>In either case, users only gain access to the object when the script evaluates to true and the user meets any conditions the ACL rule has. Both the conditions and the script must evaluate to true for a user to access the object.</p>

Requires role	Use this related list to specify the roles a user must have in order to access the object. If you list multiple roles, a user with any one of the listed roles can access the object. <b>Note:</b> Users with the admin role will always pass this permissions check because ServiceNow automatically grants admin users all roles.
---------------	---

## Granting or Denying Access

When a user attempts to access a particular object, ServiceNow first searches for ACL rules that match the requested object's type and operation. From this list, ServiceNow then searches for an ACL rule that matches the object's name. If an ACL rule matches the object's name, then the user must meet the permissions described in this rule to access the secured object.

If the user fails to meet the permissions required by the first rule, ServiceNow searches for the next ACL rule that matches the object's name. For each matching ACL rule, the user must meet the required permissions in order to access the object. ServiceNow stops searching for matching ACL rules after the user meets the minimum required permissions for the current object type. If the user does not meet the permission requirements in any matching ACL rule, the instance denies the user access to the object.

The effects of the being denied access to an object depend on the ACL rule that the user failed. For example, failing a read operation ACL rule prevents the user from seeing the object. Depending on the object secured, the ACL rule could hide a field on a form, hide rows from a list, or prevent a user from accessing a particular UI page. See the table for a complete list of results of failing an ACL rule for a given operation and object type.

Operation	Results of Failing an ACL Rule on Object
execute	User cannot execute scripts on record or UI page.
create	User cannot see the <b>New</b> UI action from forms. The user also cannot insert records into a table using API protocols such as web services. Note that a create ACL with a condition that a field contain a specific value always evaluates as false, as fields on new records are considered empty until saved.
read	User cannot see the object in forms or lists. The user also cannot retrieve records using API protocols such as web services.
write	User sees a read-only field in forms and lists, and the user cannot update records using API protocols such as web services.
delete	User cannot see the <b>Delete</b> UI action from forms. The user also cannot remove records from a table using API protocols such as web services.
edit_task_relations	User cannot extend the task table.
edit_ci_relations	User cannot extend the Configuration Item [cmdb_ci] table.
save_as_template	User cannot see the UI action to save a record as a template.
add_to_list	User cannot view or personalize specific columns in the list mechanic.
list_edit	User cannot update records (rows) from a list.
report_on	User cannot create reports on the object.
personalize_choices	User cannot right-click a choice list field and select Personalize Choices.

## Matching ACL Rules to Objects

Each object type has its own matching requirements.

Object Type	Matching ACL Rules Required to Access Object	Existing Wildcard ACL Rules
Client-callable script includes	Users must meet the permissions of two ACL rules:	By default, there are no wildcard (*) rules for these object types. If you create a wildcard ACL rule for one of these objects, then the ACL rule applies to all objects of this type.
Processors	1. <b>All</b> wildcard ACL rules for the object (if any ACL rule exists for the operation).	
UI pages	2. The <b>first</b> ACL rule that matches the object's name (if any ACL rule exists for the operation).	
Record	Users must meet the permissions of two ACL rules:	By default, ServiceNow provides wildcard table rules (*) for the create, read, write, and delete operations and provides wildcard field rules (*.*) for the personalize_choices, create, and save_as_template operations. When you create a new table, create new ACL rules for the table unless you want to use the provided wildcard ACL rules.
	1. The <b>first</b> ACL rule that matches the record's field (if any ACL rule exists for the operation).	
	2. The <b>first</b> ACL rule that matches the record's table (if any ACL rule exists for the operation).	



**Note:** ServiceNow uses the high security property *Security manager default behavior* (`glide.sm.default_mode`) to determine whether users can access objects that only match against wildcard table ACL rules. When this property is set to **Deny access**, only administrators can access objects that match the wildcard table ACL rules.

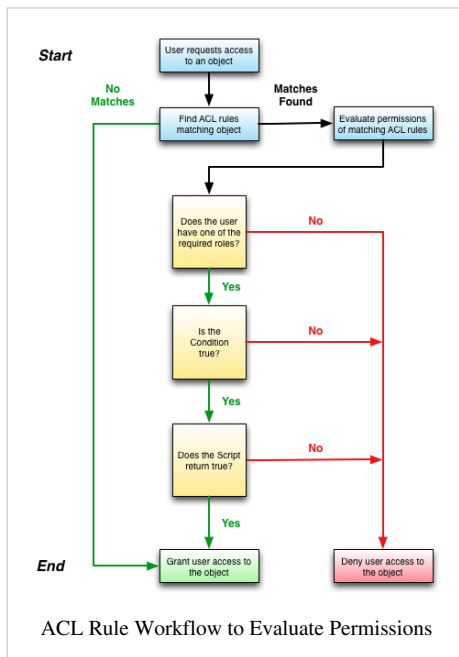


**Note:** The wildcard field ACL rule (\*.\*) for the create operation reuses the same permissions as the write operation. This means that the create permissions are the same as the write permissions unless you define an explicit create operation ACL rule.

## Evaluating ACL Rule Permission Requirements

An ACL rule only grants a user access to an object if the user meets **all** of the permissions required by the matching ACL rules.

- The condition must evaluate to **true**.
- The script must evaluate to **true** or return an answer variable with the value of **true**.
- The user must have one of the roles in the required roles list.
- The other matching ACL rules for the object type must evaluate to **true**.



## Record ACL Rules

Record ACL rules consist of two parts:

- **Table name:** the table being secured. If other tables extend from this table, then the table is considered a parent table. ACL rules for parent tables apply to any table that extends the parent table.
- **Field name:** the field being secured. Some fields are part of multiple tables because of table extension. ACL rules for fields in a parent table apply to any table that extends the parent table.

ACL rules can secure the following record operations:

Operation	Description
execute	Allows users to run an application or script.
create	Allows users to insert new records (rows) into a table.
read	Allows users to display records from a table.
write	Allows users to update records in a table.
delete	Allows users to remove records from a table or drop a table.
edit_task_relations	Allows users to extend the Task table.
edit_ci_relations	Allows users to extend the Configuration Item [cmdb_ci] table.
save_as_template	Allows users to save a record as a template.
add_to_list	Allows users to insert records (rows) into a table from a list.
list_edit	Allows users to update records (rows) from a list.
report_on	Allows users to create reports on the table.
personalize_choices	Allows users to personalize the table or field.



## Processing Order for Record ACL Rules

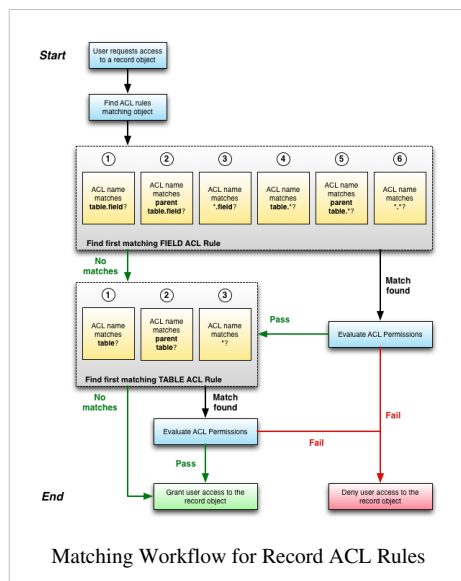
ServiceNow processes ACL rules in the following order:

1. Match the object against field ACL rules.
2. Match the object against table ACL rules.

This processing order ensures that users gain access to more specific objects before gaining access to less specific ones.

A user must pass **both** field and table ACL rules in order to access a record object.

- If a user fails a field ACL rule but passes a table ACL rule, ServiceNow denies the user access to the field described by the field ACL rule.
- If a user fails a table ACL rule, ServiceNow denies the user access to all fields in the table even if the user previously passed a field ACL rule.



### Field ACL Rules

ServiceNow processes field ACL rules in the following order:

1. Match the *table* and *field* name. For example, **incident.number**.
2. Match the *parent table* and *field* name. For example, **task.number**.
3. Match *any table (wildcard)* and *field* name. For example, **\*.number**.
4. Match the *table* and *any field (wildcard)*. For example, **incident.\***.
5. Match the *parent table* and *any field (wildcard)*. For example, **task.\***.
6. Match *any table (wildcard)* and *any field (wildcard)*. For example, **\*,\***.

The first successful evaluation stops ACL rule processing at the field level. This means that when a user passes a field ACL rule's permissions, ServiceNow grants the user access to the record object

secured by the field ACL rule and stops searching for matching field ACL rules. For example, if a user passes the field ACL rule for incident.number, ServiceNow stops searching for rules that secure the Number field and grants the user access to the field. If a user passes the field ACL rule for task.number, ServiceNow stops searching for matching ACL rules and grants the user access to the number field in the parent table and also to extended tables that use the field.

### Table ACL Rules

In most cases there is not an individual field ACL rule for every field in the table the users is trying to access. If no field ACL rule matches the record object, the user must pass the table ACL rule. Since ServiceNow provides wildcard table ACL rules that match every table, the user must always pass at least one table ACL rule. ServiceNow provides additional table ACL rules to control access to specific tables.

ServiceNow processes table ACL rules in the following order:

1. Match the *table* name. For example, **incident**.
2. Match the *parent table* name. For example, **task**.
3. Match *any table name (wildcard)*. For example, **\***.

Just like with field ACL rules, ServiceNow grants the user access to the record object secured by the ACL rule and stops searching for matching ACL rules the first time a user passes a table ACL rule's permissions. A user who passes the table ACL rule for incident has access to all fields in the Incident table. A user who passes the table ACL

rule for task has access to all fields in the Task table as well as the fields in extended tables. A user who passes the table ACL rule for any table has access to all fields in all tables.

## Multiple ACL Rules at the Same Point in the Processing Order

If ServiceNow matches two or more rules at the same point in the processing order, the user must pass any one of the ACL rules permissions in order to access the object. For example, if you create two field ACL rules for incident.number, then a user who passes one rule has access to the number field regardless of whether the user failed any other field ACL rule at the same point in the processing order.

## UI Page ACL Rules

UI page ACL rules specify the UI page to be secured. Use the asterisk character as a wildcard to search for any UI page. For a list of available UI pages, navigate to **System UI > UI Pages**.

ACL rules can secure the following UI page operations:

Operation	Description
execute	Allows users to run an application or script.
create	Allows users to insert new UI page records.
read	Allows users to display the UI page.
write	Allows users to update UI page records.
delete	Allows users to remove UI page records.
edit_task_relations	Allows users to extend the Task table.
edit_ci_relations	Allows users to extend the Configuration Item [cmdb_ci] table.
save_as_template	Allows users to save a UI page record as a template.
add_to_list	Allows users to insert UI page records from a list.
list_edit	Allows users to update UI page records from a list.
report_on	Allows users to create reports on UI page records.
personalize_choices	Allows users to personalize UI page records.

Since UI pages typically only display read-only information, the most common UI page ACL rule is for the "read" operation. For an example of limiting access to live feed with this type of rule, see [Limiting Live Feed Access by Role](#).

## Processor ACL Rules

ACL rules can secure access to the *execute* operation of all or specific processors (Calgary release). Processor ACL rules specify the processor you want to secure. Use the asterisk character as a wildcard to search for any processor. For a list of available processors, navigate to **System Definition > Processors**.

By default, ServiceNow includes an ACL rule for the EmailClientProcessor to restrict the email client to users with the itil role. See [Enabling the Email Client](#) for more information.

## Client-Callable Script Include ACL Rules

ACL rules can secure access to the *execute* operation of all or specific client-callable scripts (Calgary release). Script include ACL rules specify the client-callable script include to be secured. Use the asterisk character as a wildcard to search for any client-callable script include. For a list of available script includes, navigate to **System Definition > Script Includes**. You can personalize the list to show the **Client callable** column.

By default, ServiceNow does not include any ACL rules for client-callable script includes.

## Debugging

ServiceNow offers the following ACL rule debugging tools:

- Field level debugging
- ACL rule output messages

To enable ACL rule debugging, navigate to **System Security > Debug Security Rules**.



**Note:** Impersonation can simplify debugging ACL rules. First enable ACL debugging, then impersonate another user to see what ACL rules the user passes and fails.

## Field Level Debugging

With debugging enabled, ServiceNow displays a small *bug* icon next to each field with an ACL rule. Clicking the icon lists the ACL rules that apply for the field and the evaluation results.

The screenshot shows the 'Incident' form in ServiceNow. At the top, there is a breadcrumb 'Incident' with a back arrow and an 'Update' button. Below this, the 'Number' field is populated with 'INC0000002'. The 'Caller' field has a small bug icon next to it. Below the fields, a list of ACL rules is displayed for the 'Caller' field:

- record/incident.caller\_id/read = true (0:00:00.000) - Green bar with a checkmark icon.
- record/incident.caller\_id/write = true (0:00:00.000) - Green bar with a checkmark icon.
- FIELD : incident.caller\_id/write = false (0:00:00.000) - Orange bar with a red 'X' icon.
- FIELD : incident.caller\_id/write = true (0:00:00.000) - Green bar with a checkmark icon.

## ACL Rule Output Messages

ServiceNow displays ACL rule output messages at the bottom of each list and form. The output message lists the ACL rule name, the permissions required, and the evaluation result (pass or fail).

### Debug Output

```

12:57:50.214: TIME = 0:00:00.000 PATH = processor/XMLHttpRequest/execute RULE = (()) RC = true
12:57:50.216: TIME = 0:00:00.000 PATH = processor/com.glide.processors.xmlhttp.AJAXEvaluator/execute RULE = (()
SEQ () SEQ () SEQ () SEQ ()) RC = true
12:57:50.222: TIME = 0:00:00.000 PATH = client_callable_script/include/AJAXClientTiming/execute RULE = (()) RC =
true
12:57:50.225: TIME = 0:00:00.000 PATH = processor/XMLHttpRequest/execute RULE = (()) RC = true
12:57:50.227: TIME = 0:00:00.000 PATH = processor/com.glide.ui_list_edit.AJAXListEdit/execute RULE = (()) SEQ ()
SEQ () SEQ ()) RC = true
12:57:50.233: TIME = 0:00:00.000 PATH = record/incident.number/read RULE = (((hasRole() AND script=typeof
g_approval_form_request != "undefined" && g_approval_form_request == true;) OR (hasRole(itol) ) OR (hasRole() AND
script=current.opened_by == gs.getUserID() || current.caller_id == gs.getUserID() ||
current.watch_list.indexOf(gs.getUserID()) > -1;)) SEQ ((hasRole() AND script=gs.hasRole('admin') ||
gs.getProperty('glide.sm.default_mode') == 'allow') OR (hasRole() AND script=var ra = false; if
(root_rule.substring(0,7)=='var_m_') { var wa = gs.getUser().hasRole('workflow_admin'); var wc =
gs.getUser().hasRole('workflow_creator'); var fn = gs.getUser().hasRole('normalizer'); if (wa || wc || fn) ra = true; } ra; ) OR
(hasRole() AND script=typeof g_approval_form_request != "undefined" && g_approval_form_request == true;)))) RC = true
12:57:50.233: TIME = 0:00:00.000 PATH = record/incident.number/list_edit RULE = () RC = true
12:57:50.233: TIME = 0:00:00.000 PATH = record/incident.number/write RULE = (((hasRole(itol) ) OR (hasRole() )) SEQ
((hasRole() AND script=gs.hasRole('admin') || gs.getProperty('glide.sm.default_mode') == 'allow') OR (hasRole() AND
script=var ra = false; if (root_rule.substring(0,7)=='var_m_' || root_rule == 'wf_workflow.vars') { ra =
gs.getUser().hasRole('workflow_creator') || gs.getUser().hasRole('normalizer'); } ra; ))) AND (((hasRole(admin) )))) RC = false

```

## Troubleshooting

Here is a list of common ACL rule errors and their solutions. Enable debugging to help troubleshoot an issue.

Error or Symptom	Solution
You cannot access records from a custom table.	Create a table ACL rule for the custom table granting users access to the table. Without an explicit table ACL rule, users must pass the permissions in the table wildcard (*) ACL rule, which by default restricts access to administrators only. Enable debugging and determine what ACL rules are evaluated for the custom table.
You create a custom ACL rule that does not work properly.	The most likely problems are that another rule takes precedence over your custom rule in the processing order or that the user does not meet all the permission requirements for the object type. Enable debugging and verify that the ACL rule is being evaluated.
Your field ACL rule does not work properly.	There is likely a table ACL rule that the user has not met. Enable debugging and determine what ACL rules are evaluated for the field. Verify that there is not a conflicting table ACL rule or duplicate field ACL rule.
Your table ACL rule does not work properly.	There is either an ACL rule higher in the processing order or a duplicate table ACL rule interfering with the table ACL rule. Enable debugging and determine what ACL rules are evaluated for the table.
You can see a field in a list but not in form.	It is possible that the ACL rule conditions or script are being triggered in the list but not in the form. Enable debugging and determine when the ACL rules evaluate to true. Update the conditions or script to have the same behavior on the list and form.
You receive an error message when trying to execute a processor or client-callable script include	There is an ACL rule for the processor or client-callable script include that the user has not met. If the user should have access to the object, enable debugging and determine what ACL rules are evaluated for the processor or script include. Update the ACL rule or the user roles as needed to access the object.

## Controlling Whether Script Conditions Apply to Reference Fields

By default, ACL rules ignore the script conditions of a table's reference fields. The default behavior is intended to improve instance performance. If you want to enable script conditions for reference fields, add the following system property.

Property	Description
glide.sys_reference_row_check	Controls whether the script conditions of Access Control Rules apply to a table's reference fields. <ul style="list-style-type: none"><li>• <b>Type:</b> true   false</li><li>• <b>Default value:</b> false</li><li>• <b>Location:</b> Add to the System Properties [sys_properties] table</li></ul>

# Additional Features and Applications

## Security Jump Start (ACL Rules) Plugin



Functionality described here requires the **Security Jump Start (ACL Rules)** plugin. The plugin is automatically installed for new instances.

### Overview

The Security Jump Start (ACL Rules) Plugin is installed automatically on all new instances. These rules were written to provide a jump start on securing many system tables, to make it easier for an organization to more quickly get into production.

This plugin is not intended for existing instances, as it might modify security access to tables that are already in use in a production environment. If an admin is interested in the new ACL rules provided by this plugin, one or more of them may be created manually in an existing instance as specific needs dictate. This list of ACLs may be used as a guideline in that case. Should an admin strongly want this plugin installed on an existing instance, we highly recommend the plugin be tested extensively in a test instance first, to ensure that the rules do not conflict with the operational needs of the organization's current implementation.

The following ACLs are included in this plugin. Click the icon in a header row to sort that column in ascending or descending order. The Operation key is as follows:

- R=read
- W=write
- D=delete
- C=create

Name	Operation	Description
cmdb_ci	WCD	asset or itil role required to write/create/delete Configuration Item records
cmn_department	WD	user_admin role required to write/delete Department records
cmn_location	WC	user_admin role required to write/create Location records
core_company	WD	user_admin role required to write/delete Company records
kb_knowledge	create	knowledge role required to created Knowledge records
ldap_ou_config	RWCD	user_admin role required to read/write/create/delete LDAP OU Definition records
ldap_server_config	RWCD	user_admin role required to read/write/create/delete LDAP Server records
process_guide	WCD	admin role required to writecreate/delete Process Guide records
process_step	WCD	admin role required to writecreate/delete Process Step records
sc_category	create	catalog_admin role required to create Service Catalog Category records
sc_category	delete	catalog_admin role required to delete Service Catalog Category records
sc_category	write	catalog_admin role required to write to Service Catalog Category records
sc_cat_item	write	catalog_admin role required to write to Catalog Item records

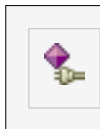
sc_cat_item	delete	catalog_admin role required to delete Catalog Item records
sc_cat_item	create	catalog_admin role required to create Catalog Item records
sysevent_email_action	read	all users can read Email Notification records (for subscription purposes)
sysevent_register	RWCD	admin role required to read/write/create/delete Event Registry records
sysevent_script_action	RWCD	admin role required to read/write/create/delete Script Action records
syslog	RWCD	admin required to read/write/create/delete Log Entry records
sysrule	RWCD	admin required to read/write/create/delete Rule records (Email Notifications, Inbound Email Actions, Approval Rules, etc.)
sysrule	read	all users can read Email Notification records for (subscription based notifications)
sys_app_application	WCD	admin required to write/create/delete Application records
sys_app_category	WCD	admin role required to write/create/delete Application Category records
sys_app_module	WCD	admin required to write/create/delete Module records
sys_audit	RWCD	admin required to read/write/create/delete Audit records
sys_dictionary	RWC	personalize_dictionary role required to read/write/create Dictionary records
sys_dictionary.*	read	personalize_dictionary role can read Dictionary fields
sys_documentation	delete	personalize_dictionary role required to delete Field Label records
sys_documentation	create	personalize_dictionary role required to create Field Label records
sys_documentation	write	personalize_dictionary role required to write to Field Label records
sys_gauge	RWCD	admin role required to read/write/create/delete Gauge records
sys_gauge_count	RWCD	admin role required to read/write/create/delete Gauge Count records
sys_group_has_role	read	itil role required to see Group Role records
sys_home	WCD	itil_admin role required to write/create/delete Welcome Page Section records
sys_installation_exit	WCD	admin role required to write/create/delete Installation Exit records
sys_job	WCD	admin role required to write/create/delete Sys Job records
sys_nav_link	WCD	admin role required to write/create/delete Navigation Link records
sys_perspective	WCD	admin role required to write/create/delete Menu List records
sys_portal	RWCD	admin role required to read/write/create/delete Portal records
sys_portal_page	RWCD	admin role required to read/write/create/delete Homepage records
sys_portal_preferences	RWCD	admin role required to read/write/create/delete Portal Preferences records
sys_processor	WC	admin role required to write/create Processor records
sys_properties	WC	admin role required to write/create System Property records
sys_properties_category	WCD	admin role required to write/create/delete Property Category records
sys_report	delete	roles that can delete Report records (does not restrict deleting through Report UI)
sys_report	write	roles that can write to Report records (does not restrict editing through Report UI)
sys_report	read	users can read their own Report records, those of their groups, and GLOBAL ones (does not affect viewing through Report UI)
sys_report	read	roles that can read Report records (does not restrict viewing through Report UI)
sys_reportroles	read	admin role required to read Report Roles records
sys_script	WCD	admin role required to write/create/delete Business Rule records
sys_script_ajax	WCD	admin role required to write/create/delete AJAX Script records

sys_script_client	WCD	admin role required to write/create/delete Client Script records
sys_script_include	WCD	admin role required to write/create/delete Script Include records
sys_security_acl	write	admin role required to write to Access Control records
sys_security_acl_role	create	admin role required to create Access Roles records
sys_security_acl_role	delete	admin role required to delete Access Roles records
sys_security_acl_role	write	admin role required to write to Access Roles records
sys_security_operation	delete	admin role required to delete Security Operation records
sys_security_operation	create	admin role required to create Security Operation records
sys_security_operation	write	admin role required to write to Security Operation records
sys_security_type	write	admin role required to write to Security Type records
sys_security_type	create	admin role required to create Security Type records
sys_security_type	delete	admin role required to delete Security Type records
sys_status	create	admin role required to create System Status records
sys_status	delete	admin role required to delete System Status records
sys_status	write	admin role required to write to System Status records
sys_template	write	template_editor role required to write to Template records
sys_template	create	template_editor role required to create Template records
sys_template	delete	template_editor role required to delete Template records
sys_template	read	template_editor role required to read Template Roles records
sys_ui_action	create	admin role required to create UI Action records
sys_ui_action	delete	admin role required to delete UI Action records
sys_ui_action	write	admin role required to write to UI Action records
sys_ui_action_view	write	admin role required to write to UI View Action records
sys_ui_action_view	create	admin role required to create UI View Action records
sys_ui_action_view	delete	admin role required to delete UI View Action records
sys_ui_policy	create	admin role required to create UI Policy records
sys_ui_policy	delete	admin role required to delete UI Policy records
sys_ui_policy	write	admin role required to write to UI Policy records
sys_ui_policy_action	create	admin role required to create UI Policy Action records
sys_ui_policy_action	delete	admin role required to delete UI Policy Action records
sys_ui_policy_action	write	admin role required to write to UI Policy Action records
sys_ui_script	write	admin role required to write to UI Script records
sys_ui_script	delete	admin role required to delete UI Script records
sys_ui_script	create	admin role required to create UI Script records
sys_user	write	Users with no role cannot update any user record but their own
sys_user_grmember	delete	user_admin role required to delete Group Member records
sys_user_grmember	write	user_admin role required to write to Group Member records
sys_user_group	create	Only itil and above can create group records
sys_user_group	write	Only itil and above can write to group records



sys_user_has_role	read	itil role required to see User Role records
sys_user_role	create	admin role required to create Role records
sys_user_role	delete	admin role required to delete Role records
sys_user_role	write	admin role required to write to Role records
sys_user_role_contains	read	itil role required to see Contained Role records
sys_user_role_contains	write	admin role required to write to Contained Role records
sys_user_token	RWCD	admin role required to read/write/create/delete User Token records

## Group On-Call Rotation Plugin



Functionality described here requires the **Group On-Call Rotation** plugin.

### Overview

Group on-call rotation provides a way of rotating an on-call position within a group of people on a regular basis. Escalation capabilities can tie into an on-call rotation, and the on-call position and escalation can both be used by business rules. There is a scripting API for use in business rules to easily access on-call rotation information. On-call rotation can help answer questions like the following:

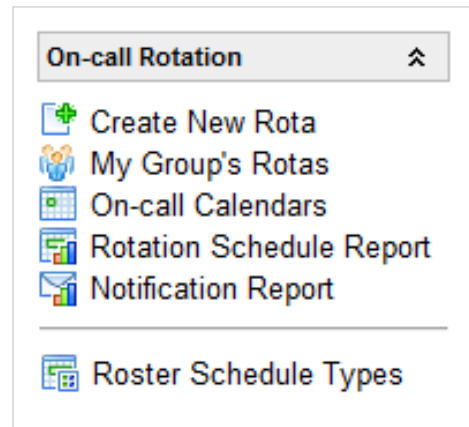
- *For a specific group, who is the primary contact person right now?*
- *Who is the primary contact at any given time?*
- *How do I escalate notifications for this group?*
- *When am I on-call for this group this year?*

### Concepts

- **Groups**- Standard groups in ServiceNow that serve as the basis for rotations
- **Rotas**- A rota in the Group On-Call Rotation application is the top level definition of on-call shift hour patterns, personnel lists, and notification rules for a group
- **Rosters**- Rosters are subsets of groups and determine who is part of a particular rotation for a group - a roster can contain only some group members
- **Calendars**- Provide information about currently defined rotations as well as an interface for manipulating these rotations
- **On-call Rotation**- An on-call rotation consists of a roster and a schedule to determine who is responsible for responding to incidents in a specific group

## On-call Rotation Plugin Modules

- **Create New Rota:** Wizard that simplifies the creation of new rosters.
- **My Groups Rotas:** Entry point for an end user of On-Call Rotation to see rosters that they are a part of.
- **On-call Calendars:** Provides information about currently defined rotations as well as an interface for manipulating these rotations.
- **Rotation Schedule Report:** Reporting mechanism for accessing information about On-Call rotations.
- **Notification Report:** Provides information on how an incident should be escalated for a certain roster.
- **Roster Schedule Types:** Used to define schedule templates that can be used in the Create New Rota wizard.



## Creating an On-Call Rotation

Navigate to Create New Rota. Select the group to which the rotation will correspond, select a start date, and select a schedule type. This associates a rotation (who) with a schedule type (when).

1. Navigate and locate the rotation you just added, and go to the associated roster (Rosters related list). Note if you are not a member of the rotation you just set up, you will have to change the list filter to drop the group condition, as it specifies the groups of which you are a member.
2. Check the time zone for this roster, and set it to the members' time zone if required. If the roster should begin at a particular time of day, clear **all day rotation**, and you will be able to specify a time.
3. Here you may change the members of the roster. Initially, they are populated from the group, but you can remove users that will not participate in the rotation by clicking the "edit" button and using the slushbucket to remove them from the selected list. Members will automatically be reordered, and the calendar will get updated. Note that you cannot add members to the roster who are not in the group.
4. You may add additional rosters to a rotation. For example, you may want to have a primary roster and a secondary roster. In this case, you can create a second roster, call it Secondary, and configure the members the same as the Primary roster, only stagger the order. This will ensure that the primary and secondary person are never the same.
5. Notification rules may also be specified on the rotation by going to the Notification rules related list on the rotation form.

For more details, see [Creating a New Roster](#).

## Scripting of On-Call Rotations

There is an on-call rotation scripting API for accessing on-call rotation information within any of the ServiceNow scripting components. This means all rotation information is available to business rules and other scripts without having to script additional GlideRecord queries. Using scripting it is possible to produce highly customized on-call rotation configurations related to after-hour incident assignments or any other configuration. For more detail, see [Scripting of On-Call Rotations](#).

Example business rules have been provided that demonstrate how the API can be used to automatically assign incidents to the on-call person for a group or to provide escalation notices to a group.

# On-call Calendars

On-call calendars provide a way of visualizing the on-call rotation for a group. Navigate to **On-Call Rotation > On-Call Calendars**. Initially, the display will default to the first group with a roster. Use the group drop-down to select the group in which you are interested.

Each timeslot specified by the schedule type for the group's roster will be displayed along with the on-call person assigned to that slot.

Use the 31, 7, and 1 buttons to change the calendar's display to monthly, weekly, and daily views. Use the left and right arrows to move the display back and forward in time. Use the calendar icon to move to a specific date.



# Rotation Schedule Report

To produce a report of on-calls for a period of time for one or more groups, navigate to On-Call Rotation -> Rotation Schedule Report. Select the start and end date for the report and select one or more groups. Click the “All groups” checkbox to list all the groups from which groups can be added. Alternately, leave “All groups” unchecked and begin typing the name of a group and all groups that begin with the letters entered will be displayed.

Once at least one group has been selected, click “Run Report”. The screen will clear and the report will display as a list which can be sorted, filtered, personalized, etc. as any other list can.

# Domain Support Plugin

## Overview

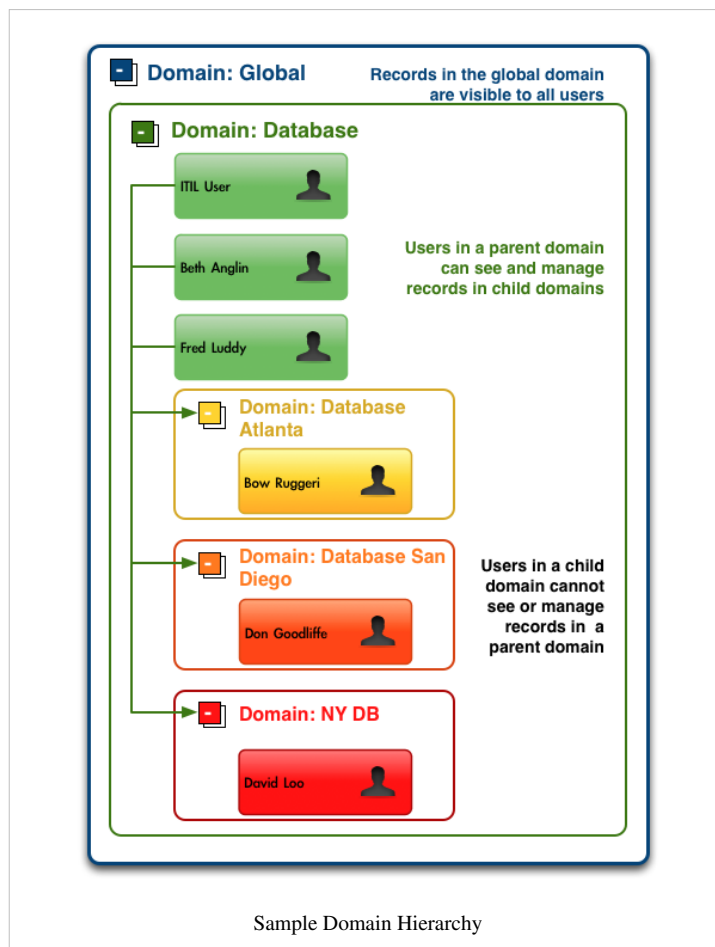
Domain separation is a way to separate data into (and optionally to separate administration by) logically-defined domains. Domain separation is best for those customers who need to:

- Enforce absolute data segregation between business entities (data separation).
- Customize business process definitions and user interfaces for each domain (delegated administration).
- Maintain some global processes and global reporting in a single instance of ServiceNow.

Domain separation is extremely well-suited for Managed Service Providers (MSPs) and global enterprises with unique business requirements in various areas of the world. Domain separation replaces Company Separation.

## Data Separation

Members of a domain only see the data contained within their domain or the child domains that are lower in the domain hierarchy. By default, all users and all records are members of the global domain unless an administrator assigns them to a particular domain. Once you assign a user or a record to a domain, the instance compares the user's domain to the record's domain to determine whether the user can view the record. For example, consider the following domain hierarchy:



In this domain hierarchy:

- Bow Ruggeri can see any records in the **Database Atlanta** or the **global** domain.
- Don Goodliffe can see any records in the **Database San Diego** or the **global** domain.
- David Loo can see any records in the **NY DB** or the **global** domain.
- Fred Luddy, ITIL User, Beth Anglin can see any records in the **Database**, **Database Atlanta**, **Database San Diego**, **NY DB**, or the **global** domain.



**Note:** Users in the **global** domain can see all records, regardless of the record's domain settings. If a user is a member of another domain, then there is no single visibility setting that allows users to see across domains or allows users to see records at a higher level in the hierarchy. See *Visibility Domains* to change what domains a user can view.

**Warning:** Guest users must be part of the global domain.

By default, domain separation adds a domain field to task and configuration item records. You can extend domain separation to any table, except the Dictionary [sys\_dictionary] and Software Model [cmdb\_software\_product\_model] tables, by adding a sys\_domain field to the table's dictionary definition. The value of the sys\_domain field equals either:

- The domain of the user who creates the record.
- The domain assigned to the record by a business rule.

If a business rule sets the domain value, the current user's domain value cannot overwrite the value set by the rule.

Visibility Domains

Domain visibility determines whether users from one domain can access records from another domain. For example, if Don Goodliffe is in the **Database** domain, and Bow Ruggeri is in the **Network** domain, and no incidents are in the global domain, then Don Goodliffe cannot access Bow Ruggeri's incidents since data separation prevents this.

Number	Caller	Short description	Category	Priority	State	Assignment group	Assigned to	Domain
INC0010001	Bow Ruggeri	Visibility domain test	Inquiry / Help	5 - Planning	New	Network		Network
INC0010002	Don Goodliffe	Domain visibility test	Inquiry / Help	5 - Planning	New	Database		Database

A sample set of domain separated incident records

You can add the Database domain as a **Visibility Domain** to the Bow Ruggeri's user record (Visibility Domains is a related list on the user record). Then, Bow Ruggeri can access Don Goodliffe's incidents since he now has visibility to the **Database** domain. If you remove the visibility domain, then Bow Ruggeri can no longer access incidents in the **Database** domain.

Welcome: Bow Ruggeri

Number	Caller	Short description	Category	Priority	State	Assignment group	Assigned to	Domain
INC0010001	Bow Ruggeri	Visibility domain test	Inquiry / Help	5 - Planning	New	Network		Network

Bow Ruggeri's incident list

Welcome: Don Goodliffe

Number	Caller	Short description	Category	Priority	State	Assignment group	Assigned to	Domain
INC0010002	Don Goodliffe	Domain visibility test	Inquiry / Help	5 - Planning	New	Database		Database

Don Goodliffe's incident list

Welcome: Bow Ruggeri

Number	Caller	Short description	Category	Priority	State	Assignment group	Assigned to	Domain
INC0010001	Bow Ruggeri	Visibility domain test	Inquiry / Help	5 - Planning	New	Network		Network
INC0010002	Don Goodliffe	Domain visibility test	Inquiry / Help	5 - Planning	New	Database		Database

Bow Ruggeri's incident list with visibility domain



**Note:** Granting users a visibility domain grants them all the rights they would normally have to the record based on ACL rule permissions.

Users can also inherit visibility domains based on their group membership if you set the domain table to the Group [sys\_user\_group] table. For example, as a member of the **Database** group, the Don Goodliffe also automatically gains the **Database** domain as a visibility domain. Group membership grants visibility to any matching domain name.

The screenshot displays three panels for user 'Don Goodliffe':

- Groups:** A list of groups including Database, Hardware, Software, Capacity Mgmt, Catalog Request Approvers for Sales, and Database San Diego. The 'Database' group is highlighted with a red box.
- Delegates:** A navigation bar with tabs for Starts, Ends, Delegate, Approvals, Assignments, CC notifications, and Meeting Invitations.
- Visibility domains:** A table showing domains granted by group membership. The 'Database' domain is listed with 'Inherited' set to 'true', 'Granted by' set to 'Database', and 'Reason' set to 'Member of group Database' (highlighted with a red box).

Visibility domains granted by group membership

## Contains Domains

Normally parent-child relationships define the domain hierarchy. A **Contains** domain allows you to relate domains *ad-hoc*, independent of parent-child relationships. For example, you cannot use parentage to make the Database domain part of both the Software domain and the Hardware domain, but you can accomplish this using a **Contains** relationship.



**Note:** *Contains* is much more powerful than visibility. Visibility controls what a user can see, but **Contains** changes the hierarchy and therefore can affect business rules and policy.

## Delegated Administration

Delegated administration allows administrators to set domain-specific policies. The policies set lower in the domain hierarchy override policies set higher in the domain hierarchy. With delegated administration, domain administrators can set domain-specific versions of these global policies and settings:

- System policies
- Application and module names
- Application roles
- Module filters

When a user has the **admin** role, then all policies in their domain or higher are visible and processed during a relevant transaction. When that administrator modifies a policy that is in a higher domain or the global domain, the system automatically creates a new record for that user's domain. It does not modify the original policy, application, or module record. (Only an administrator in the global domain can change global records.) This new record *overrides* the original.

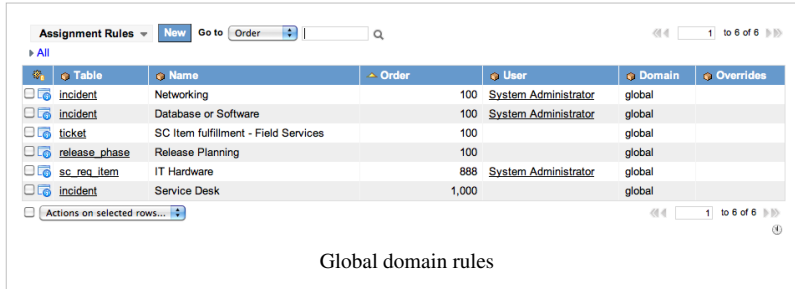
The **sys\_overrides** field indicates that a policy, application, or module at a lower level in the hierarchy overrides a record at a higher level. The system automatically sets this field when a domain administrator attempts to modify policy, application, or module at a higher level. Again, rather than actually changing the higher level record, the attempted update is changed into an insert, and the **sys\_overrides** field is set to indicate the higher level policy, application, or module that is being overridden. Later when the records for a relevant transaction are loaded, the

overriding domain-specific policy, application, or module is used ahead of the original.

## Example Delegated Administration with Domain Specific Policies

The following screens illustrate changing assignment rules at various levels of a domain hierarchy. To begin, David Loo (placed in the *Database* domain) makes a change to the global assignment policy. Then Don Goodliffe (placed in the *Database/Database San Diego* domain) also makes a change to the same policy.

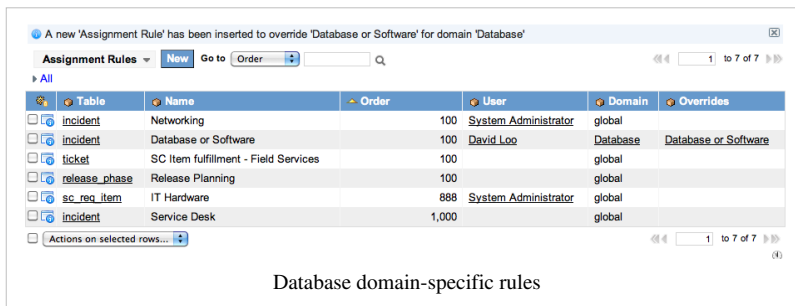
Initially, all assignment rules have a global domain as shown below:



Global domain rules

Table	Name	Order	User	Domain	Overrides
incident	Networking	100	System Administrator	global	
incident	Database or Software	100	System Administrator	global	
ticket	SC Item fulfillment - Field Services	100		global	
release_phase	Release Planning	100		global	
sc_req_item	IT Hardware	888	System Administrator	global	
incident	Service Desk	1,000		global	

If David Loo updates the assignment rule for **Database or Software**, the following list appears:



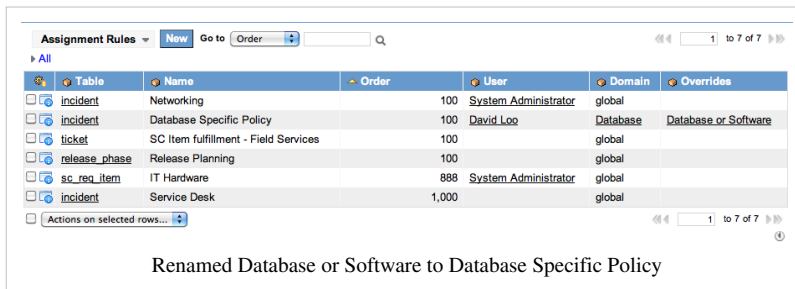
Database domain-specific rules

Table	Name	Order	User	Domain	Overrides
incident	Networking	100	System Administrator	global	
incident	Database or Software	100	David Loo	Database	Database or Software
ticket	SC Item fulfillment - Field Services	100		global	
release_phase	Release Planning	100		global	
sc_req_item	IT Hardware	888	System Administrator	global	
incident	Service Desk	1,000		global	

The following policy changes occur:

- When the policy was chosen and updated, the system detected that David Loo was not at the right level of the hierarchy to change this record. Therefore, the update was changed into an insert, and a new record was created.

Notice that there are now two policy entries with the same name. Because this is not desirable, David opens the record and changes the name to something appropriate. After the update, the list appears as follows.



Renamed Database or Software to Database Specific Policy

Table	Name	Order	User	Domain	Overrides
incident	Networking	100	System Administrator	global	
incident	Database Specific Policy	100	David Loo	Database	Database or Software
ticket	SC Item fulfillment - Field Services	100		global	
release_phase	Release Planning	100		global	
sc_req_item	IT Hardware	888	System Administrator	global	
incident	Service Desk	1,000		global	

This time, the record being updated was at the same level in the domain hierarchy as the user, so the record was simply updated with a more appropriate name. Here is the rule that David just created. Notice that database incidents will now be directly assigned to David.

Assignment Rule

Name:Database Specific Policy

Order:100

Table:Incident (Incident)

User:David Loo

Match conditions:All

Group:Software

Conditions

CategoryisDatabase

Script:

UpdateDelete

Database Specific Policy assignment rule

start out with the following assignment policy:

Assignment Rules

Go to

Order

1 to 7 of 7

Table	Name	Order	User	Domain	Overrides
incident	Networking	100	System Administrator	global	
incident	Database Specific Policy	100	David Loo	Database	Database or Software
ticket	SC Item fulfillment - Field Services	100		global	
release_phase	Release Planning	100		global	
sc_req_item	IT Hardware	888	System Administrator	global	
incident	Service Desk	1,000		global	

1 to 7 of 7

Don Goodliffe's starting view of assignment rules

A new 'Assignment Rule' has been inserted to override 'Database San Diego Specific Policy' for domain 'Database/Database San Diego'

Assignment Rules

Go to

Order

1 to 8 of 8

Table	Name	Order	User	Domain	Overrides
incident	Networking	100	System Administrator	global	
incident	Database San Diego Specific Policy	100	Don Goodliffe	Database/Database San Diego	Database Specific Policy
ticket	SC Item fulfillment - Field Services	100		global	
release_phase	Release Planning	100		global	
sc_req_item	IT Hardware	888	System Administrator	global	
incident	Service Desk	1,000		global	

1 to 8 of 8

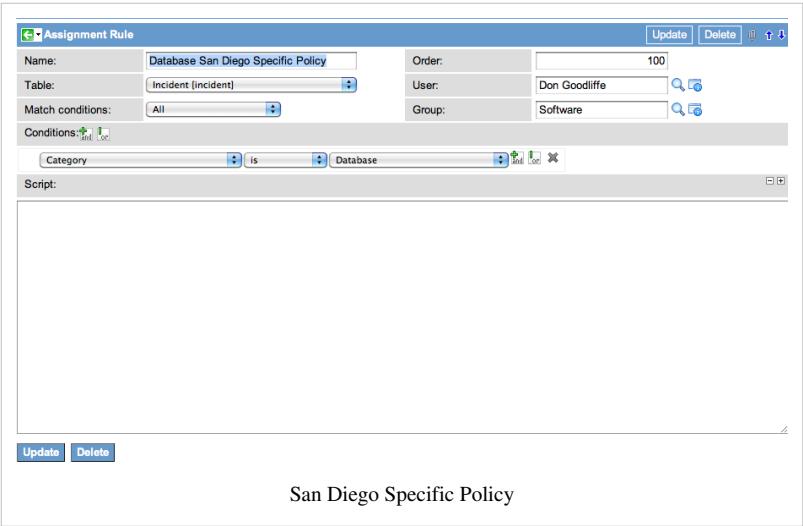
Database San Diego rules override Database Specific Policy rules

If a new incident is created in the Database domain (or lower in the hierarchy), the new rule is applied. It has overridden the global assignment rule. If a new incident is created in the global domain (or any other domain not within the Database domain hierarchy), then the global rule applies. In the following scenario, Don Goodliffe, in the Database/Database San Diego domain hierarchy, decides that database incidents created in his domain should be assigned to him rather than to David Loo. As an administrator, Don Goodliffe would

Notice that this level of the hierarchy starts out with the policy established at the parent level (the Database domain). After changing the **Database Specific Policy**, this list would now look like this:

Again, the attempted update was changed automatically to an insert, and the override value was supplied to indicate that the higher-level policy is being overridden. Here is the rule that was just created; it shows that database incidents created in the *Database San Diego* domain will be assigned to Don Goodliffe.





The result of the above customization is:

- A database incident from the *Database San Diego* domain will be assigned to Don Goodliffe
- A database incident from the Database hierarchy other than *Database San Diego* will be assigned to David Loo
- A database incident from any other domain, including *global*, will be assigned to the System Administrator

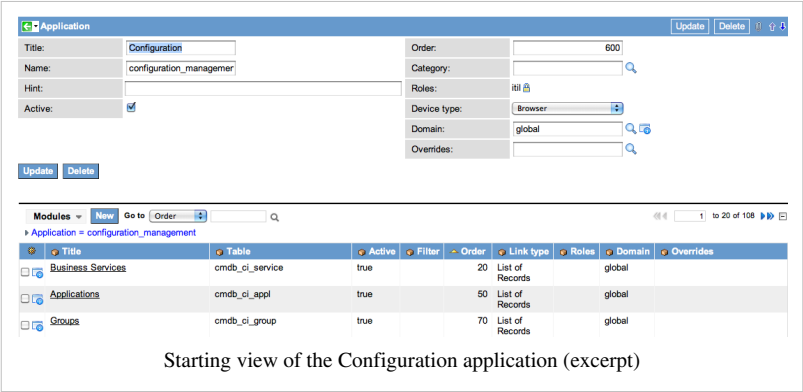
The above customizations all show changes to higher-level policy. However, new policy can also be created any any level of the domain hierarchy.

During a transaction, the current user's domain normally determines the policy to load. For example when a user in the Database domain updates an incident, the Database domain is used for business rules and policies even if the incident record was originally created in the *Database San Diego* domain. By default, the user's domain supersedes the record's domain.

There is a system setting that can change this behavior. If **Using the Current Record's Domain Instead of the Current User's Domain** is set to **true**, then the above behavior is reversed. The domain of the record is used to determine which policy to load, not the domain of the user. For example if a user in the *Database* domain updates an incident that is in the *Database San Diego* domain, then the business rules and policy that exist for *Database San Diego* are executed. The domain of the user still determines the records that are visible to the user, and the domain of the user sets the domain for records that user creates, but is not a factor in determining rules and policies.

Example Delegated Administration with Domain Specific Applications and Modules

As the administrator of the Database domain, David Loo decides to customize the Configuration application. To start with, David reviews the modules available in the Configuration application module.



David decides to rename the Configuration application to CMDB and to allow the inventory\_admin role to see the application.

Applications

New

Go to Title

1 to 20 of 46

All > Active = true > Device type != Mobile

Title	Active	Order	Roles	Name	Domain	Overrides
Asset Management	true	900	asset	asset	global	
BSM Map	true		admin	bsm_map	global	
Change	true	400	itil	change_management	global	
CMDB	true	600	itil inventory_admin	configuration_management	Database configuration_management	
Content Management	true		content_admin	cms	global	
Contract Management	true	1,000	asset contract_manager	asset_contracts	global	
Domain Admin	true		domain_admin	domain_admin	global	
ECC	true		admin	ecc	global	
Homepage Admin	true		admin	home	global	
Incident	true	200	itil	incident_management	global	
Instance Clone	true		clone_admin	instance_clone	global	
Knowledge Base	true	800	knowledge	km	global	
Metrics	true		itil_admin metric_admin	metrics	global	
MID Server	true		admin	MID	global	
Organization Management	true	875	asset	organization_management	global	
Problem	true	300	itil	problem_management	global	
Reports	true	1,100	itil	reports	global	
SAML 2 Single Sign-on	true		admin	saml_2_single_sign_on	global	
SAML Single Sign-on	true		admin	SAML Single Sign-on	global	

Activate

Deactivate

Actions on selected rows...

1 to 20 of 46

Sample domain-specific changes to the Configuration application

Next, David decides to change the Incident application by activating the **Open - in "New" State** module and adding a new filter item to show open incidents in the Database category.

Module

Required field

Update

Delete

Title:

Open - in "New" state

Link type:

List of Records

Table:

Incident [incident]

View name:

Order:

200

Roles:

Application:

incident\_management

Hint:

Active:

☒

Image:

Filter:

Incident state is New

and Active is true

and Category is Database

Arguments:

incident\_state=1\*active=true

Update

Delete

Sample domain-specific changes to the Open - "New" State module

This creates a new module entry in the application rather than overwriting the existing module in the global domain.

Title: Incident

Name: incident\_management

Hint:

Active: ☒

Order: 200

Category:

Roles: itil

Device type: Browser

Domain: global

Overrides:

Update

Delete

Modules

New

Go to

Order

1 to 12 of 12

	Title	Table	Active	Filter	Order	Link type	Roles	Domain	Overrides
<input type="checkbox"/>	Create New	incident	true		100	URL (from Arguments.)		global	
<input type="checkbox"/>	Assigned to me	incident	true	active=true*assigned_to=javascript:getMy...	150	List of Records		global	
<input type="checkbox"/>	Open	incident	true	active=true*EQ	200	List of Records		global	
<input type="checkbox"/>	Open - in "New" state	incident	true	incident_state=1*active=true*category=da...	200	List of Records		Database incident	
<input type="checkbox"/>	Open - Unassigned	incident	true	assigned_to=NULL*active=true*EQ	300	List of Records		global	
<input type="checkbox"/>	Resolved	incident	true	state=0*EQ	325	List of Records		global	
<input type="checkbox"/>	Closed	incident	true	active=false*EQ	350	List of Records		global	
<input type="checkbox"/>	All	incident	true		400	URL (from Arguments.)		global	
<input type="checkbox"/>	Overview		true		500	URL (from Arguments.)		global	
<input type="checkbox"/>	Critical Incidents Map		true		600	URL (from Arguments.)		global	
<input type="checkbox"/>	Trend Chart	sys_dashboard_template	false		700			global	

Actions on selected rows...

Domain-specific view of the Incident application

If another administrator from another domain, such as Fred Luddy, logs in and looks at the Configuration application, he see the settings from the global domain.

Welcome: David Loo

Type filter text

Self-Service

Service Desk

Incident

Problem

Change

CMDB

Service Catalog

Knowledge Base

Organization Management

Asset Management

Contract Management

David Loo's view of applications

Domain Query Methods

A domain query method allows the instance to efficiently query large numbers of domains. There are two domain query methods.

- Domain paths
- (Legacy) Domain numbers

Part of Domain Support 2.0 is a new query engine designed to perform and scale to tens of thousands of domains. Prior methods, including domain numbering, have had limitations that domain paths resolves. While you have the flexibility to continue using your existing query method, we highly recommend that you switch to domain paths through the new Domain Configuration screen at your earliest convenience.

Domain Paths

A domain path is a series of three-character codes separated by a slash (/) delimiter that uniquely identifies a domain. Each digit in the three-character code consists of one of

the following 60 possible characters:

Welcome: Fred Luddy

Type filter text

Self-Service

Service Desk

Incident

Problem

Change

Configuration

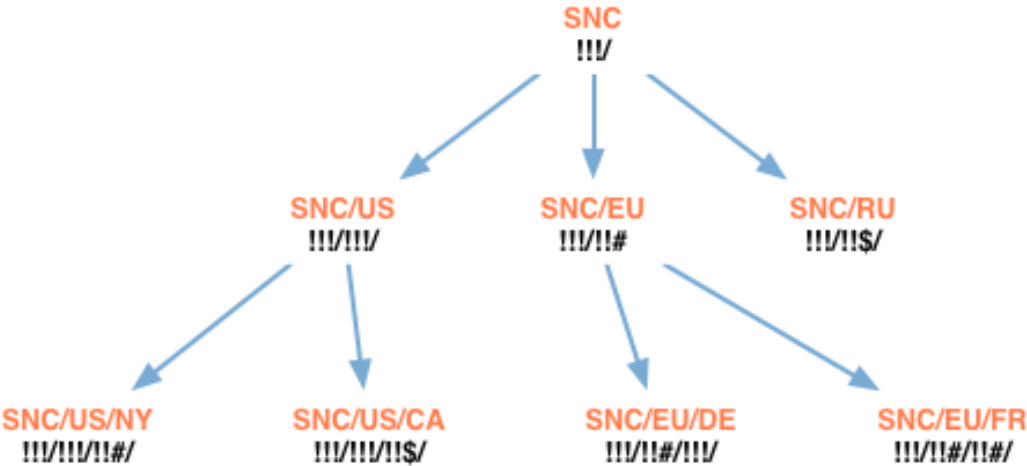
Service Catalog

Knowledge Base

Fred Luddy's view of applications

!#\$%&()\*+,-.0123456789:;<?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^`{|{~

The three-character codes that make up a path are not unique across a domain tree. Rather, the entire path string itself is unique. For example:



Domain Name	Parent Domain	Domain Path
SNC	None	!!!/
SNC/US	SNC	!!!/!!!/
SNC/EU	SNC	!!!/!!#/
SNC/RU	SNC	!!!/!\$/
SNC/US/NY	SNC/US	!!!/!!!/!!#/
SNC/US/CA	SNC/US	!!!/!!!/!\$/
SNC/EU/DE	SNC/EU	!!!/!!#/!!!/
SNC/EU/FR	SNC/EU	!!!/!!#/!!#/



**Note:** With three-character codes delimited by a single character in a path string of 255 total characters, each node of the domain tree supports up to 216,000 child domains, and the maximum depth of the tree is 63 levels.

## Legacy: Domain Numbering

Domain numbering is a legacy query method that assigns simple decimal reference numbers to each domain. These numbers are easier to query than strings of long domain names. Customers whose networks include thousands of domains, such as managed service providers (MSP), used the domain numbering query method to improve the efficiency of database queries.

Domain numbering has been superseded by domain paths, which is even more efficient, consistent and scalable. ServiceNow recommends disabling domain numbering after you successfully test and validate the domain paths query method.

## Enhancements

### Dublin

- Two new properties are available to handle on-screen notifications that appear when the domain picker automatically changes based on which domain the user is currently in:
  - `glide.domain.notify_change`: When enabled, a notification appears telling the user that the domain picker automatically changed. The default value is **true** after administrators add this property to the System Properties [sys\_properties] table.
  - `glide.domain.notify_record_change`: When enabled, a notification appears telling the user that the domain picker automatically changed because the record that the user is viewing changed the domain in which the user is in. The default value is **false** after administrators add this property to the System Properties [sys\_properties] table.

# Role Delegation Plugin

## Overview

With the Role Delegation plugin, an administrator can grant a user the right to delegate roles within a particular group with the `role_delegator` role. Roles available to a role delegator can come from roles specifically granted to that user or roles that the user inherits by being the member of a group. For example, if a user is a role delegator in the Network and Database groups, he may delegate roles he inherits from the Network group to members of the Database group.

## Defining a Role Delegator

When approved, the `role_delegator` role is granted to a specified user in the named group. That user may then delegate any role they have to any member of the group.

To designate a role delegator:

1. Navigate to **User Administration > Designate Role Delegator**.
2. Select the group in which a member shall be a role delegator.
3. Select the member of the group who will be the role delegator.



**Catalog Item - Grant role delegation rights within a group**

A role delegator may delegate any role they have to any member of the specified group

**Group**

[More information](#)

Network

**User**

[More information](#)

Fred Luddy

**Submit**

4. Click **Submit**.

A change request is created for the role delegator request. The change request is approved automatically.

Change Request: CHG0030002 opened for your request to grant role delegation rights to user: Fred Luddy in group: Network

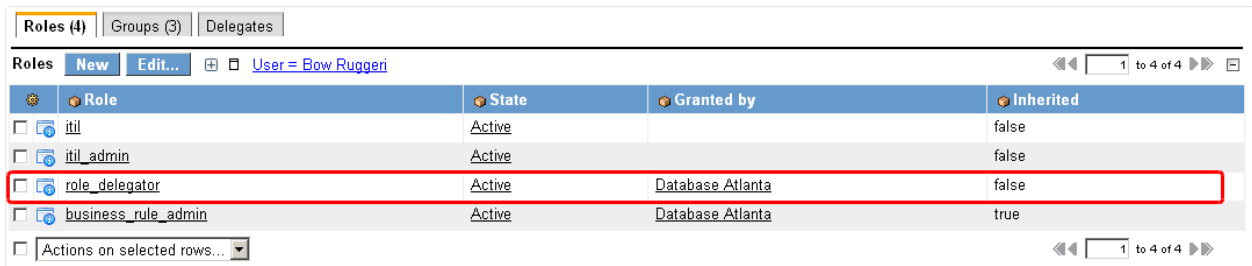
## Viewing Delegated Roles

An administrator can view role designation in the following locations on the platform:

- User records
- Role Delegators module
- Role Audit module

### User Records

Open a user's record (**User Administration > Users**) to view all the roles assigned to that user.

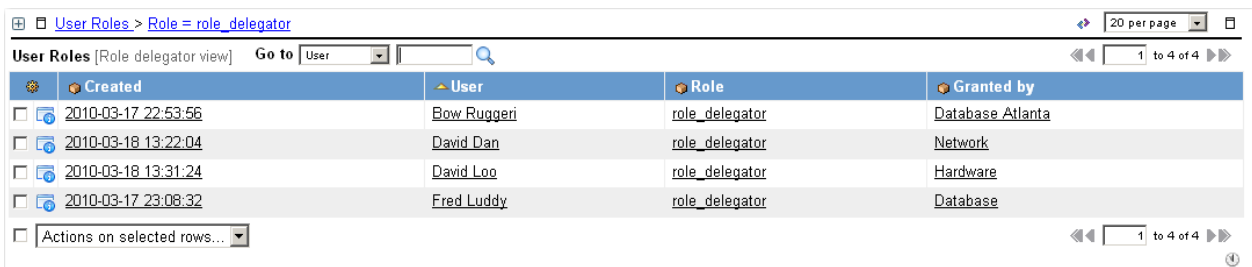


The screenshot shows the 'User Roles' view for the user 'Bow Ruggeri'. The table lists roles assigned to the user, with the 'role\_delegator' role highlighted by a red box.

Role	State	Granted by	Inherited
itil	Active		false
itil_admin	Active		false
role_delegator	Active	Database Atlanta	false
business_rule_admin	Active	Database Atlanta	true

### Role Delegators

To view existing role delegators (and the groups in which they can delegate roles), navigate to **User Administration > Role Delegators**. All the role delegators in the instance are listed, showing the groups in which they have the role\_delegator role.

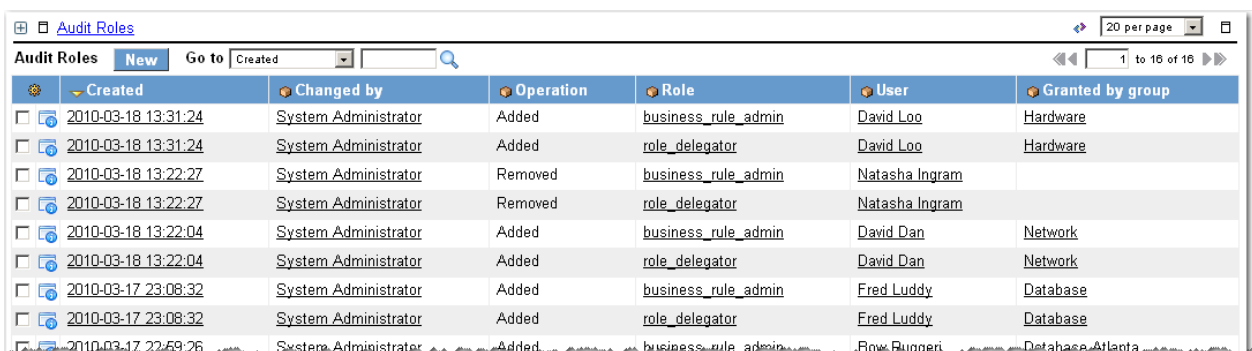


The screenshot shows the 'Role Delegators' view. The table lists role delegators, including the user, role, and granted by group.

Created	User	Role	Granted by
2010-03-17 22:53:56	Bow Ruggeri	role_delegator	Database Atlanta
2010-03-18 13:22:04	David Dan	role_delegator	Network
2010-03-18 13:31:24	David Loo	role_delegator	Hardware
2010-03-17 23:08:32	Fred Luddy	role_delegator	Database

### Role Audit

The Audit Role list view displays all the role changes made in the instance by user and group. To access the Audit Role list, navigate to **System Security > Reports > Audit Roles**.



The screenshot shows the 'Audit Roles' view. The table lists role changes, including the created date, changed by, operation, role, user, and granted by group.

Created	Changed by	Operation	Role	User	Granted by group
2010-03-18 13:31:24	System Administrator	Added	business_rule_admin	David Loo	Hardware
2010-03-18 13:31:24	System Administrator	Added	role_delegator	David Loo	Hardware
2010-03-18 13:22:27	System Administrator	Removed	business_rule_admin	Natasha Ingram	
2010-03-18 13:22:27	System Administrator	Removed	role_delegator	Natasha Ingram	
2010-03-18 13:22:04	System Administrator	Added	business_rule_admin	David Dan	Network
2010-03-18 13:22:04	System Administrator	Added	role_delegator	David Dan	Network
2010-03-17 23:08:32	System Administrator	Added	business_rule_admin	Fred Luddy	Database
2010-03-17 23:08:32	System Administrator	Added	role_delegator	Fred Luddy	Database
2010-03-17 22:59:26	System Administrator	Added	business_rule_admin	Bow Ruggeri	Database Atlanta

## Delegating Roles

To delegate specific roles to members of a group, navigate to **User Administration > Delegate Roles in Group**. This module is available to users with the role\_delegator role. The role delegator provides the following fields:

Field	Input Value
Group	Select the group in which a member shall be delegated a role or roles
User	Select the member who shall be delegated roles in that group.
Roles to delegate	Select the roles to delegate to the group member

**Catalog Item - Delegate roles to group member**

A role delegator may delegate any role they have to any member of the selected group. This allows for fine-tuned role management within a group.

**Group**  
[More information](#)

**User**  
[More information](#)

**Roles to delegate**  
[More information](#)

Collection

Search

- agent\_admin
- approval\_admin
- asset
- assignment\_rule\_admin
- catalog
- catalog\_admin
- form\_admin
- gauge\_maker
- image\_admin
- import\_admin
- itil
- itil\_admin
- knowledge
- knowledge\_admin
- list\_updater
- metric\_admin
- personalize
- personalize\_choices

Add

Remove

List

- filter\_admin
- filter\_global
- filter\_group

Name

**Submit**

Upon submission, a change request is created for the delegation request. This change request is approved automatically, and the specified roles are granted to the named user in the group selected.

Change Request: CHG0030003 opened for your request to delegate roles to user: Fred Luddy in group: Network

Adding role filter\_global to Fred Luddy

Adding role filter\_group to Fred Luddy



## Removing Roles

Delegated roles can be removed in the same form by reversing the process. Select the group and user, remove the unwanted roles from the Roles slushbucket, and then re-submit the request.

## Administration

### Record Producers

The Role Delegation modules link to Record Producers. These Record Producers create Change Requests that are, by default, automatically approved by the following graphical workflows:

- *Grant role\_delegator role to user in group*
- *Delegate roles to group member* graphical workflows.



**Note:** *These workflows can be customized as desired to add approval steps.*

### can\_delegate Field

The Roles [sys\_user\_role] table has a **can\_delegate** field. A role can be delegated if this field's value is **true**. In the base system, the following fields are not delegatable:

- **admin**
- **role\_delegator** (a user with the role\_delegator role cannot, by default, delegate *this* role to other group members)
- **public**
- **nobody**

### Group Manager Change Business Rule

The **Group Manager Change** business rule, disabled by default, will automatically grant the role\_delegator role to a user when they become manager of a group (using the **Manager** field on the Group form). The role is removed when the user is no longer the manager of the group.

To take advantage of this business rule, simply activate it.

# User Self-Registration Plugin

## Overview

The **User Self-Registration Plugin** provides the ability for unregistered users to request access to a ServiceNow instance.

## Requesting an Account

If a user would like to request an account, they navigate directly to the instance. If the plugin is installed, the following section is added to the welcome screen:

### Request a user account

If you do not yet have a user account you can request one using the [self registration form](#).

Once the user has clicked on the link, they will be presented with a form to fill in with their first and last names, and email address:

Once they submit the form, they will see a confirmation that their request has been submitted:

Your request has been submitted and is pending review. You will receive an email when your request is processed.

If the email matches an email in the system, their request will not submit:

A user account for 'guy.yedwab@service-now.com' already exists in the system

Invalid update

## Approving Accounts

Administrators can approve accounts by navigating to **User Administration > Pending User Registration**. Pending registration request will appear in the list:

User Registration Requests		New	Go to	First name			1 to 1 of 1
All > State = Pending							
	First name	Last name	Email	State	User	Created	
<input type="checkbox"/>	Guy	Yedwab	guy.yedwab@service-now.com	Pending		2011-04-28 09:16:31	

On the registration request's form, the UI actions **Create User** and **Reject** can be used to approve or deny the request.

**User Registration Request** = Required field Update Delete

First name:  State:

Last name:  User:

Email:  Created:

Update Delete

**Related Links**  
[Create User](#)  
[Reject](#)

If the UI action **Create User** is selected, a new user will be created using the email address as the User ID:

Primary email device created for Guy Yedwab  
 User account created, user will be notified.

The user will be informed by an email notification.

If the UI action **Reject** is selected, the request will be marked **Rejected** and the user will be notified:

User request has been rejected, user will be notified.

To view past registration requests, remove the **State = Pending** breadcrumb from the list view:

**User Registration Requests** New Go to   1 to 3 of 3

► All

	First name	Last name	Email	State	User	Created
<input type="checkbox"/>	<a href="#">Guy</a>	Yedwab	guy.yedwab@service-now.com	Processed	<a href="#">Guy Yedwab</a>	2011-04-28 09:16:31
<input type="checkbox"/>	<a href="#">Steve</a>	Wood	steve.wood@service-now.com	Rejected		2011-04-28 09:18:49
<input type="checkbox"/>	<a href="#">Wally</a>	Marx	wally.marx@service-now.com	Pending		2011-04-28 09:29:52

## Auto-Processing

To enable auto-processing of requests, navigate to **System Properties > System** and set the property **Enable auto processing of user registration requests...** to true. If true, registration requests will not require approval. Instead, the business rule **Auto-Process User Registration** will create the user record from the information provided.

## Installed with the Plugin

### Applications and Modules

The module **Pending User Registrations** is added to the **User Administration** application.

### Database Table Structure

The following tables will be added:

Display Name (Table Name)	Description
User Registration Request [user_registration_request]	The table of all requests made by users for access to the instance.

## Scripts

The following business rules will be added to **sys\_script**:

- **Validate registration**
- **Auto Process User Registration**

The following UI Actions will be added to **sys\_ui\_action**:

- **Create User**
- **Reject**

The following email notifications will be added to **sysevent\_email\_action**:

- **User Registration Reject**
- **User Registration Processed**

## Getting Started

### Requesting the Plugin

Click the plus to expand instructions for activating a plugin.

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

If the plugin depends on other plugins, these plugins and their activation status are listed.

3. [Optional] Select the **Load demo data** check box.

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

4. Click **Activate**.
-

# Article Sources and Contributors

**User Administration** *Source:* <http://wiki.servicenow.com/index.php?oldid=164526> *Contributors:* Cheryl.dolan, G.yedwab, Guy.yedwab, Joe.Westrich, Joseph.messerschmidt, Vhearne

**Managing User Sessions** *Source:* <http://wiki.servicenow.com/index.php?oldid=189837> *Contributors:* CapaJC, Christen.mitchell, David Loo, Emily.partridge, G.yedwab, Guy.yedwab, Joseph.messerschmidt, Neola, Phillip.salzman, Rachel.sienko, Suzannes, Vhearne

**Creating Roles** *Source:* <http://wiki.servicenow.com/index.php?oldid=193861> *Contributors:* Cheryl.dolan, Emily.partridge, Guy.yedwab, Joseph.messerschmidt, Neola, Steve.wood, Suzannes

**Counting Licensed Users** *Source:* <http://wiki.servicenow.com/index.php?oldid=201588> *Contributors:* Gadi.yedwab, Guy.yedwab, Joseph.messerschmidt, Neola, Steve.wood, Suzannes

**Adding a New Department** *Source:* <http://wiki.servicenow.com/index.php?oldid=100045> *Contributors:* CapaJC, G.yedwab, Guy.yedwab, Joseph.messerschmidt, Neola, Steve.wood, Vhearne

**Impersonating a User** *Source:* <http://wiki.servicenow.com/index.php?oldid=174776> *Contributors:* CapaJC, Eric.jacobson, Guy.yedwab, Joseph.messerschmidt, Mark.stanger, Neola, Pat.Casey, Phillip.salzman, Rachel.sienko, Steve.wood, Vhearne

**Skills Management** *Source:* <http://wiki.servicenow.com/index.php?oldid=207483> *Contributors:* Cheryl.dolan, Emily.partridge, G.yedwab, Guy.yedwab, John.roberts, Joseph.messerschmidt, Neola, Rachel.sienko, Steve.wood, Wallymarx

**Defining Locations** *Source:* <http://wiki.servicenow.com/index.php?oldid=208298> *Contributors:* Cheryl.dolan, Guy.yedwab, Joseph.messerschmidt, Mark.stanger, Phillip.salzman, Rachel.sienko, Steve.wood

**Creating Groups** *Source:* <http://wiki.servicenow.com/index.php?oldid=200632> *Contributors:* CapaJC, Cheryl.dolan, Emily.partridge, G.yedwab, Guy.yedwab, Jeremiah.hall, Joseph.messerschmidt, Neola, Peter.smith, Suzannes, Vaughn.romero, Vhearne

**Associating Users to Groups** *Source:* <http://wiki.servicenow.com/index.php?oldid=195755> *Contributors:* CapaJC, Cheryl.dolan, Emily.partridge, Guy.yedwab, Joe.Westrich, Joseph.messerschmidt, Neola, Phillip.salzman, Steve.wood, Suzannes, Vaughn.romero, Vhearne

**Granting Access** *Source:* <http://wiki.servicenow.com/index.php?oldid=60514> *Contributors:* Cheryl.dolan, Guy.yedwab, Jessi.graves, Joseph.messerschmidt, Neola, Rachel.sienko, Steve.wood, Vaughn.romero, Wallymarx

**Using Access Control Rules** *Source:* <http://wiki.servicenow.com/index.php?oldid=162076> *Contributors:* CapaJC, G.yedwab, Grant.hulbert, Guy.yedwab, Joseph.messerschmidt, Neola, Phillip.salzman, Rachel.sienko, Steve.wood, Suzannes, Vaughn.romero

**Security Jump Start (ACL Rules) Plugin** *Source:* <http://wiki.servicenow.com/index.php?oldid=89889> *Contributors:* Aleck.lin, CapaJC, Guy.yedwab, Joseph.messerschmidt, Neola, Rachel.sienko, Steve.wood, Vhearne

**Group On-Call Rotation Plugin** *Source:* <http://wiki.servicenow.com/index.php?oldid=89840> *Contributors:* Cheryl.dolan, G.yedwab, Guy.yedwab, Joe.Westrich, Joseph.messerschmidt, Lee.carver, Prasad.Rao, Steve.wood

**Domain Support Plugin** *Source:* <http://wiki.servicenow.com/index.php?oldid=122820> *Contributors:* CapaJC, Cheryl.dolan, Don.Goodliffe, G.yedwab, Gadi.yedwab, Guy.yedwab, Jared.laethem, Joseph.messerschmidt, Michael.hoefer, Neola, Nick.roberts, Phillip.salzman, Rachel.sienko, Richard.motteram, Rob.phillips, Steve.wood, Vaughn.romero, Vhearne, Wallymarx

**Role Delegation Plugin** *Source:* <http://wiki.servicenow.com/index.php?oldid=89883> *Contributors:* CapaJC, Emily.partridge, Joseph.messerschmidt, Neola, Pat.Casey, Steve.wood

**User Self-Registration Plugin** *Source:* <http://wiki.servicenow.com/index.php?oldid=82784> *Contributors:* Emily.partridge, Guy.yedwab, Joseph.messerschmidt, Rachel.sienko

# Image Sources, Licenses and Contributors

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

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

**Image:Lock out User.jpg** Source: [http://wiki.servicenow.com/index.php?title=File:Lock\\_out\\_User.jpg](http://wiki.servicenow.com/index.php?title=File:Lock_out_User.jpg) License: unknown Contributors: Christen.mitchell

**Image:User active.png** Source: [http://wiki.servicenow.com/index.php?title=File:User\\_active.png](http://wiki.servicenow.com/index.php?title=File:User_active.png) License: unknown Contributors: Pat.Casey

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

**Image:DepartmentSetup1.png** Source: <http://wiki.servicenow.com/index.php?title=File:DepartmentSetup1.png> License: unknown Contributors: CapaJC

**Image:Impersonate button.png** Source: [http://wiki.servicenow.com/index.php?title=File:Impersonate\\_button.png](http://wiki.servicenow.com/index.php?title=File:Impersonate_button.png) License: unknown Contributors: Phillip.salzman

**Image:Impersonate first.JPG** Source: [http://wiki.servicenow.com/index.php?title=File:Impersonate\\_first.JPG](http://wiki.servicenow.com/index.php?title=File:Impersonate_first.JPG) License: unknown Contributors: Eric.jacobson, Guy.yedwab

**Image:ImpersonateLog1.png** Source: <http://wiki.servicenow.com/index.php?title=File:ImpersonateLog1.png> License: unknown Contributors: CapaJC

**Image:skills\_record\_calgary.png** Source: [http://wiki.servicenow.com/index.php?title=File:Skills\\_record\\_calgary.png](http://wiki.servicenow.com/index.php?title=File:Skills_record_calgary.png) License: unknown Contributors: Suzannes

**Image:skills\_user\_record\_Calgary.png** Source: [http://wiki.servicenow.com/index.php?title=File:Skills\\_user\\_record\\_Calgary.png](http://wiki.servicenow.com/index.php?title=File:Skills_user_record_Calgary.png) License: unknown Contributors: Suzannes

**Image:skills\_group\_record\_Calgary.png** Source: [http://wiki.servicenow.com/index.php?title=File:Skills\\_group\\_record\\_Calgary.png](http://wiki.servicenow.com/index.php?title=File:Skills_group_record_Calgary.png) License: unknown Contributors: Suzannes

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

**Image:acl\_workflow.png** Source: [http://wiki.servicenow.com/index.php?title=File:Acl\\_workflow.png](http://wiki.servicenow.com/index.php?title=File:Acl_workflow.png) License: unknown Contributors: Vaughn.romero

**Image:acl\_evaluate\_permissions.png** Source: [http://wiki.servicenow.com/index.php?title=File:Acl\\_evaluate\\_permissions.png](http://wiki.servicenow.com/index.php?title=File:Acl_evaluate_permissions.png) License: unknown Contributors: Vaughn.romero

**Image:acl\_matching.png** Source: [http://wiki.servicenow.com/index.php?title=File:Acl\\_matching.png](http://wiki.servicenow.com/index.php?title=File:Acl_matching.png) License: unknown Contributors: Vaughn.romero

**Image:Acl debug.png** Source: [http://wiki.servicenow.com/index.php?title=File:Acl\\_debug.png](http://wiki.servicenow.com/index.php?title=File:Acl_debug.png) License: unknown Contributors: Pat.Casey, Vaughn.romero

**Image:Acl debug list.png** Source: [http://wiki.servicenow.com/index.php?title=File:Acl\\_debug\\_list.png](http://wiki.servicenow.com/index.php?title=File:Acl_debug_list.png) License: unknown Contributors: Pat.Casey, Vaughn.romero

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

**Image:On call menu.jpg** Source: [http://wiki.servicenow.com/index.php?title=File:On\\_call\\_menu.jpg](http://wiki.servicenow.com/index.php?title=File:On_call_menu.jpg) License: unknown Contributors: CapaJC, Jared.laethem

**File:OnCallCalendar.png** Source: <http://wiki.servicenow.com/index.php?title=File:OnCallCalendar.png> License: unknown Contributors: Jhopwood

**Image:domain\_hierarchy.png** Source: [http://wiki.servicenow.com/index.php?title=File:Domain\\_hierarchy.png](http://wiki.servicenow.com/index.php?title=File:Domain_hierarchy.png) License: unknown Contributors: Vaughn.romero

**Image:Caution-diamond.png** Source: <http://wiki.servicenow.com/index.php?title=File:Caution-diamond.png> License: unknown Contributors: John.roberts

**Image:add\_domain.png** Source: [http://wiki.servicenow.com/index.php?title=File:Add\\_domain.png](http://wiki.servicenow.com/index.php?title=File:Add_domain.png) License: unknown Contributors: Vaughn.romero

**Image:domain\_visibility\_02.png** Source: [http://wiki.servicenow.com/index.php?title=File:Domain\\_visibility\\_02.png](http://wiki.servicenow.com/index.php?title=File:Domain_visibility_02.png) License: unknown Contributors: Vaughn.romero

**Image:domain\_visibility\_03.png** Source: [http://wiki.servicenow.com/index.php?title=File:Domain\\_visibility\\_03.png](http://wiki.servicenow.com/index.php?title=File:Domain_visibility_03.png) License: unknown Contributors: Vaughn.romero

**Image:domain\_visibility\_04.png** Source: [http://wiki.servicenow.com/index.php?title=File:Domain\\_visibility\\_04.png](http://wiki.servicenow.com/index.php?title=File:Domain_visibility_04.png) License: unknown Contributors: Vaughn.romero

**Image:visibility\_domains\_by\_group.png** Source: [http://wiki.servicenow.com/index.php?title=File:Visibility\\_domains\\_by\\_group.png](http://wiki.servicenow.com/index.php?title=File:Visibility_domains_by_group.png) License: unknown Contributors: Vaughn.romero

**Image:delegated\_administration\_01.png** Source: [http://wiki.servicenow.com/index.php?title=File:Delegated\\_administration\\_01.png](http://wiki.servicenow.com/index.php?title=File:Delegated_administration_01.png) License: unknown Contributors: Phillip.salzman, Vaughn.romero

**Image:delegated\_administration\_02.png** Source: [http://wiki.servicenow.com/index.php?title=File:Delegated\\_administration\\_02.png](http://wiki.servicenow.com/index.php?title=File:Delegated_administration_02.png) License: unknown Contributors: Phillip.salzman, Vaughn.romero

**Image:delegated\_administration\_03.png** Source: [http://wiki.servicenow.com/index.php?title=File:Delegated\\_administration\\_03.png](http://wiki.servicenow.com/index.php?title=File:Delegated_administration_03.png) License: unknown Contributors: Phillip.salzman, Vaughn.romero

**Image:delegated\_administration\_04.png** Source: [http://wiki.servicenow.com/index.php?title=File:Delegated\\_administration\\_04.png](http://wiki.servicenow.com/index.php?title=File:Delegated_administration_04.png) License: unknown Contributors: Phillip.salzman, Vaughn.romero

**Image:delegated\_administration\_05.png** Source: [http://wiki.servicenow.com/index.php?title=File:Delegated\\_administration\\_05.png](http://wiki.servicenow.com/index.php?title=File:Delegated_administration_05.png) License: unknown Contributors: Phillip.salzman, Vaughn.romero

**Image:delegated\_administration\_06.png** Source: [http://wiki.servicenow.com/index.php?title=File:Delegated\\_administration\\_06.png](http://wiki.servicenow.com/index.php?title=File:Delegated_administration_06.png) License: unknown Contributors: Phillip.salzman, Vaughn.romero

**Image:delegated\_administration\_07.png** Source: [http://wiki.servicenow.com/index.php?title=File:Delegated\\_administration\\_07.png](http://wiki.servicenow.com/index.php?title=File:Delegated_administration_07.png) License: unknown Contributors: Phillip.salzman, Vaughn.romero

**Image:domain\_overrides\_appmod\_01.png** Source: [http://wiki.servicenow.com/index.php?title=File:Domain\\_overrides\\_appmod\\_01.png](http://wiki.servicenow.com/index.php?title=File:Domain_overrides_appmod_01.png) License: unknown Contributors: Vaughn.romero

**Image:domain\_overrides\_app\_02.png** Source: [http://wiki.servicenow.com/index.php?title=File:Domain\\_overrides\\_app\\_02.png](http://wiki.servicenow.com/index.php?title=File:Domain_overrides_app_02.png) License: unknown Contributors: Phillip.salzman, Vaughn.romero

**Image:domain\_overrides\_mod\_03.png** Source: [http://wiki.servicenow.com/index.php?title=File:Domain\\_overrides\\_mod\\_03.png](http://wiki.servicenow.com/index.php?title=File:Domain_overrides_mod_03.png) License: unknown Contributors: Vaughn.romero

**Image:domain\_overrides\_mod\_04.png** Source: [http://wiki.servicenow.com/index.php?title=File:Domain\\_overrides\\_mod\\_04.png](http://wiki.servicenow.com/index.php?title=File:Domain_overrides_mod_04.png) License: unknown Contributors: Phillip.salzman, Vaughn.romero

**Image:domain\_overrides\_appmod\_04.png** Source: [http://wiki.servicenow.com/index.php?title=File:Domain\\_overrides\\_appmod\\_04.png](http://wiki.servicenow.com/index.php?title=File:Domain_overrides_appmod_04.png) License: unknown Contributors: Vaughn.romero

**Image:domain\_overrides\_appmod\_05.png** Source: [http://wiki.servicenow.com/index.php?title=File:Domain\\_overrides\\_appmod\\_05.png](http://wiki.servicenow.com/index.php?title=File:Domain_overrides_appmod_05.png) License: unknown Contributors: Vaughn.romero

**Image:Domain paths.png** Source: [http://wiki.servicenow.com/index.php?title=File:Domain\\_paths.png](http://wiki.servicenow.com/index.php?title=File:Domain_paths.png) License: unknown Contributors: Vaughn.romero

**Image:Role Delegator1.png** Source: [http://wiki.servicenow.com/index.php?title=File:Role\\_Delegator1.png](http://wiki.servicenow.com/index.php?title=File:Role_Delegator1.png) License: unknown Contributors: Steve.wood

**Image:Role Delegator Change.png** Source: [http://wiki.servicenow.com/index.php?title=File:Role\\_Delegator\\_Change.png](http://wiki.servicenow.com/index.php?title=File:Role_Delegator_Change.png) License: unknown Contributors: Steve.wood

**Image:Role User Record.png** Source: [http://wiki.servicenow.com/index.php?title=File:Role\\_User\\_Record.png](http://wiki.servicenow.com/index.php?title=File:Role_User_Record.png) License: unknown Contributors: Steve.wood

**Image:Role Delegator List.png** Source: [http://wiki.servicenow.com/index.php?title=File:Role\\_Delegator\\_List.png](http://wiki.servicenow.com/index.php?title=File:Role_Delegator_List.png) License: unknown Contributors: Steve.wood

**Image:Role Audit.png** Source: [http://wiki.servicenow.com/index.php?title=File:Role\\_Audit.png](http://wiki.servicenow.com/index.php?title=File:Role_Audit.png) License: unknown Contributors: Steve.wood

**Image:Role Delegation.png** Source: [http://wiki.servicenow.com/index.php?title=File:Role\\_Delegation.png](http://wiki.servicenow.com/index.php?title=File:Role_Delegation.png) License: unknown Contributors: Steve.wood

**Image:Role Change Request.png** Source: [http://wiki.servicenow.com/index.php?title=File:Role\\_Change\\_Request.png](http://wiki.servicenow.com/index.php?title=File:Role_Change_Request.png) License: unknown Contributors: Steve.wood

**Image:USR-request1.png** Source: <http://wiki.servicenow.com/index.php?title=File:USR-request1.png> License: unknown Contributors: Guy.yedwab

**Image:USR-request2.png** Source: <http://wiki.servicenow.com/index.php?title=File:USR-request2.png> License: unknown Contributors: Guy.yedwab

**Image:USR-Request3.png** Source: <http://wiki.servicenow.com/index.php?title=File:USR-Request3.png> License: unknown Contributors: Guy.yedwab

**Image:USR-Request4.png** Source: <http://wiki.servicenow.com/index.php?title=File:USR-Request4.png> License: unknown Contributors: Guy.yedwab

**Image:USR-approve1.png** Source: <http://wiki.servicenow.com/index.php?title=File:USR-approve1.png> License: unknown Contributors: Guy.yedwab

**Image:USR-approve2.png** Source: <http://wiki.servicenow.com/index.php?title=File:USR-approve2.png> License: unknown Contributors: Guy.yedwab

**Image:USR-approve3.png** Source: <http://wiki.servicenow.com/index.php?title=File:USR-approve3.png> License: unknown Contributors: Guy.yedwab

**Image:USR-approve4.png** Source: <http://wiki.servicenow.com/index.php?title=File:USR-approve4.png> License: unknown Contributors: Guy.yedwab

**Image:USR-approve5.png** Source: <http://wiki.servicenow.com/index.php?title=File:USR-approve5.png> License: unknown Contributors: Guy.yedwab