Using the User Administration Application

PDF generated using the open source mwlib toolkit. See http://code.pediapress.com/ for more information.
PDF generated at: Wed, 29 Nov 2017 20:32:57 PST

Introduction

User Administration

Note: *This article applies to Fuji and earlier releases. For more current information, see User Administration* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

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.

On-Call Scheduling

Ensure optimum availability of on-call personnel.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/roles/concept/c_UserAdministration.html

Functions

Managing User Sessions

Note: *This article applies to Fuji. For more current information, see Manage User Sessions* [1] *at* http://docs.servicenow.com The ServiceNow Wiki is no longer being updated. Please refer to http://docs.servicenow.com for the latest product documentation.

Overview

The ServiceNow platform provides the ability to view and terminate individual user sessions, lock out users from the system, and make users inactive.

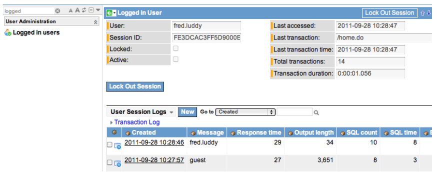
- Terminating a user session effectively logs that user out of the next transaction, which is usually the next browser click. Use the terminate sessions feature when you want to perform system maintenance.
- Locking a user out of the system means the user can no longer log in or generate any actions from any email messages that the user sends to the instance. Locking out users also terminates their user sessions.
- Making a user inactive means that the user does not show up in any fields that reference active users on the User table.

Terminating a Specific User Session

1. Navigate to **User Administration > Logged in users**.

   You can only see users who are logged into the same application node as you. If the **Active** field on a user record value is **false**, the user is logged in but not currently running a transaction. Most users appear as inactive at any given time.
2. Select the session you want to end.
3. Click **Lock Out Session**.

   The session is terminated, and the user is redirected to the login page at the next attempted transaction. Multiple user sessions may be associated with one user, so terminating a user session only affects the specific session. A user is not yet "locked out" when you terminate the user sessions. The user can still log back in at any time.



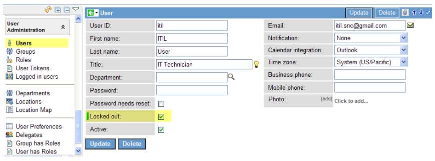
User: fred.luddy
Session ID: FE3DCAC3FF5D9000E
Locked: □
Active: □
Last accessed: 2011-09-28 10:28:47
Last transaction: /home.do
Last transaction time: 2011-09-28 10:28:47
Total transactions: 14
Transaction duration: 0:00:01.056

Created	Message	Response time	Output length	SQL count	SQL time
2011-09-28 10:28:46	fred.luddy	29	34	10	8
2011-09-28 10:27:57	guest	27	3,651	8	3

Terminate a user session

Locking out a User

1. Navigate to **User Administration > Users**.
2. Select the user from the list.
3. Select the **Locked Out** check box, and update the record.



User ID: itil
First name: ITIL
Last name: User
Title: IT Technician
Department:
Password:
Password needs reset: □
Locked out: ☑
Active: ☑
Email: itil.snc@gmail.com
Notification: None
Calendar integration: Outlook
Time zone: System (US/Pacific)
Business phone:
Mobile phone:
Photo: [add] Click to add...

Lock out a user

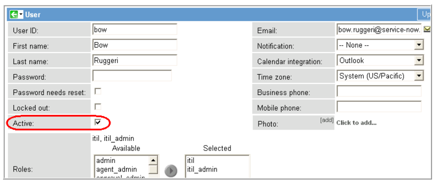
Note: *The system prevents users with the admin role from locking themselves out. This feature is available starting with the Fuji release.*

Marking a User Inactive

Making a user inactive does not lock out the user. The **Lock Out Inactive Users** business rule, which is active by default in all instances, sets the **Locked Out** flag to **true** on the User record when the Active flag is set to **false**. If you do not have this business rule active, inactive users are not automatically locked out and can still log in the instance.

To make a user inactive:

1. Navigate to **User Administration > Users**, and select the user from the list.
2. Clear the **Active** checkbox.
3. Click **Update**.



User ID: bow
First name: Bow
Last name: Ruggeri
Password:
Password needs reset: □
Locked out: □
Active: ☑
Roles: itil, itil_admin
Available: admin, agent_admin
Selected: itil, itil_admin
Email: bow.ruggeri@service-now.
Notification: -- None --
Calendar integration: Outlook
Time zone: System (US/Pacific)
Business phone:
Mobile phone:
Photo: [add] Click to add...

Mark user as inactive

Enhancements

Fuji

- Prevents users with the admin role from locking themselves out.

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.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/user-sessions/concept/c_ManageUserSessions.html

Creating Roles

Note: *This article applies to Fuji. For more current information, see Roles* [1] *at* http://docs.servicenow.com The ServiceNow Wiki is no longer being updated. Please refer to http://docs.servicenow.com for the latest product documentation.

Overview

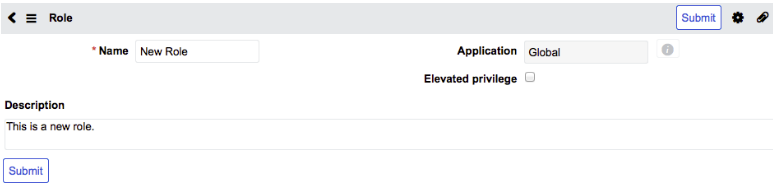
Roles control access to features and capabilities in applications and modules. Once access has been granted to a role, all of the groups or users assigned to the role are granted the access. Roles can contain other roles, and any access granted to a role is granted to any role that contains it.

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. Fill out the form fields (see table).
4. Click **Submit**.

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



* Name: New Role

Application: Global

Elevated privilege □

Description

This is a new role.

Creating a role

Field	Input Value
Name	Enter a name for the role.
Elevated privilege	Select this option to mark this role as required to elevate to high security. Roles that require users to elevate to high security grant modification access to the High Security Settings and allows the user to modify the Access Control List, directly import XML files, and access the *Scripts - Background* module.
Description	Select the roles to delegate to the group member.

Adding Roles to an Existing Role

When you add a new role to an existing role for a user, the user inherits the access that is granted by the new role.

1. Open the existing role and click **Edit** in the **Contains Roles** related list.
2. Use the slushbucket to add one or more roles to the existing role.
3. Click **Save**.

   The users with the existing role inherit the access that is granted by the new role.

Granting a Role Access to Applications and Modules

You add a role to an application or module to enable the role to grant access to the application or module for all users with the role.

1. Navigate to **System Definition > Applications** or **System Definition > Modules**.
2. Click the appropriate application or module to open it in the 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, and then save your changes.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/roles/concept/c_Roles.html

Counting Licensed Users

Overview

Note: *The **Licensed Users** module is no longer available with the Eureka release. The following description is in place so that you can save any data that you need.*

As of October 1, 2013, for the Eureka release only, most ServiceNow products are licensed per user. 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.*

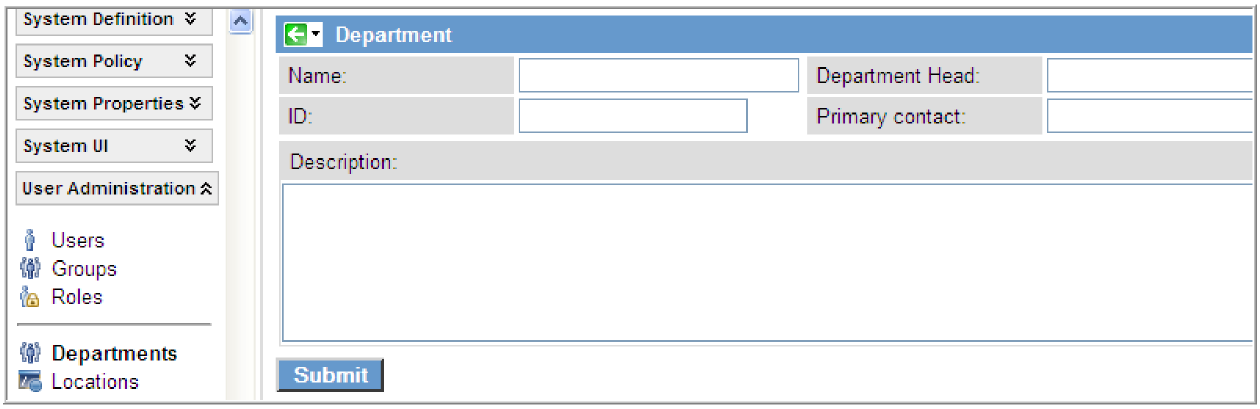
Adding a New Department

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**



Impersonating a User

Note: *This article applies to Fuji and earlier releases. For more current information, see Impersonate a User* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

Overview

Administrators can impersonate other users [2] 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:

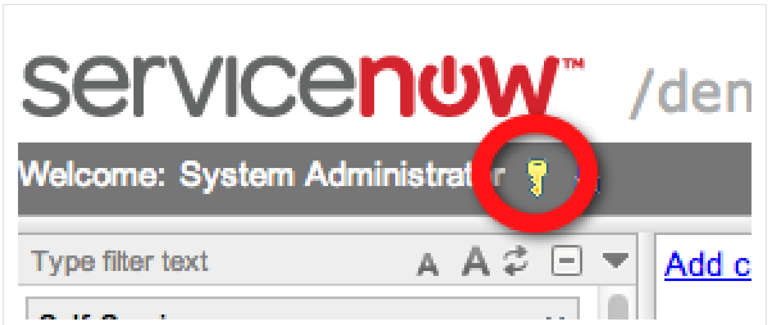
- An **admin** account to do work.
- An **itil** (or similar) login to test as a technician.
- An **ess** login to test as an end user.

More logins may be required to adequately test the system.

Note: *When you impersonate a user who is locked out or is inactive, the system forces you out of the system as well after you generate an event or click a link.*

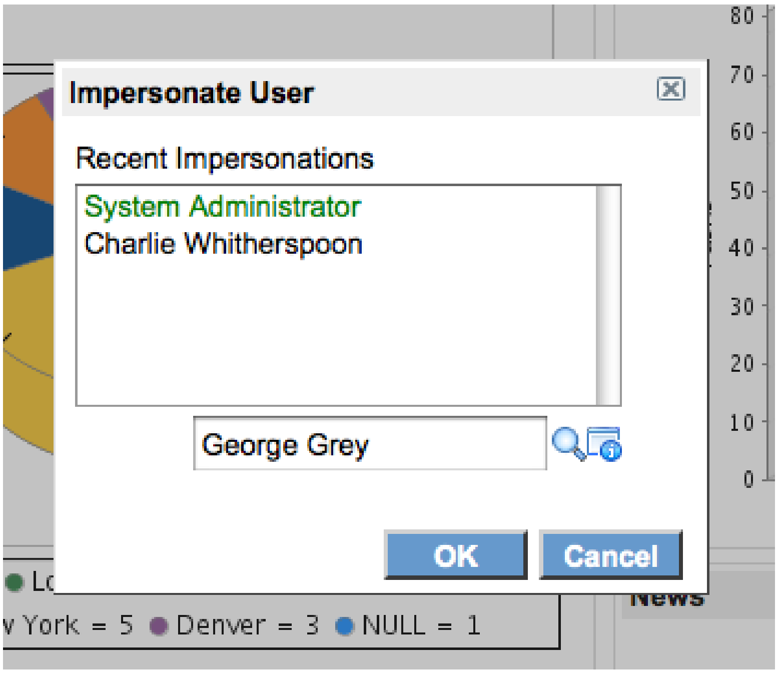
Impersonating A User

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.



Select the user to impersonate

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

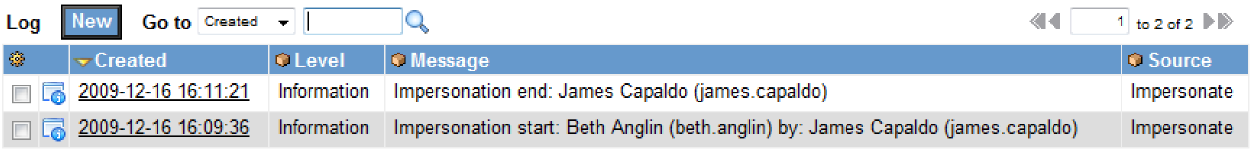
The Impersonate button and its effects are contained in a UI Macro called impersonate_button. Modifying the impersonate_button is not recommended.

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.



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.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/users-and-groups/concept/c_ImpersonateAUser.html

[2] https://docs.servicenow.com/bundle/helsinki-servicenow-platform/page/administer/users-and-groups/concept/c_ImpersonateAUser.html

Skills Management

❗ **Note:** *This article applies to Fuji and earlier releases. For more current information, see Skills Management* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**'

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.

❗ **Note:** *References to Work Management indicate that the information on this page is valid for ServiceNow versions prior to Fuji. In the Fuji release, Work Management was renamed Field Service 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:

- Work Management
- Project Management v2 Plugin

Administrators can also activate the Skills plugin manually.

Click the plus to expand instructions for activating a plugin.

If you have the admin role, use the following steps to activate the plugin.

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

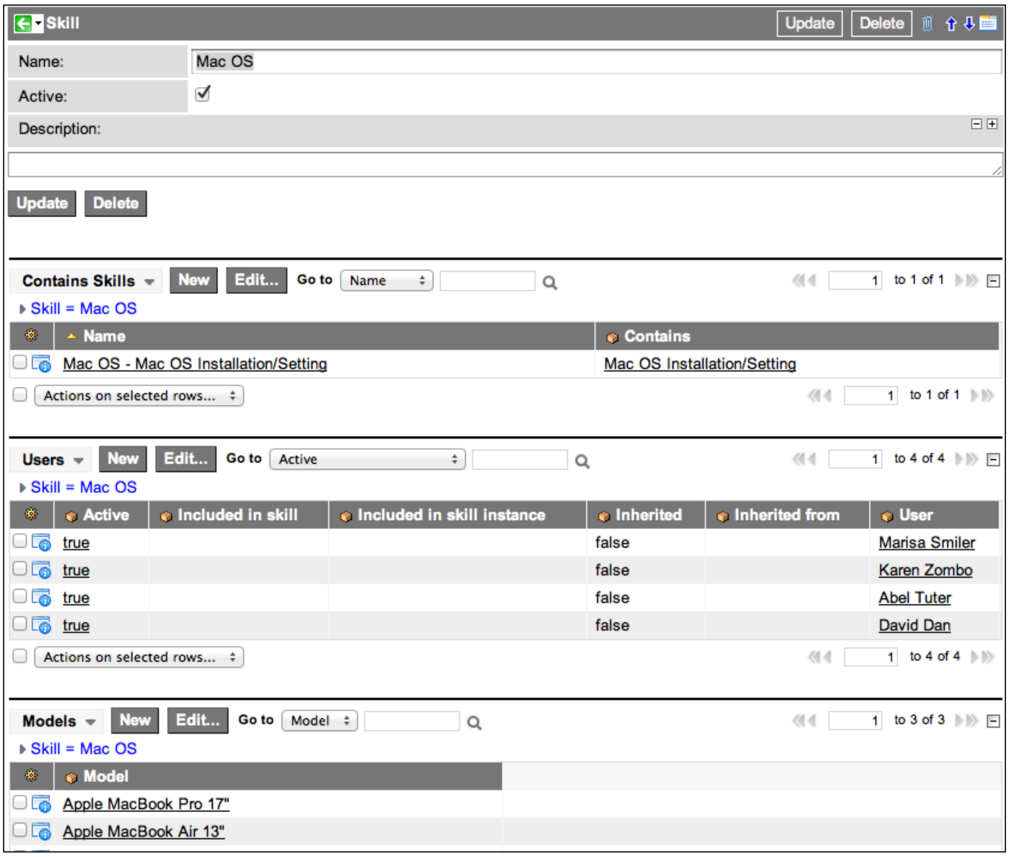
   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. [Optional] If available, select the **Load demo data** check box.

   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 you first activate 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) wgi have the skill and sub-skills named in this record.



Skill

Name:	Mac OS
Active:	☑
Description:	

Contains Skills

Skill = Mac OS

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

1 to 1 of 1

Users

Skill = Mac OS

Active	Included in skill	Included in skill instance	Inherited	Inherited from	User
true			false		Marisa Smiler
true			false		Karen Zombo
true			false		Abel Tuter
true			false		David Dan

1 to 4 of 4

Models

Skill = Mac OS

Model
Apple MacBook Pro 17"
Apple MacBook Air 13"

1 to 3 of 3

Assigning Skills

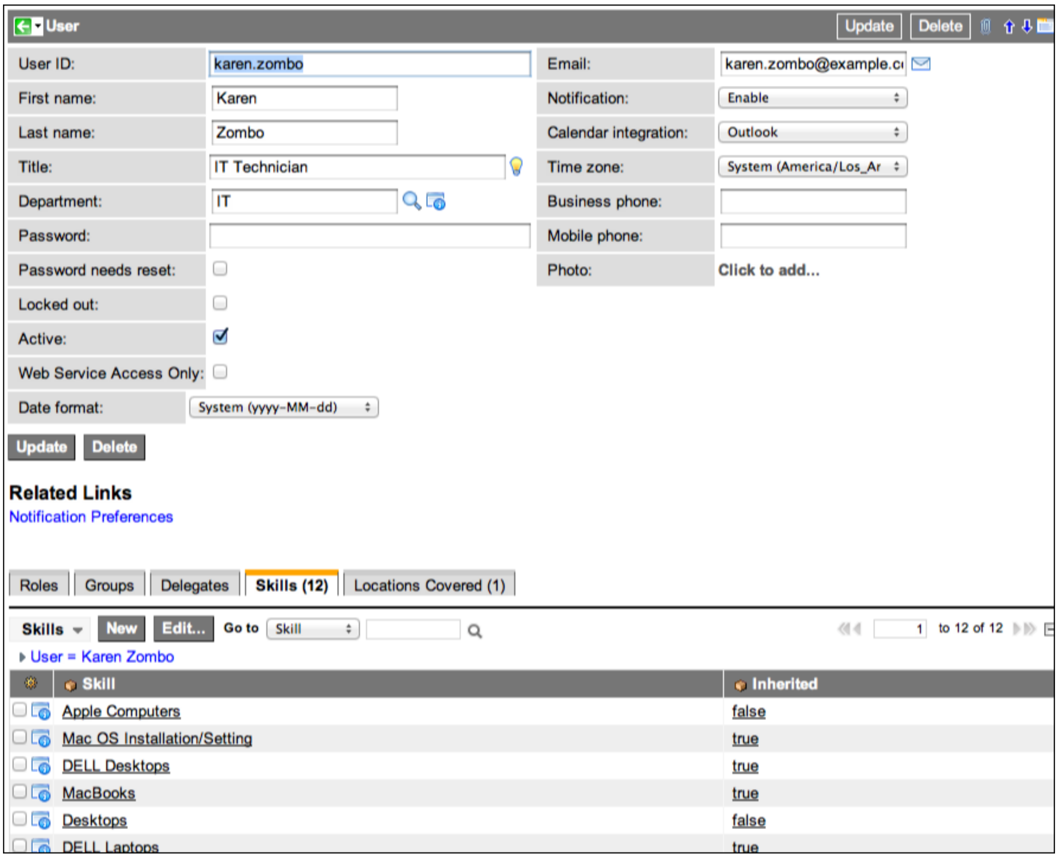
Assign skills to individual users or to groups. Members of a group inherit all the skills configured for their group.

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 ID:	karen.zombo	Email:	karen.zombo@example.c
First name:	Karen	Notification:	Enable
Last name:	Zombo	Calendar integration:	Outlook
Title:	IT Technician	Time zone:	System (America/Los_Ar
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)		

Related Links
Notification Preferences

User = Karen Zombo

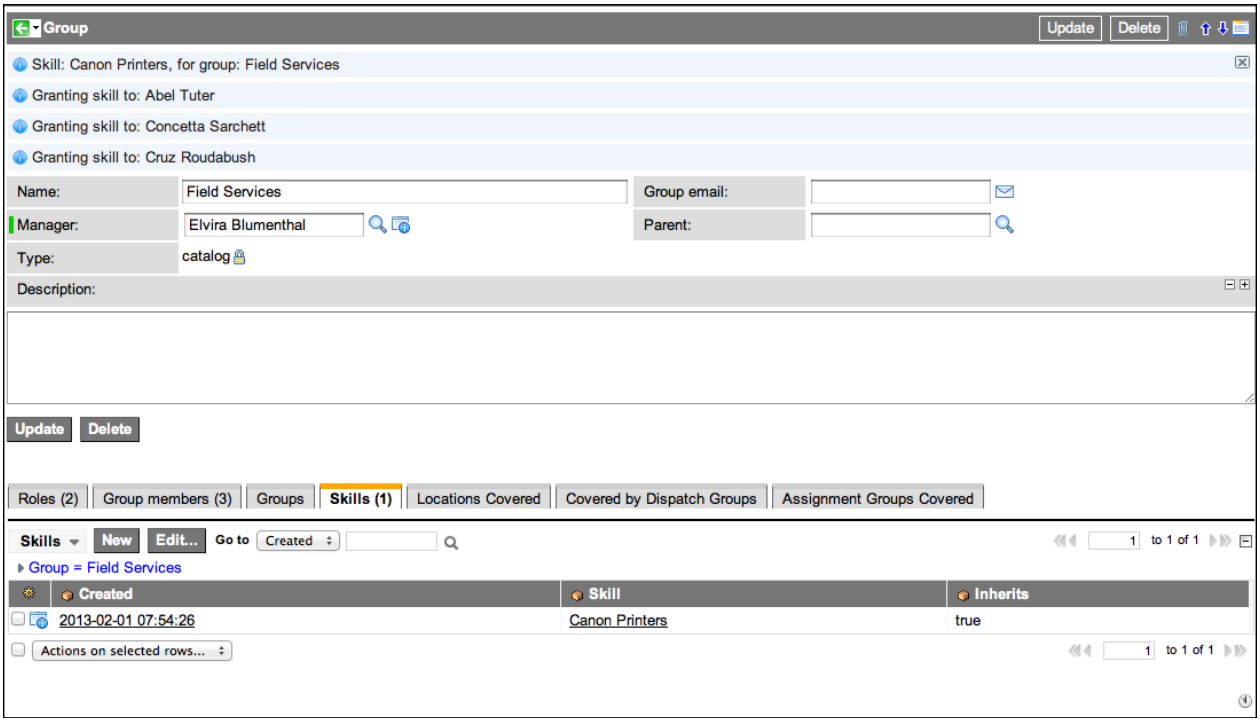
	Skill	Inherited
☐	Apple Computers	false
☐	Mac OS Installation/Setting	true
☐	DELL Desktops	true
☐	MacBooks	true
☐	Desktops	false
☐	DELL Laptops	true

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.



Skill: Canon Printers, for group: Field Services

Granting skill to: Abel Tuter

Granting skill to: Concetta Sarchett

Granting skill to: Cruz Roudabush

Name:	Field Services	Group email:	
Manager:	Elvira Blumenthal	Parent:	
Type:	catalog		
Description:			

Group = Field Services

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 if filtered on group members with the defined skills.

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

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/users-and-groups/concept/c_SkillsManagement.html

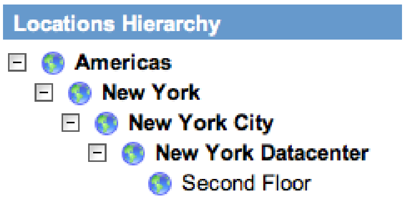
Defining Locations

Note: *This article applies to Fuji and earlier releases. For more current information, see Location Setup* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

Overview

Locations are used by various applications to locate users, facilities, or configuration items (CI). You can configure different levels of location in a parent-child hierarchy. For example, an email server might be associated with a location of **Second Floor**, whereas the email business service might be associated with **New York City**.

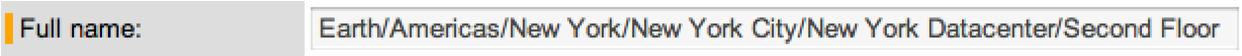
Each level of this hierarchy contains a separate Location record, with the next higher level specified as a **Parent**. In this example each location is selectable as a hierarchy from reference fields:



Locations Hierarchy

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

The location is also used to generate a full identifier in the **Full name** field (available by configuring the form):



Full name: Earth/Americas/New York/New York City/New York Datacenter/Second Floor

Locations are stored on the **Location [cmn_location]** table.

Defining a Location

To create a location, navigate to **User Administration > Locations** and click **New** (see table).

Field	Input Value
Name	The name of the location. This is the display value that the the system uses when referencing this location on a form.
Street	The street address of the location.
City	The city of the location.
State / Province	State or province of the location.
Zip / Postal Code	The zip or postal code of the location.
Country	The country of the location.
Contact	Name of a user who is the contact for this location.
Phone	The phone number for the location.
Fax phone	The fax number for the location.
Parent	Name of the parent location for this location. See the Overview for information about location hierarchies.
Latitude	The latitude of the location. This field is populated automatically by the **get_lat_long** business rule when the form contains enough information, such as an address or city name and a postal code. Deactivate this business rule to prevent the system from overwriting any values populated in the field manually. Latitude is expressed as a floating point data type starting with the Eureka release.

Longitude	The longitude of the location. This field is populated automatically by the **get_lat_long** business rule when the form contains enough information, such as an address or city name and a postal code. Deactivate this business rule to prevent the system from overwriting any values populated in the field manually. Longitude is expressed as a floating point data type starting with the Eureka release.
Fields that can be added by configuring the form:	
Company	A reference field to the **Company [cmn_company]** 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	The location's time zone. By default, the location uses the system time zone. For more information, see Using Time Zones.

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.

Enhancements

Eureka

- Latitude and longitude are expressed as a floating point data type. Previous releases expressed this information as a string. During an upgrade, the system converts the data where possible.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/localization/concept/c_LocationSetup.html

Creating Groups

Note: *This article applies to Fuji. For more current information, see Groups* [1] *at* http://docs.servicenow.com The ServiceNow Wiki is no longer being updated. Please refer to http://docs.servicenow.com for the latest product documentation.

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.

Note: *References to Work Management indicate that the information on this page is valid for ServiceNow versions prior to Fuji. In the Fuji release, Work Management was renamed Field Service Management.*

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 configure the form.

   Note: *The default_assignee field has no functionality associated with it.*

Field	Description
Name	Name of the group.
Manager	Group manager or lead.
Type	Category for this group. For example, a group designated as type **catalog** is a service catalog group and can also be accessed under the **Service Catalog > Catalog Policy > Fulfillment Groups** module. You may need to configure the form to add the **Type** field. Activating the Work Management plugin (Calgary release) adds the **Type** field automatically. See also Configuring Group Types for Assignment Groups.
Group email	Group email distribution list or the email address of the group's point of contact, such as the group manager.
Parent	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 not members of the parent group. For example, if an incident is assigned to the parent group and you click the **Assigned to** lookup icon, only the members in the parent group are available. The members of the child group are not available.
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: • lists of groups • the reference lookup list for reference fields • the autocomplete list of groups displayed when you type into a reference field

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 **Group Email** address. The only exception to this functionality is for approval notifications, whereby all members of a group receive an approval notification, regardless of the **Include members** selection.
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**.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/users-and-groups/concept/c_Groups.html

Associating Users to Groups

Note: *This article applies to Fuji. For more current information, see Create a User* [1] *at* http://docs.servicenow.com The ServiceNow Wiki is no longer being updated. Please refer to http://docs.servicenow.com for the latest product documentation.

Overview

When you add users to the ServiceNow system, 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] Configure 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	Description
User ID	Create a unique identifier for this user's ServiceNow login user name. Typical examples of user IDs are **cwitherspoon** and **charlie.witherspoon**. You cannot create a new user whose User ID duplicates an existing user. If you do import duplicates from an update set, the more recently created name 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 of the user's active sessions. The system prevents users with the admin role from locking themselves out (starting with the Fuji release).
Active	Select this check box to make this user active. Only the administrator sees inactive users in: • Lists of users • The selection list on reference fields (magnifying glass icon) • The auto-complete list that appears when you type into a reference field
Web service access only	Select this check box to designate this user as a non-interactive user. This field is available with Non-Interactive Sessions, starting with the Calgary release.
Internal Integration User	Select this check box to designate this user as an internal integration user. This field is available starting with the Dublin release.
Date format	Select the user's preferred format for dates.

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. 1. Navigate to **System Definition > Validation Scripts**. 2. Select the **email** record. 3. Clear the **Active** check box and save the change. 4. Complete the user profile, including the email address, and update or submit the record. 5. Reactivate the email validation script.
Notification	Select the type of notification to send to this user. The default is **Email**. If you select **None**, 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 ensure that user’s notifications remain active, the user must have at least one primary email along with their SMS device. This is because the business rule **Update User Record** accounts only for email devices, and if there are none, it disables the user's notification preferences regardless of having or not having an SMS device. 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 **None**.
Calendar integration	Select **Outlook** to have this user receive meeting notifications via email directly to the calendar. Otherwise, select **None**.
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.
Geolocation tracked	Select the check box to enable location tracking. This field is available when geolocation is active, starting with the Eureka release.
Location	Select the user's usual location. This field is visible when geolocation is active, starting with the Eureka release.

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.

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 the system property.

1. Navigate to the System Properties table by entering **sys_properties.list** in the navigation filter.
2. Search for the `glide.ui.welcome.profile_link` property.
3. Set the value to **true**.

Enhancements

Fuji

- Prevents users with the admin role from locking themselves out.

Eureka

- The **Geolocation tracked** field, which is available when Geolocation is activated, provides the option to track a user's location.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/users-and-groups/task/t_CreateAUser.html

User Security

Granting Access

Note: *This article applies to Fuji. For more current information, see Contextual Security Manager* [1] *at* http://docs.servicenow.com The ServiceNow Wiki is no longer being updated. Please refer to http://docs.servicenow.com for the latest product documentation.

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 configure 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.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/roles/reference/r_ContextualSecurity_1.html

Using Access Control Rules

Note: *This article applies to Fuji and earlier releases. For more current information, see Access Control List Rules* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

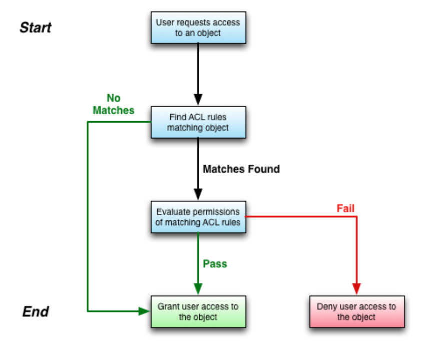
Overview

An instance 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** and **operation** being secured
- The **permissions** required to access the object

The system searches for ACL rules that **match** both the object and operation the user wants to access. If there are no matching ACL rules for the object and operation combination, then the object does not require any additional security checks and the instance grants the user access to them. By default, the system provides ACL rules to restrict access to all database and configuration operations.

After finding a matching ACL rule, the system *evaluates* if the user has the permissions required to access the object and operation. If a user meets the ACL rule permissions, the instance grants the user access to the listed operation on the object. If a user does not meet the ACL rule permissions of the first matching rule, the system evaluates the next matching ACL rule. If the user fails to meet the ACL rule permissions of any matching ACL rule, the system denies the user access to the operation on the object.


Start
User requests access to an object
No Matches
Find ACL rules matching object
Matches Found
Evaluate permissions of matching ACL rules
Fail
Pass
End
Grant user access to the object
Deny user access to the object

ACL Rule Workflow

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

Creating ACL Rules

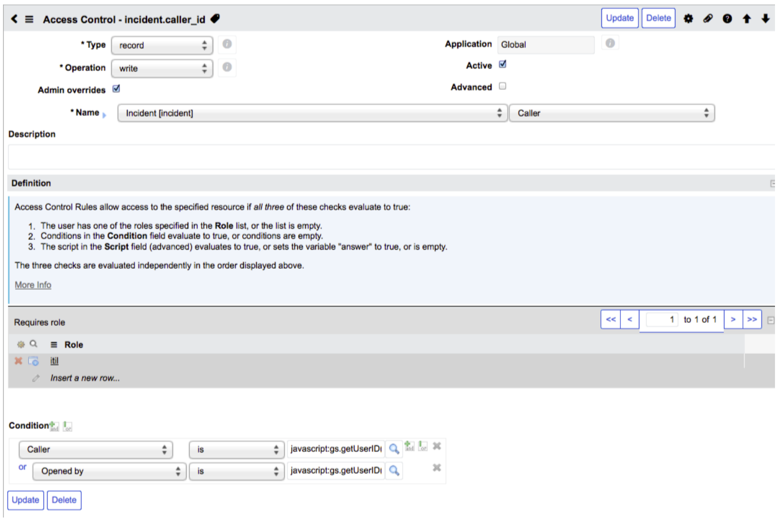
Create custom ACL rules to secure access to new objects or to change the default security behavior. To create new ACL rules, you must elevate privileges to the security_admin role. You can create ACL rules only for objects that are in the same scope as the ACL rule and for other tables that have at least one field in the same scope as the ACL rule (starting with the Fuji release). For tables that are in a different scope than the ACL rule record, the types of rules are limited.

To create an ACL rule:

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. Right-click the form header and select **Save**.

Access Control Fields

Access control records use the following fields.



Access Control - incident.caller_id

* Type: record

* Operation: write

Admin overrides ☑

Application: Global

Active ☑

Advanced □

* Name: Incident [incident] | Caller

Description

Definition

Access Control Rules allow access to the specified resource if *all three* of these checks evaluate to true:

1. The user has one of the roles specified in the **Role** list, or the list is empty.
2. Conditions in the **Condition** field evaluate to true, or conditions are empty.
3. The script in the **Script** field (advanced) evaluates to true, or sets the variable "answer" to true, or is empty.

The three checks are evaluated independently in the order displayed above.

More Info

Requires role

<< < 1 to 1 of 1 > >>

Role
itil
Insert a new row...

Condition

Caller | is | javascript:gs.getUserID

or Opened by | is | javascript:gs.getUserID

Update Delete

Access Control form for the Fuji release

Field	Description
Type	Select what kind of object this ACL rule secures. The type of object 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.
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. However, the nobody role takes precedence over the admin override option, so even admins cannot have access if they are assigned the nobody role. 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 **Requires role** field), use the condition builder or **Script** field to create a permissions check that administrators must pass.
Active	Select this check box to enforce this ACL rule.
Advanced	Select this check box to display the **Script** field.
Name	Enter the name of the object being secured, either the record name or the table and field names. 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. **Note:** The list shows only objects that meet the scope protections for ACL rules (starting with the Fuji release).
Description	[Optional] Enter a description of the object or permissions this ACL rule secures.
Requires role	Use this 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. **Note:** Users with the admin role will always pass this permissions check because the admin role automatically grants users all other roles. The **Requires role** list appears as an embedded list starting with the Fuji release. In previous releases, it is a related list.
Condition	Use this condition builder to select the fields and values that must be true for users to access the object.
Script	Enter a custom script describing the permissions required to access the object. The script can use the values of the **current** and **previous** global variables as well as system properties. The script must generate a true or false response in one of two ways: • return an **answer** variable set to a value of true or false • evaluate to true or false 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.

ACL Rules in Scoped Applications

Every ACL rule is assigned to either a unique scope or to the global scope (starting with the Fuji release). You can create ACL rules for objects in the same scope as the ACL rule and for tables with at least one field that is in the same scope as the ACL rule. For tables that are in a different scope than the ACL rule record, the types of rules are limited.

- You can create an ACL rule for any table, UI page, or other object that is in the same scope as the ACL rule.
- You can create an ACL for a field that is in the same scope as the ACL rule.
  - If the table is in the same scope, you can use a script to evaluate permissions.
  - If the table is in a different scope, you cannot use a script to evaluate permissions.
- You cannot create or modify ACL rules for objects that are in a different scope than the application you have selected in the application picker, including adding a role to an ACL in a different scope.
- You can create wildcard table rules (*) only in the global scope.
- You can create wildcard field rules (*) only for tables in the same scope as the ACL rule.

Granting or Denying Access

When a user attempts to access a particular object, the system searches for ACL rules that match the requested object's type, operation, and name. If an ACL rule matches these elements, 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, the system searches for the next matching ACL rule. For each matching ACL rule, the user has a chance to meet the required permissions in order to access the object. The system stops searching for matching ACL rules if the user ever meets a matching ACL rule's permissions. If the user cannot meet the permissions of any matching ACL rules, the system denies the user access to the object.

The effects of 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 **New** 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 **Delete** 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 define relationships between task tables.
edit_ci_relations	User cannot define relationships between Configuration Item [cmdb_ci] tables.
save_as_template	Used to control the fields that should be saved when a template is created.
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 **Configure Choices** (**Personalize Choices** in versions prior to Fuji).

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: 1. **All** wildcard ACL rules for the object (if any ACL rule exists for the operation). 2. The **first** ACL rule that matches the object's name (if any ACL rule exists for the operation).	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		
UI pages		
Record	Users must meet the permissions of two ACL rules: 1. The **first** ACL rule that matches the record's field (if any ACL rule exists for the operation). 2. The **first** ACL rule that matches the record's table (if any ACL rule exists for the operation).	By default, there are wildcard table rules (*) for the create, read, write, and delete operations and 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.

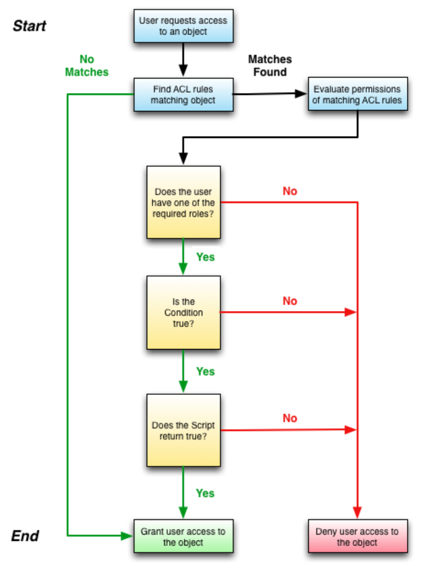
Note: *The high security property Security manager default behavior (`glide.sm.default_mode`) determines whether users can access objects that only match against wildcard table ACL rules. When this property is set to **Deny access**, only administrators can access objects that match the wildcard table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

- 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. If the list is empty, this condition evaluates to **true**.
- [Record ACL rules only] The matching table-level and field-level ACL rules must both evaluate to **true**.


Start
User requests access to an object
No Matches
Matches Found
Find ACL rules matching object
Evaluate permissions of matching ACL rules
Does the user have one of the required roles?
No
Yes
Is the Condition true?
No
Yes
Does the Script return true?
No
Yes
End
Grant user access to the object
Deny user access to the object

ACL Rule Workflow to Evaluate Permissions

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 tables. This operation is not valid for field ACL rules.
personalize_choices	Allows users to configure the table or field.

Processing Order for Record ACL Rules

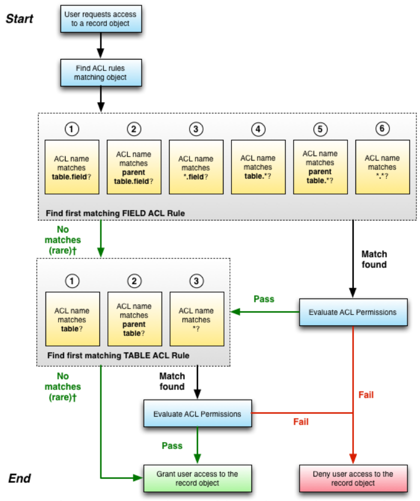
Record ACL rules are processed in the following order:

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, the user is denied access to the field described by the field ACL rule.
- If a user fails a table ACL rule, the user is denied access to all fields in the table even if the user previously passed a field ACL rule.


Start
User requests access to a record object
Find ACL rules matching object
1
ACL name matches table.field?
2
ACL name matches parent table.field?
3
ACL name matches *.field?
4
ACL name matches table.*?
5
ACL name matches parent table.*?
6
ACL name matches *.*?
Find first matching FIELD ACL Rule
No matches (rare)†
Match found
1
ACL name matches table?
2
ACL name matches parent table?
3
ACL name matches *?
Find first matching TABLE ACL Rule
Pass
Evaluate ACL Permissions
No matches (rare)†
Match found
Fail
Evaluate ACL Permissions
Fail
Pass
End
Grant user access to the record object
Deny user access to the record object

† High security settings ensure that all record objects and most record operations match to a global field ACL rule *.* or a global table ACL rule *. Only the report_on and personalize operations do not have global field and global table ACL rules to match against.

Matching Workflow for Record ACL Rules

Field ACL Rules

Field ACL rules are processed 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 matching evaluation stops ACL rule processing at that field level. This means that when a user passes or fails a field ACL rule, the system stops searching for matching field ACL rules below that level. For example, if there is a matching rule for the incident.number field, the system stops searching for matching field ACL rules such as task.number or incident.* because the user has already been granted or denied access to the field.

Note: *The user must also pass the table ACL rules to be granted access to the record object. For example, if a user passes the field ACL rule for incident.number, the system stops searching for field ACL rules, but the user must also pass any table ACL rules that match to the incident table.*

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 the base system includes wildcard table ACL rules that match every table, the user must always pass at least one table ACL rule. The base system provides additional table ACL rules to control access to specific tables.

Table ACL rules are processed 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, the system 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 two or more rules match at the same point in the processing order, the user must pass any one of the ACL rules permissions 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. For a list of available UI pages, navigate to **System UI > UI Pages**.

Note: *You cannot use the wildcard * character in the **Name** field on **ui_page** type ACLs to match any UI pages. You must enter the exact name of the UI pages in the **Name** field on the Access Control form.*

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 configure UI page records.

Because 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. 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, an ACL rule for the EmailClientProcessor is included 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. 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.

The base system does not include any ACL rules for client-callable script includes.

Debugging

The following ACL rule debugging tools are available:

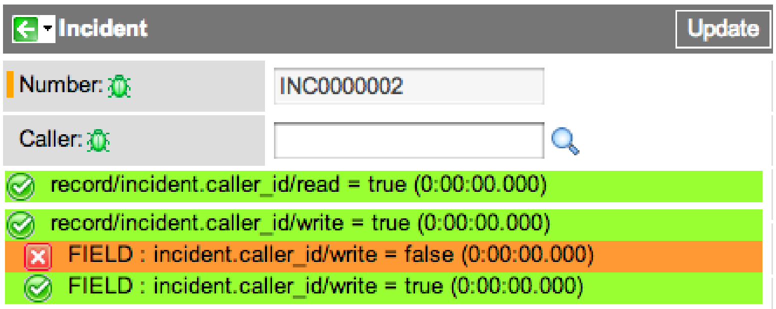
- Field level debugging
- ACL rule output messages

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

When debugging is enabled, a small bug icon (🐞) appears beside each field with an ACL rule. Clicking the icon lists the ACL rules that apply for the field and the evaluation results.



Number: INC0000002

Caller:

✓ record/incident.caller_id/read = true (0:00:00.000)

✓ record/incident.caller_id/write = true (0:00:00.000)

☒ FIELD : incident.caller_id/write = false (0:00:00.000)

✓ FIELD : incident.caller_id/write = true (0:00:00.000)

Field-level debugging

ACL Rule Output Messages

ACL debugging displays ACL rule output messages at the bottom of each list and form. The output message was redesigned in the Eureka release to:

- Improve readability
- Include context information
- Show the results of each type of ACL test
- To provide hyperlinks to the ACLs that run on the list or form.

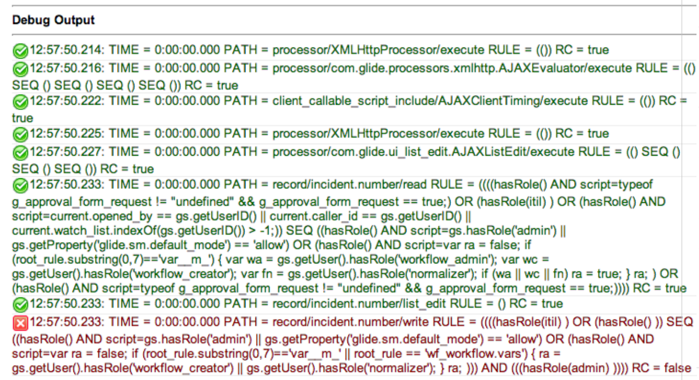
Each message displays the following information:

Message element	Description
TIME	The total time used to process this ACL rule.
PATH	Information that uniquely identifies each ACL rule in the format: *<ACL rule type>/<ACL rule name>/<Operation>*.
CONTEXT	The object being evaluated by the ACL rule. This element is available starting with the Eureka release.
RC	The return code of the ACL rule. A **true** value passes the ACL rule. A **false** value fails the ACL rule.
RULE	A brief summary of processors and scripts, followed by ACL results for each table-level and field-level ACL evaluation. Most ACL evaluations show an overall pass or fail result followed by a breakdown of the results for each type of ACL criteria: • **Role** • **Condition** • **Script**

The icons that appear show how the ACL was evaluated:

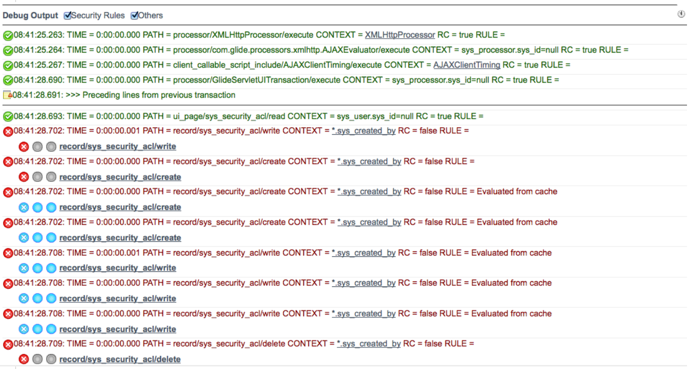
Icon	Description
A green checkmark (✓)	Indicates the table or field passed the criteria.
A red **x** (☒)	Indicates the table or field did not pass.
An empty gray circle (○)	Indicates the ACL evaluation did not need to be performed.
A blue checkmark, **x**, or empty circle	Indicates that the ACL was taken from a cached result of a previous ACL check. The icons mean the same as the above.

Click the name of the ACL next to any of the output messages to open that ACL record.



```
Debug Output
✓12:57:50.214: TIME = 0:00:00.000 PATH = processor/XMLHttpProcessor/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/XMLHttpProcessor/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(itil) ) 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(itil) ) 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
```

Output message prior to Eureka



```
Debug Output ☑Security Rules ☑Others
✓08:41:25.263: TIME = 0:00:00.000 PATH = processor/XMLHttpProcessor/execute CONTEXT = XMLHttpProcessor RC = true RULE =
✓08:41:25.264: TIME = 0:00:00.000 PATH = processor/com.glide.processors.xmlhttp.AJAXEvaluator/execute CONTEXT = sys_processor.sys_id=null RC = true RULE =
✓08:41:25.267: TIME = 0:00:00.000 PATH = client_callable_script_include/AJAXClientTiming/execute CONTEXT = AJAXClientTiming RC = true RULE =
✓08:41:28.690: TIME = 0:00:00.000 PATH = processor/GlideServletUITransaction/execute CONTEXT = sys_processor.sys_id=null RC = true RULE =
08:41:28.691: >>> Preceding lines from previous transaction
✓08:41:28.693: TIME = 0:00:00.000 PATH = ui_page/sys_security_acl/read CONTEXT = sys_user.sys_id=null RC = true RULE =
✗08:41:28.702: TIME = 0:00:00.001 PATH = record/sys_security_acl/write CONTEXT = *.sys_created_by RC = false RULE =
    ✗ ○ ○ record/sys_security_acl/write
✗08:41:28.702: TIME = 0:00:00.000 PATH = record/sys_security_acl/create CONTEXT = *.sys_created_by RC = false RULE =
    ✗ ○ ○ record/sys_security_acl/create
✗08:41:28.702: TIME = 0:00:00.000 PATH = record/sys_security_acl/create CONTEXT = *.sys_created_by RC = false RULE = Evaluated from cache
    ✗ ○ ○ record/sys_security_acl/create
✗08:41:28.702: TIME = 0:00:00.000 PATH = record/sys_security_acl/create CONTEXT = *.sys_created_by RC = false RULE = Evaluated from cache
    ✗ ○ ○ record/sys_security_acl/create
✗08:41:28.708: TIME = 0:00:00.001 PATH = record/sys_security_acl/write CONTEXT = *.sys_created_by RC = false RULE = Evaluated from cache
    ✗ ○ ○ record/sys_security_acl/write
✗08:41:28.708: TIME = 0:00:00.000 PATH = record/sys_security_acl/write CONTEXT = *.sys_created_by RC = false RULE = Evaluated from cache
    ✗ ○ ○ record/sys_security_acl/write
✗08:41:28.708: TIME = 0:00:00.000 PATH = record/sys_security_acl/write CONTEXT = *.sys_created_by RC = false RULE = Evaluated from cache
    ✗ ○ ○ record/sys_security_acl/write
✗08:41:28.709: TIME = 0:00:00.000 PATH = record/sys_security_acl/delete CONTEXT = *.sys_created_by RC = false RULE =
    ✗ ○ ○ record/sys_security_acl/delete
```

Output message starting with Eureka

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. • **Type:** true | false • **Default value:** false • **Location:** Add to the System Properties [sys_properties] table

Enhancements

Fuji

- Security restraints are on database views. You need to create a read ACL for your users on the tables in a view to generate reports on database views. Non-admin users do not have access to database view records unless a read ACL on the database view record allows access.
- These changes support developing scoped applications:
  - The **Name** field shows only objects that meet the scope protections for ACL rules.
  - For tables that are in a different scope than the ACL rule, the types of rules are limited.
  - The **Application** field is added on the form view.
- The list of required roles appears as an embedded list on the Access Control form, rather than as a related list.

Eureka

- The ACL rule output message was redesigned to improve readability and show additional information.
- Administrators can click the name of triggered ACLs to access the Access Control record.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/contextual-security/concept/access-control-rules.html

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

Note: *This article applies to Fuji and earlier releases. For more current information, see On-Call Scheduling* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

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?*

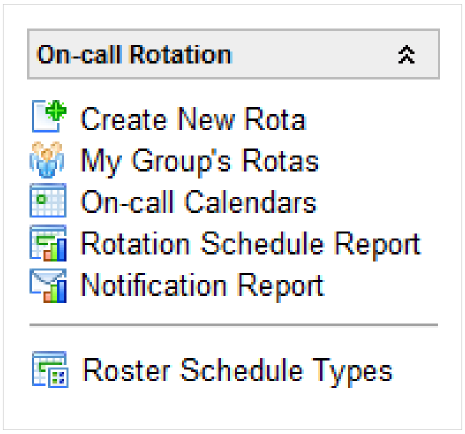
Note: *The on-call scheduling feature replaces group on-call rotation starting with the Eureka release.*

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.

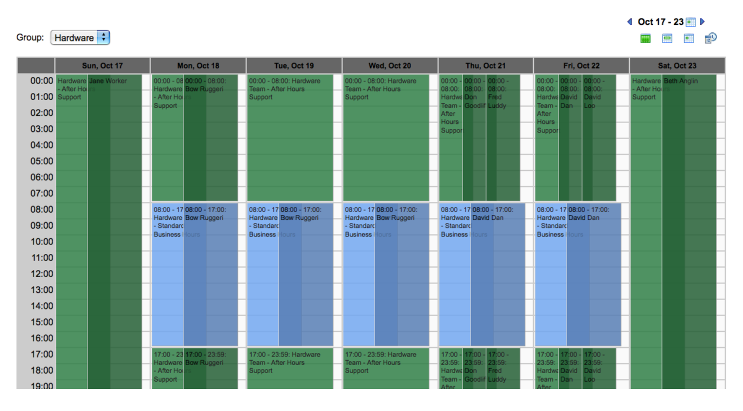
For more details, see Creating a New Roster.

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


Group: Hardware
Oct 17 - 23
Sun, Oct 17
Mon, Oct 18
Tue, Oct 19
Wed, Oct 20
Thu, Oct 21
Fri, Oct 22
Sat, Oct 23
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
Hardware - After Hours Support
Jane Worker
00:00 - 08:00: Hardware - After Hours Support
00:00 - 08:00: Bow Ruggeri
00:00 - 08:00: Hardware Team - After Hours Support
00:00 - 08:00: Don Goodlif
00:00 - 08:00: Fred Luddy
00:00 - 08:00: David Dan
00:00 - 08:00: David Loo
08:00 - 17:00: Hardware - Standard Business Hours
08:00 - 17:00: Bow Ruggeri
08:00 - 17:00: David Dan
17:00 - 23:59: Hardware - After Hours Support
17:00 - 23:59: Bow Ruggeri
17:00 - 23:59: Hardware Team - After Hours Support
17:00 - 23:59: Don Goodlif
17:00 - 23:59: Fred Luddy
17:00 - 23:59: David Dan
17:00 - 23:59: David Loo
Beth Anglin

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.

References

[1] https://docs.servicenow.com/bundle/jakarta-it-service-management/page/administer/user-administration/concept/c_OnCallScheduling.html

Domain Support Plugin

Note: *This article applies to Fuji and earlier releases. For more current information, see Domain Separation* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

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:

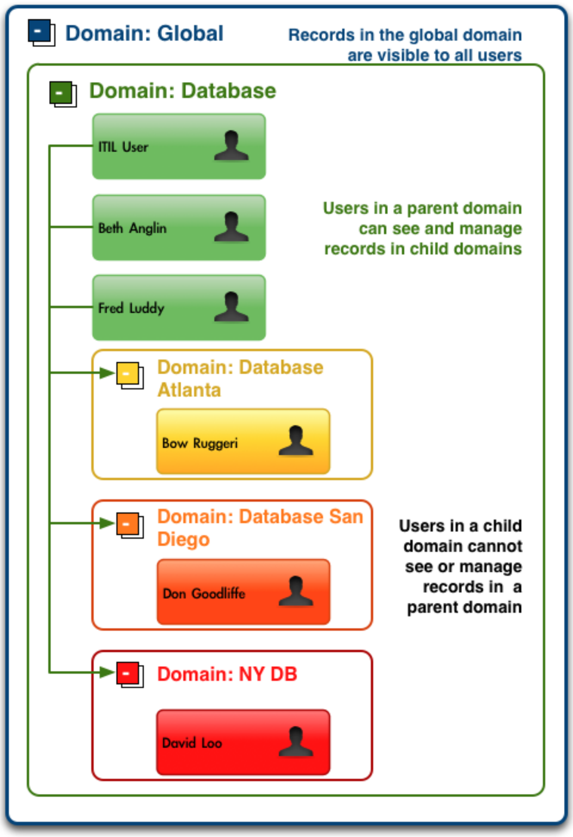
- Enforce absolute data segregation between business entities (data separation).
- Customize business process definitions and user interfaces for each domain (delegated administration).
- Maintain some global processes and global reporting in a single instance of ServiceNow.

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.

Warning: *Before* activating domain separation, consult your ServiceNow representative to verify that it is suitable for your environment. Domain separation adds a level of administration overhead. Although it can be disabled, it *cannot* be removed from an instance.

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:


Domain: Global
Records in the global domain are visible to all users
Domain: Database
ITIL User
Beth Anglin
Fred Luddy
Users in a parent domain can see and manage records in child domains
Domain: Database Atlanta
Bow Ruggeri
Domain: Database San Diego
Don Goodliffe
Users in a child domain cannot see or manage records in a parent domain
Domain: NY DB
David Loo

Sample 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.

Domain Assignment

By default, domain separation adds a domain field to the Task [task] and Configuration Item [cmdb_ci] tables and their extensions. You can also extend domain separation to any new tables you create by adding a **sys_domain** field to the table's dictionary definition. The tables provided with the ServiceNow system are domain separated where appropriate.

Note: *ServiceNow does not recommend domain separating platform tables such as the Dictionary Entry [sys_dictionary] and Dictionary Entry Override [sys_dictionary_override] tables because doing so can produce unexpected results.*

The value of the **sys_domain** field contains the domain assigned to the record by any of the following:

- Business rule

- Module
- Form template
- Parent record
- User who creates the record

Domain Separation Restrictions

Starting with the Fuji Release, the system prevents the following tables from being domain separated :

- Access Control [sys_security_acl]
- Script Include [sys_script_include]
- System Property [sys_properties]
- Security Black/Whitelist Entities [sys_security_restricted_list]
- Dictionary Entry [sys_dictionary]
- Dictionary Entry Override [sys_dictionary_override]

It is recommended that administrators do not domain separate these tables for versions prior to the Fuji release.

Assigning New Records to a Domain from a Business Rule

Administrators can use a business rule to automatically set a domain value when creating a record. The business rule must set a value in the `sys_domain` field. Administrators must ensure there is `sys_domain` column available for the record's table.

Assigning New Records to a Domain from a Module

Administrators can use the **sysparm_domain** URL parameter to automatically assign new records to a particular domain from a module. Administrators must create a new module with an **Argument** value of: `sysparm_domain=`*sys_ID of domain*. The **sysparm_domain** URL parameter is available starting with the Eureka release.

Assigning New Records to a Domain from a Form Template

Administrators can use a form template to automatically assign new records to a particular domain. Administrators must add the `sys_domain` field to the form and select a domain value. For example, setting the **sys_domain** field to **TOP/ACME domain** automatically assigns all records from this template to the TOP/ACME domain. Setting a domain from a form template is available starting with the Eureka release.

Assigning Related Records to a Domain Based on the Parent Record

By default, related records inherit the domain of the parent record. For example:

- A change task record inherits the domain of the parent change request record.
- A problem record inherits the domain of the parent incident record.

Related records inheriting the domain of the parent record is available starting with the Eureka release.

Assigning Records to a Domain Based on the User's Domain

If no other domain conditions apply, a record automatically inherits the domain of the user who creates it.

Domains Visible to Users in the Global Domain

By default, when a user in the global domain views a table containing a **sys_overrides** column, the user sees records from only the global domain. To view records from all domains, click **Expand Scope** under **Related Links**. To return to viewing records from the global domain only, click **Collapse Scope**. This feature is available starting with the Eureka release. This feature is enabled when the `glide.sys.restrict_global_domain_processes` property is set to **true**. If you are upgrading to Eureka, you can add this property.

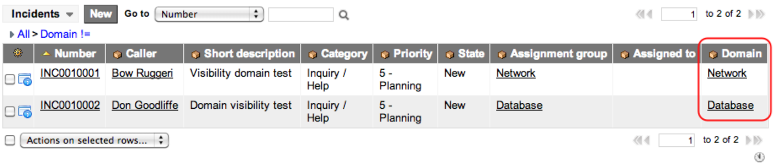
These links are named **Expand Domain Scope** and **Collapse Domain Scope** starting with the Fuji Release.

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.

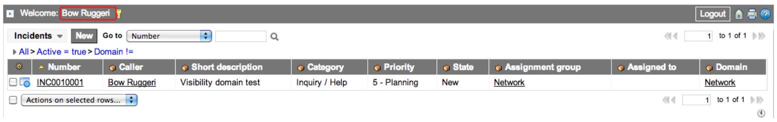
Note: *While visibility is one method to allow users to access records, it is recommended that you use* ***Contains*** *for more robust control. For more information on using* ***Contains****, see the Contains Domains section.*

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.



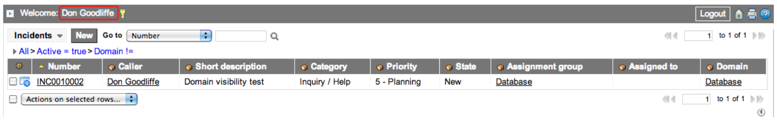
	Number	Caller	Short description	Category	Priority	State	Assignment group	Assigned to	Domain
□	INC0010001	Bow Ruggeri	Visibility domain test	Inquiry / Help	5 - Planning	New	Network		Network
□	INC0010002	Don Goodliffe	Domain visibility test	Inquiry / Help	5 - Planning	New	Database		Database

A sample set of domain separated incident records



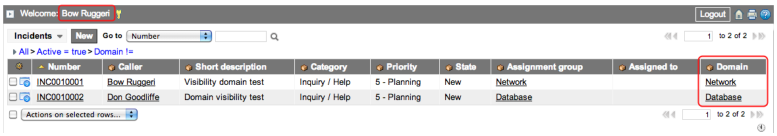
	Number	Caller	Short description	Category	Priority	State	Assignment group	Assigned to	Domain
□	INC0010001	Bow Ruggeri	Visibility domain test	Inquiry / Help	5 - Planning	New	Network		Network

Bow Ruggeri's incident list



	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

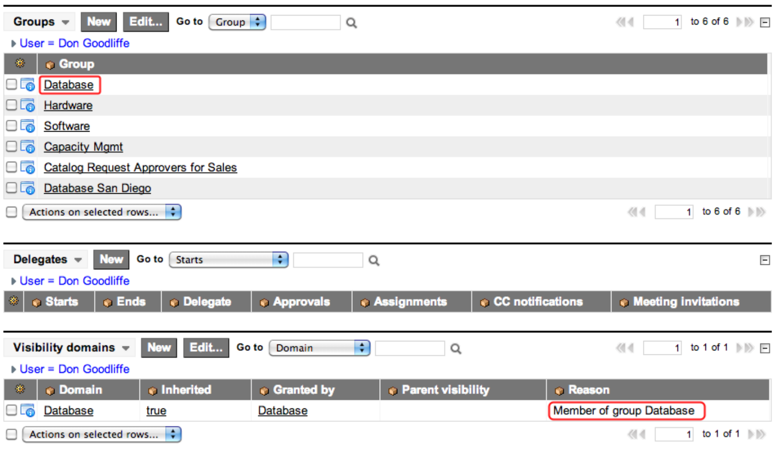


	Number	Caller	Short description	Category	Priority	State	Assignment group	Assigned to	Domain
□	INC0010001	Bow Ruggeri	Visibility domain test	Inquiry / Help	5 - Planning	New	Network		Network
□	INC0010002	Don Goodliffe	Domain visibility test	Inquiry / Help	5 - Planning	New	Database		Database

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, Don Goodliffe also automatically gains the **Database** domain as a visibility domain. Group membership grants visibility to any matching domain name.



Group
Database
Hardware
Software
Capacity Mgmt
Catalog Request Approvers for Sales
Database San Diego

Starts	Ends	Delegate	Approvals	Assignments	CC notifications	Meeting invitations

Domain	Inherited	Granted by	Parent visibility	Reason
Database	true	Database		Member of group Database

Visibility domains granted by group membership

Domain Scope

Every user has two domain scopes when establishing a session in a domain separated instance.

- **Session scope:** is set upon session establishment to the domain listed in the user's user record. Users can manually change their session domain scope from the domain picker.
- **Record scope:** uses the domain of the record and is active when viewing the form of any record.

By default, the record scope takes precedence over the session scope so that fulfillers in higher level domains adhere to each record's data and process constraints. However, these fulfillers can choose to *expand* or *collapse* the domain scope to show or hide data from other domains. For example, a user in the MSP domain also has visibility into child domains such as the ACME domain. When looking at an incident record from the ACME domain, the user can choose to expand the domain scope to show values from the MSP domain or collapse the domain scope to only show record values that match the record's ACME domain.

Note: *Users always have access to data from domains that have been explicitly granted to them by domain visibility.*

Users with the domain_expand_scope user role can select the domain scope from the **Toggle Domain Scope** UI action on the form. When record scope is in effect, click the UI action to expand to session scope and display all data available based to the user's domain and child domains. When session scope is in effect, click the UI action to collapse to record scope and display only data that matches the current record's domain.

Note: *A record will not display the UI action to toggle the domain scope if the record is in the global domain or if the user's domain matches the record's domain.*

The option to select the domain scope is available starting with the Fuji release.

Selecting Record Values from Other Domains

Users who can see multiple domains have the option to select record values from a domain that is different than the record's domain. For example, service desk agents working for a managed service provider might want to assign certain incidents to themselves to resolve issues on behalf of their customers. When they do this, the incident **Assigned to** field might contain a user from the MSP domain, even though the incident record itself is associated with a child domain such as ACME.

Selecting a record value from another domain does not change the record's domain. The record retains its original domain. When a user views a record with values from multiple domains, the user's domain visibility determines what they see.

When these conditions are met	The user has access to these UI elements
The user has access to the domain of the current record referenced in a field.	The user can: • See reference field display value. For example, sees the user name in the **Assigned to** field. • See the related record from reference icon. For example, sees the user record for the user in the **Assigned to** field. • Select values from any visible domain. For example, can select users from either the MSP and ACME domains.
The user does not have access to the domain of the current record referenced in a field.	The user can: • See the reference field display value. For example, sees the user name in the **Assigned to** field. • Only select values from the record's domain. For example, can only select user's from the ACME domain.

The option to select values from visibility domains is available starting with the Fuji release.

Contains Domains

Normally parent-child relationships define the domain hierarchy. A **Contains** domain allows you to relate domains on an as-needed basis, independent of parent-child relationships. However, contains domains only grant visibility to domain data. Processes remain unaffected by contains relationships.

Note: *Visibility controls what a particular user can see, while* ***Contains*** *controls what an entire domain of users can see.*

Contains Domains Versus Visibility Domains

Contains domains and visibility domains differ in several respects.

A contains domain:

- Is a many-to-many, domain-to-domain relationship.
- Is hierarchical. When a domain is selected, you can see the data from that domain and its children.
- Is controlled by the selection in the domain picker.

A visibility domain:

- Is a user-to-domain relationship and is explicitly granted.
- Is not hierarchical.
- Is not controlled by the selection in the domain picker. Once the user is granted access to a visibility domain, they always see data in that domain and its children.

For example, there is a user who has access to domain A (the user's home domain) and is granted visibility to domains B and C. The user selects domain A in the domain picker. In this case, the user has access to domains A, B, and C. If the user changes the domain picker to domain B, B and C are visible. C is still visible because the user still has visibility to it. A is not visible, because it is not selected in the domain picker and it is not a visibility domain.

Using visibility domains excessively is not recommended.

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. While in a domain, administrators can set domain-specific versions of these global policies and settings:

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

Warning: All users with the admin role have special access to all system features, functions, and data because administrators can override ACL rules and pass all role checks. Grant this privilege carefully.

When users have the **admin** role, then all policies in the instance are available to them regardless of the assigned domain. They can enter a specific domain, and then only policies in that domain or higher are visible and processed during a relevant transaction. When an administrator modifies a policy that is in a higher domain or the global domain, the system automatically creates a new record for that administrator's current domain. It does not modify the original policy, application, or module record. This new record *overrides* the original.

Note: *To make changes to a policy in a lower-level domain, go into that domain and modify the policy. This approach creates the new policy record in your domain that overrides the original, higher-level policy record.*

Do not *simply make changes on the higher-level policy and then change the* ***Domain*** *field on that policy. This approach does not create a new policy record in your lower-level domain, nor does it keep the policy record for the higher-level domain.*

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 an administrator attempts to modify a policy, application, or module belonging to another domain higher in the hierarchy. 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 instead of the original.

Determining the Domain Used for Delegated Administration

By default, delegated administration always uses the record's domain to determine what policies to apply (starting with the Eureka release). The record's domain takes precedence over the user's domain. If there are no policies in the record's domain, delegated administration checks for policies in the next highest level of the domain hierarchy. The search for domain policies continues up the domain hierarchy until reaching the global domain. If there are no domain policies lower in the domain hierarchy, delegated administration uses the policies for the global domain.

For example, Fred Luddy is a user in the **Database** domain who can see records in the **Database: Atlanta**, **Database: San Diego**, and **NY DB** child domains. When he opens a record in the **Database: San Diego** domain, delegated administration first checks for policies in the **Database: San Diego** domain. If there are no policies at this level of the domain hierarchy, delegated administration checks for policies from the **Database** domain. If there are no policies in the **Database** domain, delegated administration uses the global domain polices as there are no other domains higher in the domain hierarchy.

Click the plus for versions prior to Eureka

In versions prior to Eureka, the user's domain determines what policies to apply unless configured to use the record's domain. The user's domain takes precedence over the record's domain. If there are no policies in the user's domain, delegated administration checks for policies in the next highest level of the domain hierarchy. The search for domain policies continues up the domain hierarchy until reaching the global domain. If there are no domain policies lower in the domain hierarchy, delegated administration uses the policies for the global domain.

Note: *When configured to use the record's domain instead of the user's domain, delegated administration functions as described for Eureka for form loads and reference fields on those forms only. If any transactions happen after the form load outside of reference field selections, the domain reverts to the users domain.*

Example Delegated Administration with Domain Specific Policies

Click the plus to see example

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

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

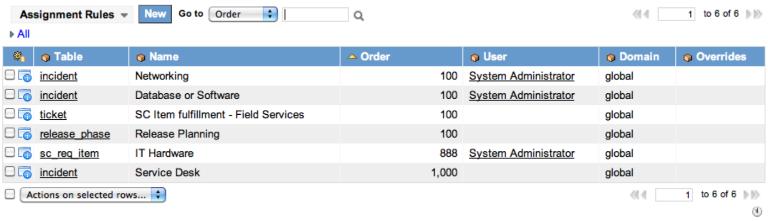


	Table	Name	Order	User	Domain	Overrides
□	incident	Networking	100	System Administrator	global	
□	incident	Database or Software	100	System Administrator	global	
□	ticket	SC Item fulfillment - Field Services	100		global	
□	release_phase	Release Planning	100		global	
□	sc_req_item	IT Hardware	888	System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

Global domain rules

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

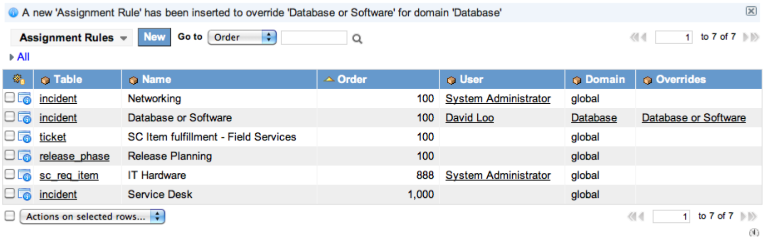

A new 'Assignment Rule' has been inserted to override 'Database or Software' for domain 'Database'

	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	

Database domain-specific rules

The following policy changes occur:

- When the policy is chosen and updated, the system detects that David Loo is not at the right level of the hierarchy to change this record. Therefore, the update is changed into an insert, and a new record is created.
- The new policy has the same name (**Database or Software**). Notice that this policy is in the **Database** domain and overrides the policy that previously applied (**Database or Software**).

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.

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

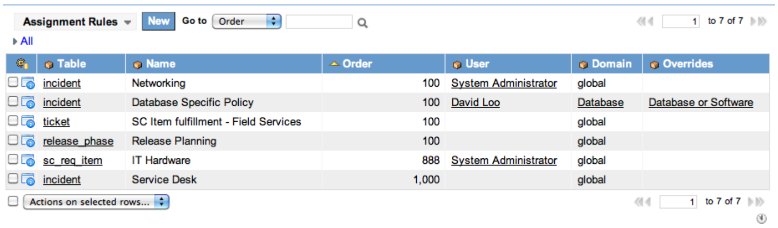
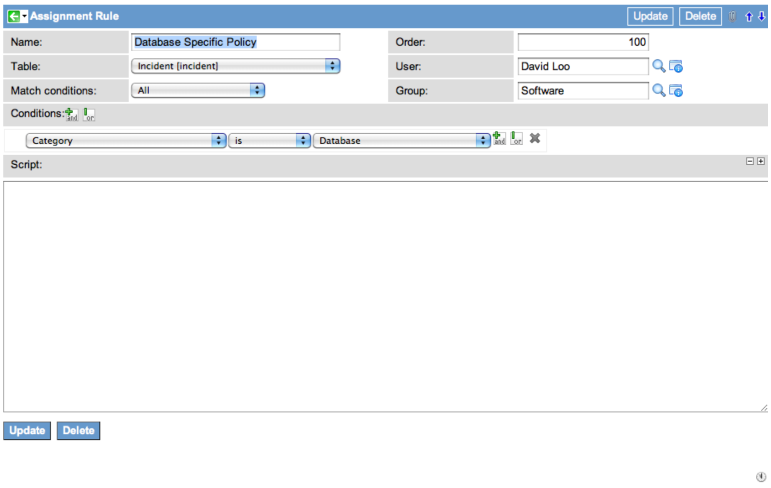

Assignment Rules ▾ New Go to Order ⇕ 1 to 7 of 7
All

	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	

□ Actions on selected rows... 1 to 7 of 7

Renamed Database or Software to Database Specific Policy

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.


Assignment Rule Update Delete

Name:	Database Specific Policy	Order:	100
Table:	Incident [incident]	User:	David Loo
Match conditions:	All	Group:	Software

Conditions:

Category is Database

Script:

Update Delete

Database Specific Policy assignment rule

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 starts out with the following assignment policy:

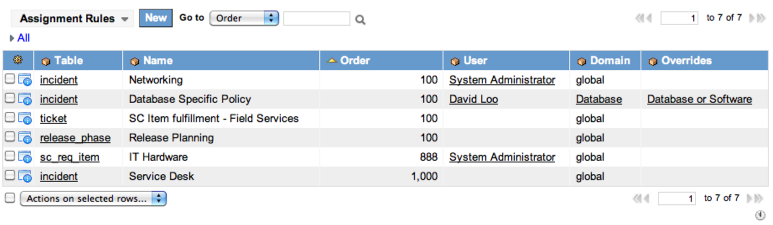
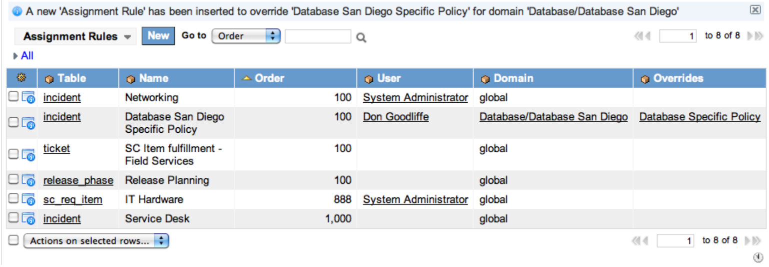

Assignment Rules ▾ New Go to Order ⇕ 1 to 7 of 7
All

	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	

□ Actions on selected rows... 1 to 7 of 7

Don Goodliffe's starting view of assignment rules

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**, the list look like this:


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

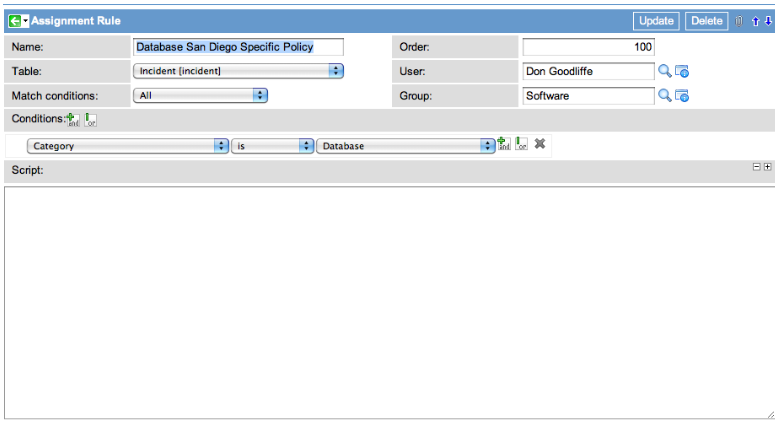
Assignment Rules ▾ New Go to Order ⇕ 1 to 8 of 8
All

	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	

□ Actions on selected rows... 1 to 8 of 8

Database San Diego rules override Database Specific Policy rules

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



Name:	Database San Diego Specific Policy	Order:	100
Table:	Incident [incident]	User:	Don Goodliffe
Match conditions:	All	Group:	Software

Conditions:

Category · is · Database

Script:

San Diego Specific Policy

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 at any level of the domain hierarchy.

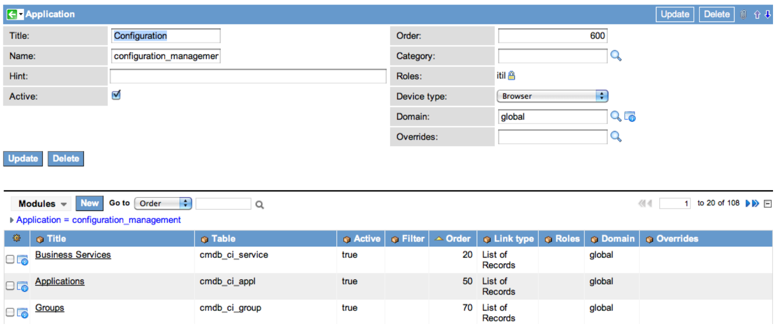
During a transaction, the current user's domain normally determines the policy to load. For example when a user in the Database domain updates an incident, the Database domain is used for business rules and policies even if the incident record was originally created in the *Database San Diego* domain.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

Click the plus to see example

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.



Title:	Configuration	Order:	600
Name:	configuration_managemer	Category:	
Hint:		Roles:	itil
Active:	☑	Device type:	Browser
		Domain:	global
		Overrides:	

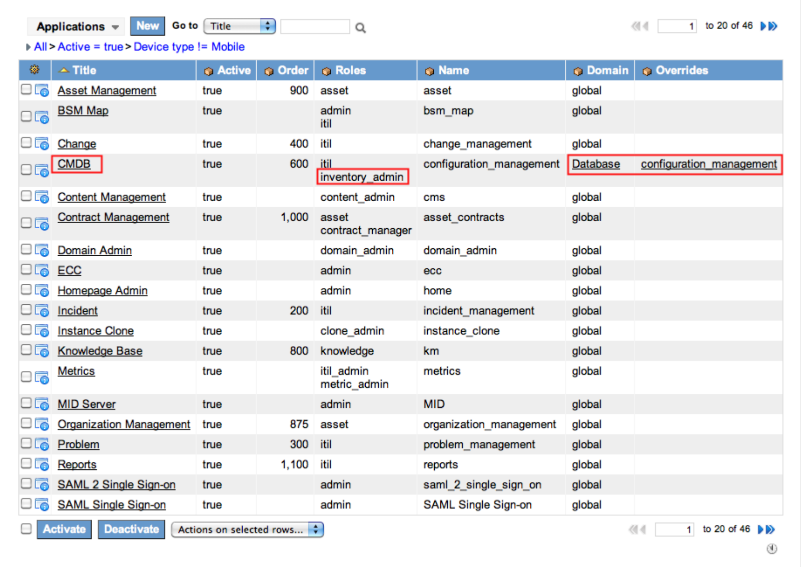
Application = configuration_management

Title	Table	Active	Filter	Order	Link type	Roles	Domain	Overrides
Business Services	cmdb_ci_service	true		20	List of Records		global	
Applications	cmdb_ci_appl	true		50	List of Records		global	
Groups	cmdb_ci_group	true		70	List of Records		global	

Starting view of the Configuration application (excerpt)

David decides to rename the Configuration application to CMDB and to allow the inventory_admin role to see the 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.

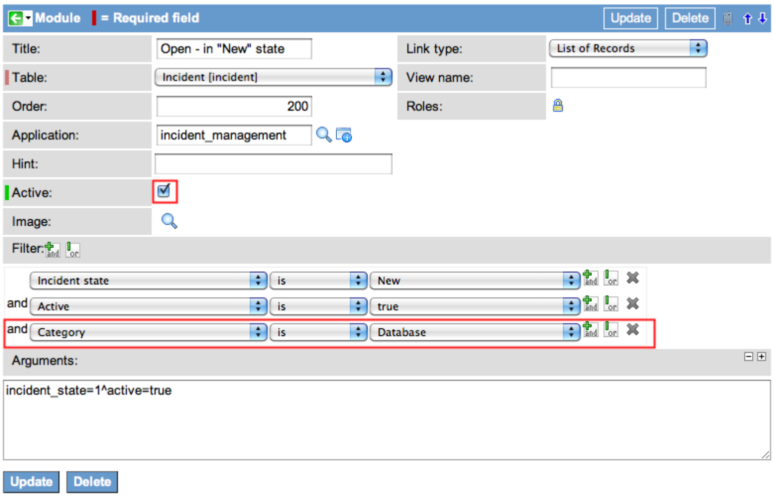


All > Active = true > Device type != Mobile

	Title	Active	Order	Roles	Name	Domain	Overrides
□	Asset Management	true	900	asset	asset	global	
□	BSM Map	true		admin itil	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	

Sample domain-specific changes to the Configuration application

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



Title:	Open - in "New" state	Link type:	List of Records
Table:	Incident [incident]	View name:	
Order:	200	Roles:	
Application:	incident_management		
Hint:			
Active:	☑		
Image:			

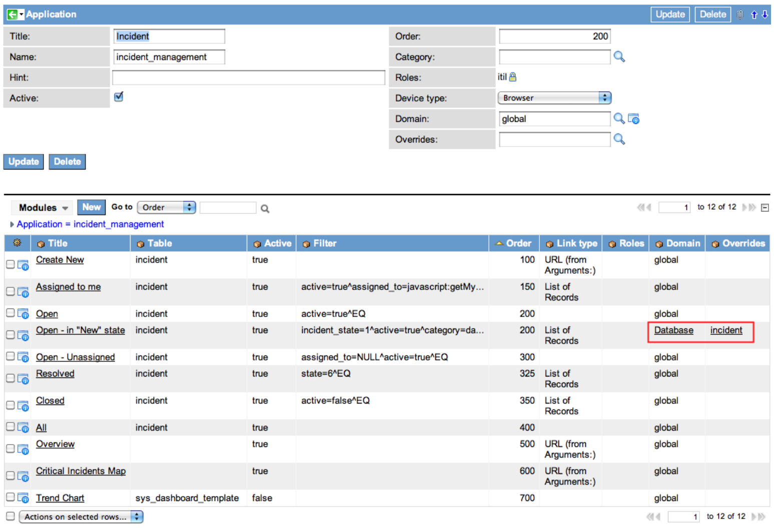
Filter:

	Incident state	is	New
and	Active	is	true
and	Category	is	Database

Arguments:

```
incident_state=1^active=true
```

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



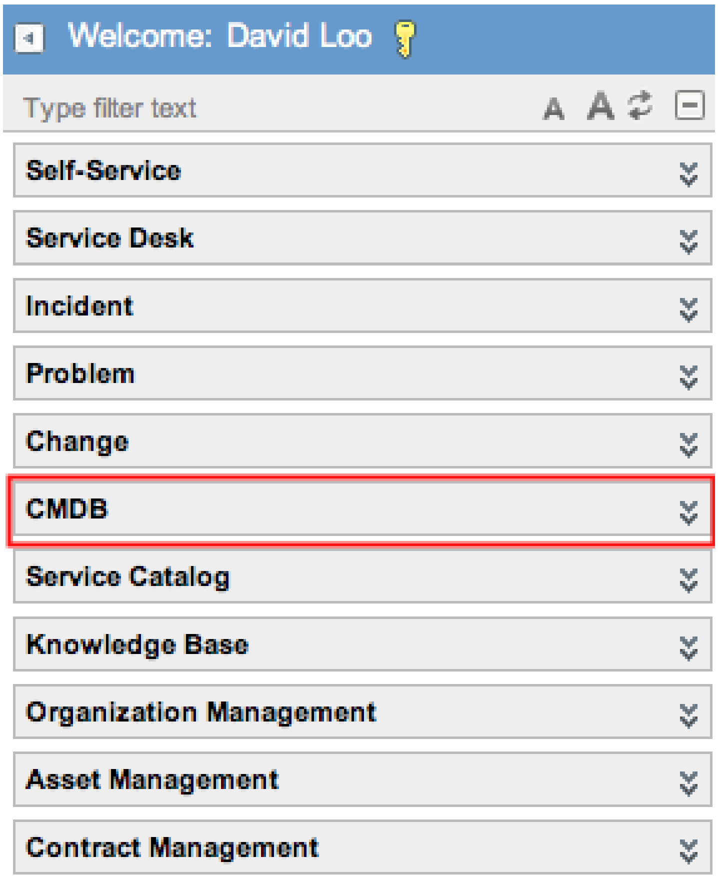
Title:	Incident	Order:	200
Name:	incident_management	Category:	
Hint:		Roles:	itil
Active:	☑	Device type:	Browser
		Domain:	global
		Overrides:	

Application = incident_management

	Title	Table	Active	Filter	Order	Link type	Roles	Domain	Overrides
□	Create New	incident	true		100	URL (from Arguments:)		global	
□	Assigned to me	incident	true	active=true^assigned_to=javascript:getMy...	150	List of Records		global	
□	Open	incident	true	active=true^EQ	200			global	
□	Open - in "New" state	incident	true	incident_state=1^active=true^category=da...	200	List of Records		Database	incident
□	Open - Unassigned	incident	true	assigned_to=NULL^active=true^EQ	300			global	
□	Resolved	incident	true	state=6^EQ	325	List of Records		global	
□	Closed	incident	true	active=false^EQ	350	List of Records		global	
□	All	incident	true		400			global	
□	Overview		true		500	URL (from Arguments:)		global	
□	Critical Incidents Map		true		600	URL (from Arguments:)		global	
□	Trend Chart	sys_dashboard_template	false		700			global	

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.



David Loo's view of applications

Domain Query Methods

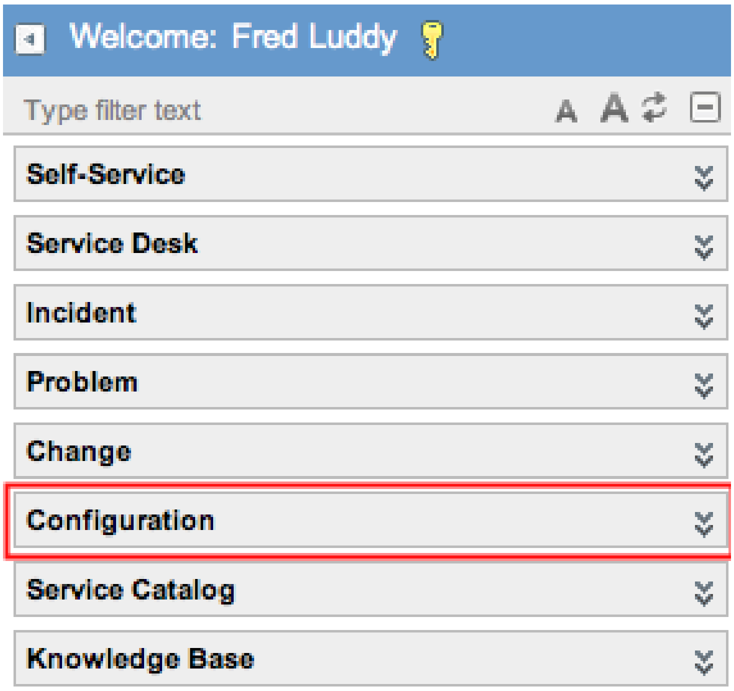
A domain query method allows the instance to efficiently query large numbers of domains. There are two domain query methods.

- Domain paths
- (Legacy) Domain numbers. Domain numbers are no longer supported starting with the (Eureka release).

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

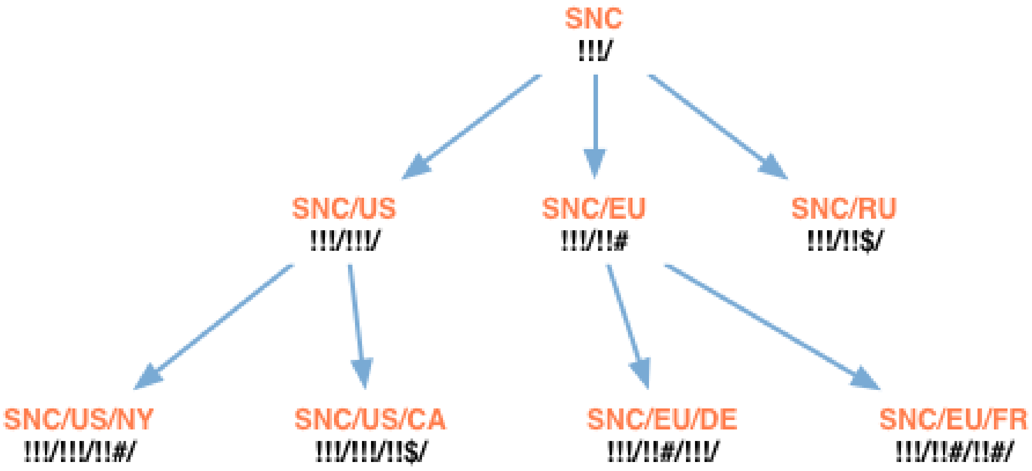
A domain path is a series of three-character codes separated by a slash (/) delimiter that uniquely identifies a domain. Each digit in the three-character code consists of one of the following 60 possible characters:



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

Fuji

- Domain separation allows users who can see multiple domains the ability to select the domain scope and select record values from other domains without changing the record's domain.
- The domain reference pickers provide access to all domains that the user has access to, regardless of which domain the user is currently in.
- The property `glide.domain.strict_override` is available to force the system to show overridden records. When this property is set to **true**, records in the parent domain do not appear when an overridden copy of the records exist.
- The **Domain Support - MSP Extensions** plugin is named **Domain Support - Domain Extensions**.

Eureka

- Users in the global domain can choose to view data from all domains or just the global domain.
- Administrators have additional options to assign records to domains.
- Delegated administration uses the record's domain when searching for policies.

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.sys.domain.domain_change_notify`: 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.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/company-and-domain-separation/reference/domain-sep-landing-page.html

Role Delegation Plugin

Note: *This article applies to Fuji and earlier releases. For more current information, see Define Role Delegators and Delegate Roles* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.'**

Overview

Administrators can grant users the right to be role delegators. These delegators can assign roles to users who are in a particular group. The roles that delegators can assign to other users include the roles that the delegator inherits from a group those roles that the administrator specifies.

Users assigned the role_delegator role can act as delegators.

Role Delegation and Record Producers

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

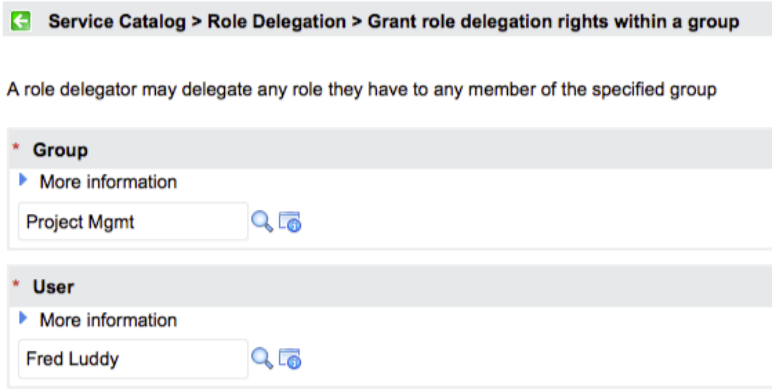
- *Grant role_delegator role to user in group*
- *Delegate roles to group member* graphical workflows.

These workflows can be customized as desired to add approval steps. For more information about workflows, see Workflow Overview.

Designating a Role Delegator

To designate a role delegator:

1. Navigate to **User Administration > Designate Role Delegator**.
2. Select the group that includes the user who you want to be the role delegator.
3. Select the user.



Service Catalog > Role Delegation > Grant role delegation rights within a group

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

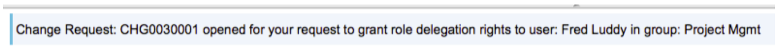
* **Group**
More information
Project Mgmt

* **User**
More information
Fred Luddy

Assigning a role delegator

4. Click **Submit**.

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



Change Request: CHG0030001 opened for your request to grant role delegation rights to user: Fred Luddy in group: Project Mgmt

The change request for the assignment of a role delegator

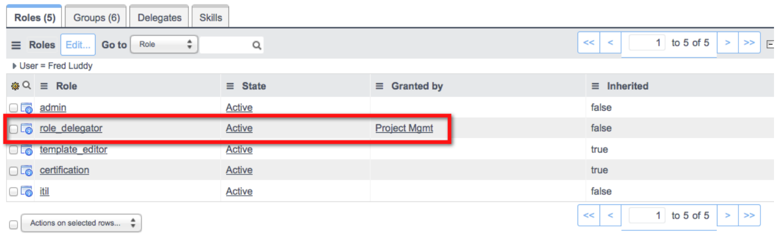
Viewing Delegated Roles

An administrator can view role designation in the following locations:

- User records
- The Role Delegators module
- The Role Audit module

User Records

Open a user record by navigating to **User Administration > Users** and selecting the user. You can see all the roles assigned to that user in the **Roles** related list.



User = Fred Luddy

Role	State	Granted by	Inherited
admin	Active		false
role_delegator	Active	Project Mgmt	false
template_editor	Active		true
certification	Active		true
itil	Active		false

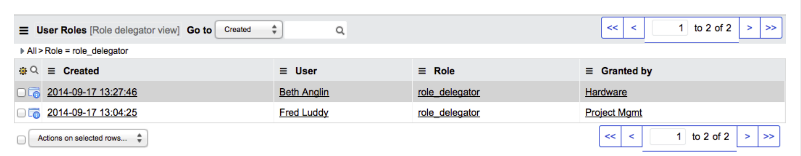
A user with the role_delegator role

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 and the groups they belong to are listed.

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 > Role Audit**.



User Roles [Role delegator view] Go to Created

All > Role = role_delegator

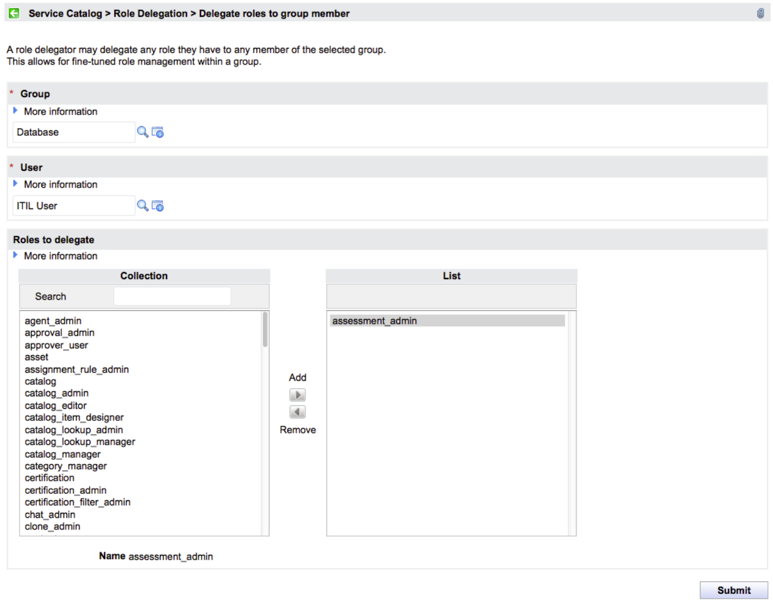
Created	User	Role	Granted by
2014-09-17 13:27:46	Beth Anglin	role_delegator	Hardware
2014-09-17 13:04:25	Fred Luddy	role_delegator	Project Mgmt

List of role delegators

Delegating Roles

To delegate specific roles to members of a group, navigate to **User Administration > Delegate Roles in Group** and fill out the form (see table).

Note: *This module is available to users with the role_delegator role.*



Service Catalog > Role Delegation > 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
Database

* User
More information
ITIL User

Roles to delegate
More information

Collection:
agent_admin
approval_admin
approver_user
asset
assignment_rule_admin
catalog
catalog_admin
catalog_editor
catalog_item_designer
catalog_lookup_admin
catalog_lookup_manager
catalog_manager
category_manager
certification
certification_admin
certification_filter_admin
chat_admin
clone_admin

List:
assessment_admin

Name assessment_admin

Delegating roles

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.

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.

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.

Preventing Roles from being Delegated

By default, the following fields are not delegatable:

- **admin**
- **public**
- **nobody**
- **role_delegator:** A user with the role_delegator role cannot delegate *this* role to other group members.

To prevent other roles from being delegated to users:

1. Navigate to **User Administration > Roles**.
2. Open the role.
3. Configure the form to add the **Grantable** or **Can delegate** fields.
4. Clear the check box for one or both of these fields.
5. Click **Update**.

Using the Group Manager Change Business Rule

The **Group Manager Change** business rule, which is disabled by default, will automatically grant the role_delegator role to a user when they become manager of a group by 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, activate it. See Business Rules for more information on accessing and enabling business rules.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/roles/task/t_RoleDelegation.html

User Self-Registration Plugin

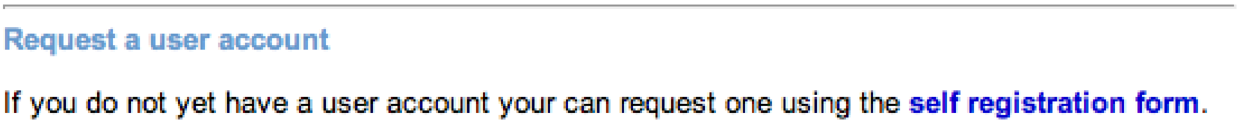
Note: *This article applies to Fuji. For more current information, see User Self-Registration* [1] *at* http://docs.servicenow.com The ServiceNow Wiki is no longer being updated. Please refer to http://docs.servicenow.com for the latest product documentation.

Overview

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

Requesting an Account

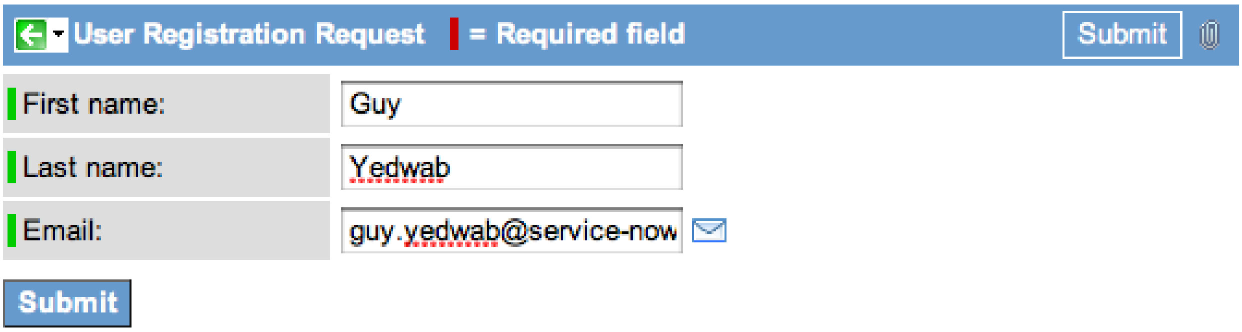
If a user would like to request an account, they navigate directly to the instance. If the plugin is installed, the following section is added to the welcome screen:



Request a user account

If you do not yet have a user account your 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:

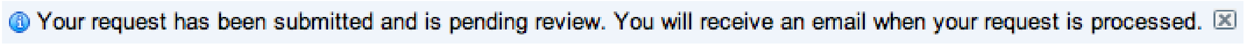


First name: Guy

Last name: Yedwab

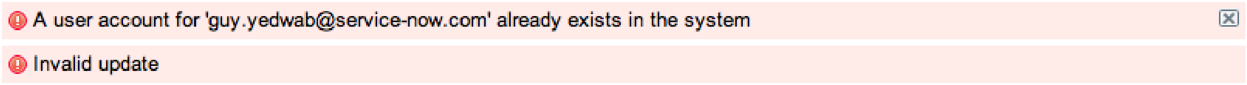
Email: guy.yedwab@service-now

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:

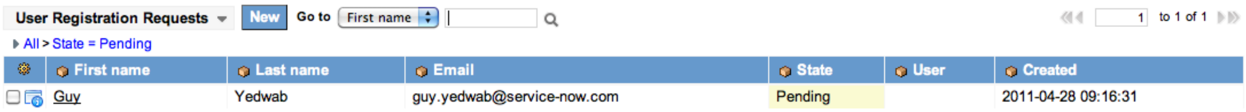


A user account for 'guy.yedwab@service-now.com' already exists in the system

Invalid update

Approving Accounts

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

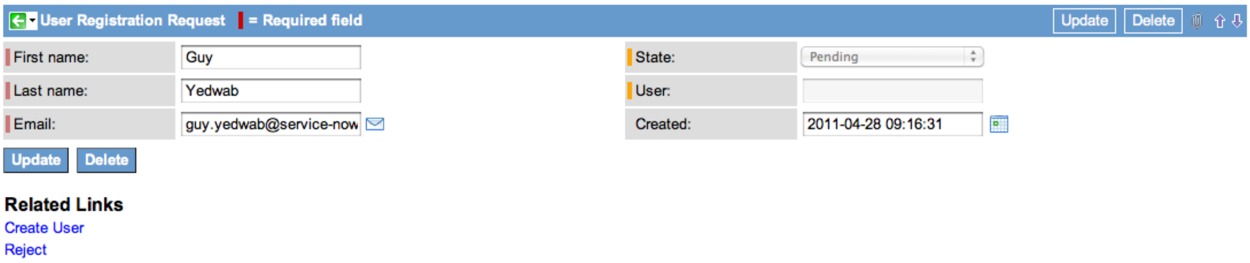


User Registration Requests ▾ New Go to First name 1 to 1 of 1

All > State = Pending

	First name	Last name	Email	State	User	Created
□	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:	Guy	State:	Pending
Last name:	Yedwab	User:	
Email:	guy.yedwab@service-now	Created:	2011-04-28 09:16:31

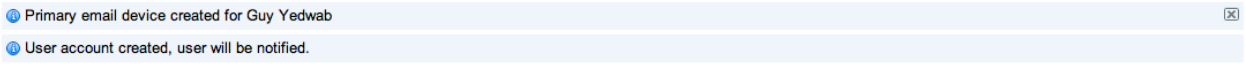
Update Delete

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:

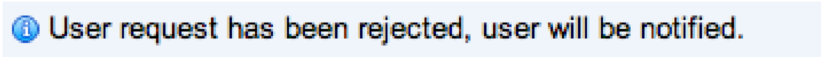


Primary email device created for Guy Yedwab

User account created, user will be notified.

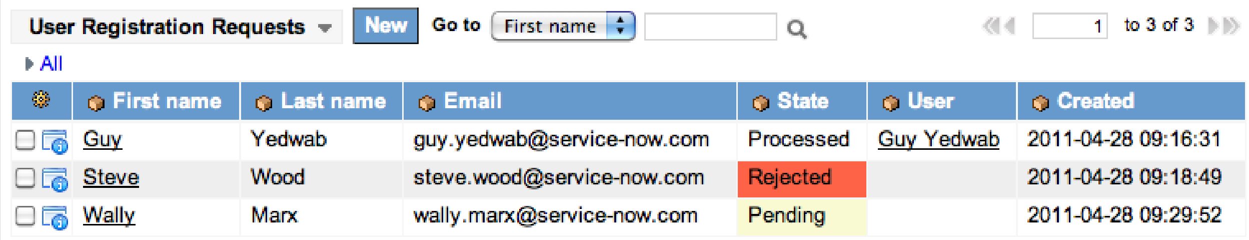
The user will be informed by an email notification.

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 First name 1 to 3 of 3

All

	First name	Last name	Email	State	User	Created
□	Guy	Yedwab	guy.yedwab@service-now.com	Processed	Guy Yedwab	2011-04-28 09:16:31
□	Steve	Wood	steve.wood@service-now.com	Rejected		2011-04-28 09:18:49
□	Wally	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.

If you have the admin role, use the following steps to activate the 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 are listed along with their activation status.
3. [Optional] If available, 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 you first activate 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**.

References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/users-and-groups/concept/c_UserRegistration.html

Article Sources and Contributors

User Administration *Source*: http://wiki.servicenow.com/index.php?oldid=250977 *Contributors*: Cheryl.dolan, David.Bailey, G.yedwab, Guy.yedwab, Joe.Westrich, John.ramos, Joseph.messerschmidt, Vhearne

Managing User Sessions *Source*: http://wiki.servicenow.com/index.php?oldid=250326 *Contributors*: CapaJC, Christen.mitchell, David Loo, Emily.partridge, Fuji.publishing.user, G.yedwab, Guy.yedwab, John.ramos, Joseph.messerschmidt, Neola, Phillip.salzman, Rachel.sienko, Suzanne.smith, Tricia.luke, Vaughn.romero, Vhearne

Creating Roles *Source*: http://wiki.servicenow.com/index.php?oldid=250449 *Contributors*: Cheryl.dolan, Emily.partridge, Fuji.publishing.user, Guy.yedwab, John.ramos, Joseph.messerschmidt, Neola, Phillip.salzman, Publishing.user, Roy.lagemann, Steven.wood, Suzanne.smith

Counting Licensed Users *Source*: http://wiki.servicenow.com/index.php?oldid=243800 *Contributors*: Fuji.publishing.user, Gadi.yedwab, Guy.yedwab, Joseph.messerschmidt, Neola, Roy.lagemann, Steven.wood, Suzanne.smith

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

Impersonating a User *Source*: http://wiki.servicenow.com/index.php?oldid=250628 *Contributors*: CapaJC, Eric.jacobson, Guy.yedwab, Joe.zucker, John.ramos, Joseph.messerschmidt, Mark.stanger, Neola, Pat.Casey, Phillip.salzman, Rachel.sienko, Steven.wood, Vhearne

Skills Management *Source*: http://wiki.servicenow.com/index.php?oldid=250894 *Contributors*: Cheryl.dolan, Dawn.bunting, Emily.partridge, Fuji.publishing.user, G.yedwab, Guy.yedwab, John.ramos, John.roberts, Joseph.messerschmidt, Neola, Peter.smith, Rachel.sienko, Steven.wood, Wallymarx

Defining Locations *Source*: http://wiki.servicenow.com/index.php?oldid=251178 *Contributors*: Cheryl.dolan, Emily.partridge, Fuji.publishing.user, Guy.yedwab, John.ramos, Joseph.messerschmidt, Mark.stanger, Phillip.salzman, Publishing.user, Rachel.sienko, Steven.wood

Creating Groups *Source*: http://wiki.servicenow.com/index.php?oldid=250436 *Contributors*: Ashley.robinson, CapaJC, Cheryl.dolan, Emily.partridge, Fuji.publishing.user, G.yedwab, Guy.yedwab, Jennifer.thorburn, Jeremiah.hall, John.ramos, Joseph.messerschmidt, Neola, Peter.smith, Phillip.salzman, Suzanne.smith, Vaughn.romero, Vhearne

Associating Users to Groups *Source*: http://wiki.servicenow.com/index.php?oldid=250220 *Contributors*: Anat.kerry, CapaJC, Cheryl.dolan, Emily.partridge, Fuji.publishing.user, Guy.yedwab, Joe.Westrich, John.ramos, Joseph.messerschmidt, Neola, Phillip.salzman, Rachel.sienko, Steven.wood, Suzanne.smith, Vaughn.romero, Vhearne

Granting Access *Source*: http://wiki.servicenow.com/index.php?oldid=60514 *Contributors*: Cheryl.dolan, Emily.partridge, Fuji.publishing.user, Guy.yedwab, Jessi.graves, John.ramos, Joseph.messerschmidt, Neola, Phillip.salzman, Rachel.sienko, Steven.wood, Vaughn.romero, Wallymarx

Using Access Control Rules *Source*: http://wiki.servicenow.com/index.php?oldid=250724 *Contributors*: CapaJC, Emily.partridge, Fuji.publishing.user, G.yedwab, Grant.hulbert, Guy.yedwab, John.ramos, Joseph.messerschmidt, Julie.phaviseth, Neil.narvaez, Neola, Phillip.salzman, Rachel.sienko, Steven.wood, Suzanne.smith, 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, Steven.wood, Vhearne

Group On-Call Rotation Plugin *Source*: http://wiki.servicenow.com/index.php?oldid=250615 *Contributors*: Cheryl.dolan, John.ramos, Ludwig.adriaansen

Domain Support Plugin *Source*: http://wiki.servicenow.com/index.php?oldid=122820 *Contributors*: CapaJC, Cheryl.dolan, Don.Goodliffe, Fuji.publishing.user, G.yedwab, Gadi.yedwab, Guy.yedwab, Jared.laethem, John.ramos, Joseph.messerschmidt, Michael.hoefer, Neil.narvaez, Neola, Nick.roberts, Phillip.salzman, Rachel.sienko, Richard.motteram, Rob.phillips, Roy.lagemann, Steven.wood, Vaughn.romero, Vhearne, Wallymarx

Role Delegation Plugin *Source*: http://wiki.servicenow.com/index.php?oldid=89883 *Contributors*: CapaJC, Emily.partridge, Fuji.publishing.user, John.ramos, Joseph.messerschmidt, Neola, Pat.Casey, Phillip.salzman, Publishing.user, Steven.wood

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

Image Sources, Licenses and Contributors

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

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 *License*: unknown *Contributors*: Christen.mitchell

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

Image:Role_form.png *Source*: http://wiki.servicenow.com/index.php?title=File:Role_form.png *License*: unknown *Contributors*: Fuji.publishing.user, Publishing.user

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 *License*: unknown *Contributors*: Phillip.salzman

Image:Impersonate first.JPG *Source*: 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 *License*: unknown *Contributors*: Suzanne.smith

Image:skills_user_record_Calgary.png *Source*: http://wiki.servicenow.com/index.php?title=File:Skills_user_record_Calgary.png *License*: unknown *Contributors*: Suzanne.smith

Image:skills_group_record_Calgary.png *Source*: http://wiki.servicenow.com/index.php?title=File:Skills_group_record_Calgary.png *License*: unknown *Contributors*: Suzanne.smith

Image:LocationHierarchy1.png *Source*: http://wiki.servicenow.com/index.php?title=File:LocationHierarchy1.png *License*: unknown *Contributors*: Guy.yedwab

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

Image:acl_workflow.png *Source*: http://wiki.servicenow.com/index.php?title=File:Acl_workflow.png *License*: unknown *Contributors*: Vaughn.romero

Image:acl_embedded_roles.png *Source*: http://wiki.servicenow.com/index.php?title=File:Acl_embedded_roles.png *License*: unknown *Contributors*: Fuji.publishing.user

Image:acl_evaluate_permissions.png *Source*: 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 *License*: unknown *Contributors*: Vaughn.romero

Image:Debug.gifx.gif *Source*: http://wiki.servicenow.com/index.php?title=File:Debug.gifx.gif *License*: unknown *Contributors*: CapaJC, Eric.jacobson, Publishing.user

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

Image:green_checkmark.png *Source*: http://wiki.servicenow.com/index.php?title=File:Green_checkmark.png *License*: unknown *Contributors*: Neil.narvaez

Image:red_x.png *Source*: http://wiki.servicenow.com/index.php?title=File:Red_x.png *License*: unknown *Contributors*: Neil.narvaez

Image:gray_circle.png *Source*: http://wiki.servicenow.com/index.php?title=File:Gray_circle.png *License*: unknown *Contributors*: Neil.narvaez

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

Image:Acl debug list Eureka.png *Source*: http://wiki.servicenow.com/index.php?title=File:Acl_debug_list_Eureka.png *License*: unknown *Contributors*: Neil.narvaez

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

Image:On call menu.jpg *Source*: 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, Joseph.messerschmidt

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

Image:domain_hierarchy.png *Source*: http://wiki.servicenow.com/index.php?title=File:Domain_hierarchy.png *License*: unknown *Contributors*: Vaughn.romero

Image:add_domain.png *Source*: http://wiki.servicenow.com/index.php?title=File:Add_domain.png *License*: unknown *Contributors*: Publishing.user, Vaughn.romero

Image:domain_visibility_02.png *Source*: 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 *License*: unknown *Contributors*: Vaughn.romero

Image:domain_visibility_04.png *Source*: 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 *License*: unknown *Contributors*: Vaughn.romero

Image:delegated_administration_01.png *Source*: 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 *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 *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 *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 *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 *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 *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 *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 *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 *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 *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 *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 *License*: unknown *Contributors*: Vaughn.romero

Image:Domain paths.png *Source*: 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 *License*: unknown *Contributors*: Fuji.publishing.user, Publishing.user, Steven.wood

Image:Role Delegator Change.png *Source*: http://wiki.servicenow.com/index.php?title=File:Role_Delegator_Change.png *License*: unknown *Contributors*: Fuji.publishing.user, Publishing.user, Steven.wood

Image:Role User Record.png *Source*: http://wiki.servicenow.com/index.php?title=File:Role_User_Record.png *License*: unknown *Contributors*: Fuji.publishing.user, Publishing.user, Steven.wood

Image:Role Delegator List.png *Source*: http://wiki.servicenow.com/index.php?title=File:Role_Delegator_List.png *License*: unknown *Contributors*: Fuji.publishing.user, Publishing.user, Steven.wood

Image:Role Delegation.png *Source*: http://wiki.servicenow.com/index.php?title=File:Role_Delegation.png *License*: unknown *Contributors*: Fuji.publishing.user, Publishing.user, Steven.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