Home

This is the home of the Rights DNA 1.0 space.

To help you on your way, we've inserted some of our favourite macros on this home page. As you start creating pages, blogging and commenting you'll see the macros below fill up with all the activity in your space.

Recently Updated

RDNA10
Feb 28, 2017 • attached by Alexandru Geageac

Rights DNA 3.0
Feb 23, 2017 • updated by Confluence Administrator

Uninstall
Feb 23, 2017 • updated by Confluence Administrator • view change

image2017-2-23 13:25:54.png
Feb 23, 2017 • attached by Confluence Administrator

image2017-2-23 13:21:59.png
Feb 23, 2017 • attached by Confluence Administrator

RDNA_UNINSTALL_3.PNG
Feb 23, 2017 • attached by Confluence Administrator

BA_UNINSTALL_2.PNG
Feb 23, 2017 • attached by Confluence Administrator

RDNA_UNINSTALL_1.PNG
Feb 23, 2017 • attached by Confluence Administrator

MANAGE_ADDONS_MENU2.PNG
Feb 23, 2017 • attached by Confluence Administrator

MANAGE_ADDONS_MENU.PNG
Feb 23, 2017 • attached by Confluence Administrator

What should I do if I installed an incompatible version?
Feb 23, 2017 • updated by Confluence Administrator • view change

Install notes for JIRA 7
Feb 23, 2017 • updated by Confluence Administrator • view change

Installation
Feb 23, 2017 • updated by Confluence Administrator • view change

Configuration
Feb 23, 2017 • updated by Confluence Administrator • view change

image2017-2-23 11:50:24.png
Feb 23, 2017 • attached by Confluence Administrator

Navigate space

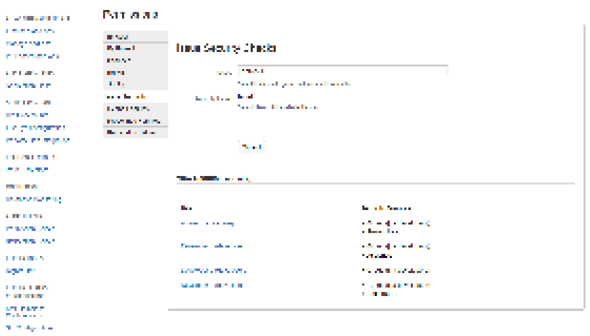
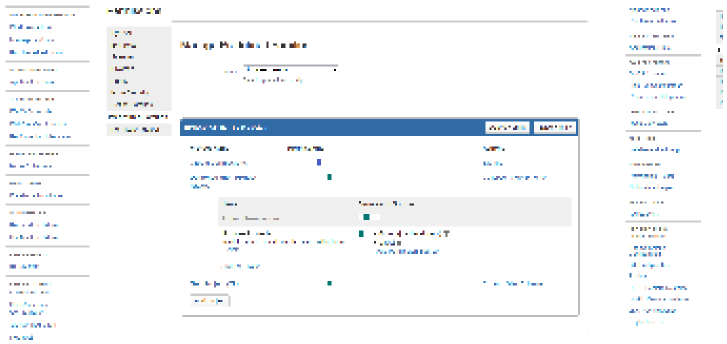
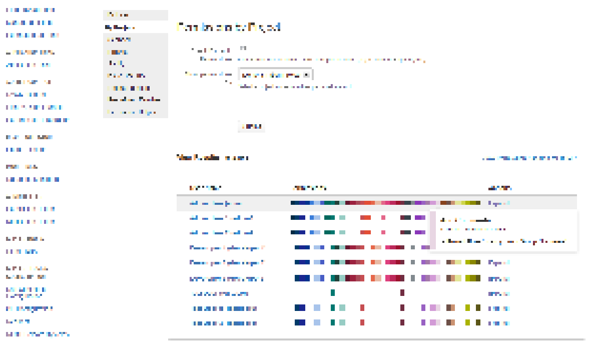
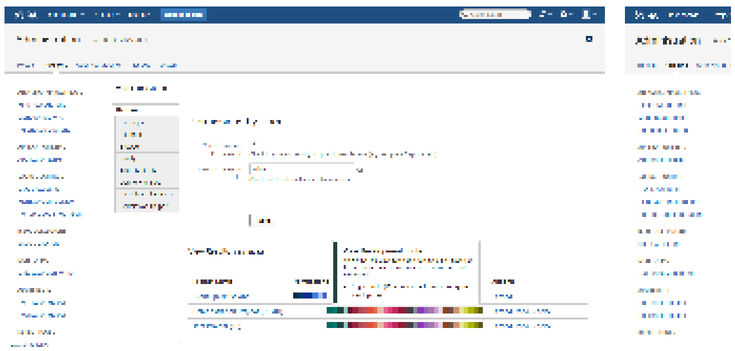
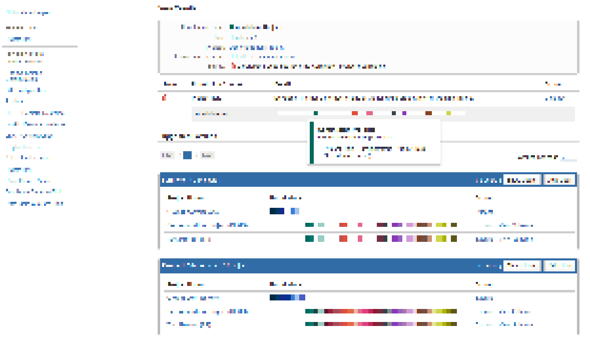
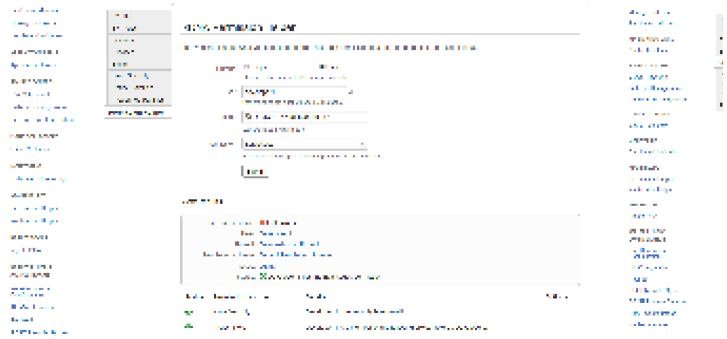
RightsDNA Documentation

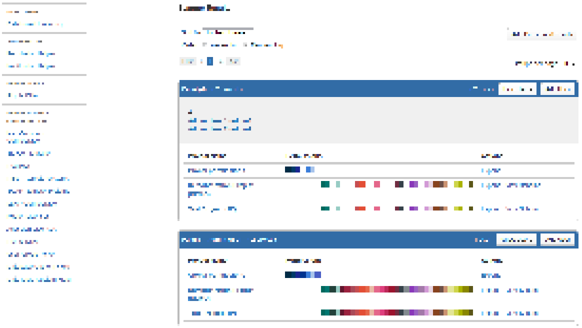
RIGHTS DNA

Safety, clarity and simplicity

- Visualize JIRA effective permissions for each project and user
- Improve your JIRA permission and security settings
- Set permissions for many users to all projects in one step
- Perform sanity checks

Gallery



Features

- Shows the permissions of a single user across the projects.
- Shows who has permissions on a single project.
- Group users with common permission families for better management.
- Shows from where a permission is granted (the sources of the permission).
- Color-coded permissions for faster browsing.

Compatibility with JIRA 6.x and JIRA 7.x.

Installation & Configuration

- Requirements
- Configuration
- Installation
- Install notes for JIRA 7
- What should I do if I installed an incompatible version?
- Licensing
- Uninstall

Requirements

A fully installed RightsDNA consists of two jar files. The simplest way is to use the bundle installer when installing RightsDNA. Please refer to the Install Guide for explanations and details.

Here is the list of available RightsDNA versions:

Version	JIRA	katl-commons
1.0	4.4.x	1.1.10
1.0.1	4.4.x	1.1.11
1.0.2	4.4.x	1.1.12
1.0.3	4.4.x	1.1.13
1.0.4	4.4.x	1.1.15
1.0.5	4.4.x	1.1.17
2.0	5.0.x	2.0.3
2.0.1	5.0.x	2.0.4
2.0.2	5.0.x	2.0.5
2.0.3	5.0.x - 5.1.x	2.0.8
2.0.4	5.0.x - 5.2.x	2.0.10
2.0.5	5.0.x - 5.2.x	2.5
3.0.x	6.x	3.0.x
3.1.x	7.x	3.1.x

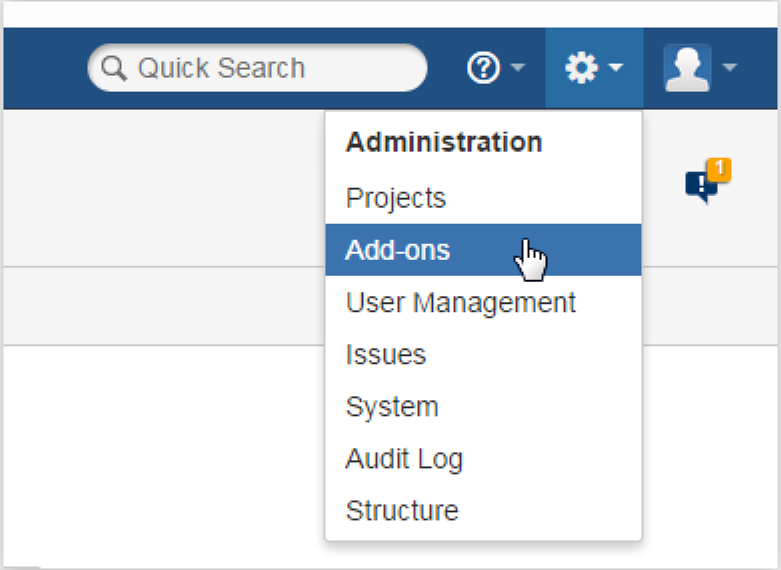
4.0.x	7.x	4.0.x

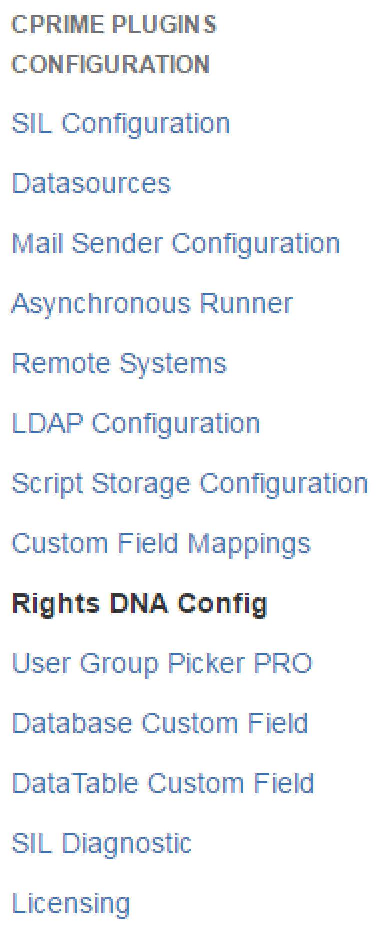
Configuration

Configuration

Configure the **RightsDNA** plugin following the steps below:

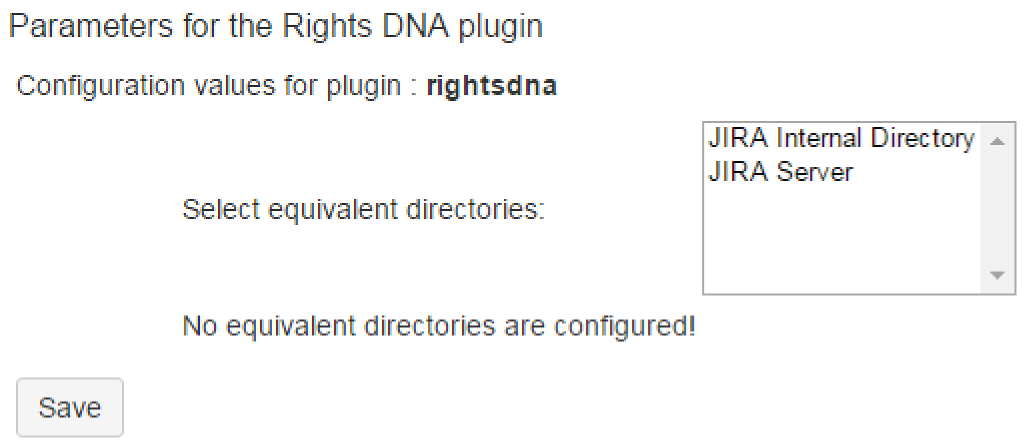
1. Access the **Administration** / **Add-ons** menu, and then select the **Rights DNA Config** link from the **cPrime Plugins Configuration** section on the bottom left of the menu:





2. Configure the parameters for **RightsDNA**:

> Because JIRA allows the same username in different directories, the users could appear twice. A user directory is a place where you store information about users and groups (e.g. LDAP, Active Directory, etc.). Two or more directories are equivalent if they store information about the same users and groups. If you don't have equivalent directories, but you still have duplicate users it means that those users are stored in more than one directory. Users found in multiple directories will be accompanied by a warning image.



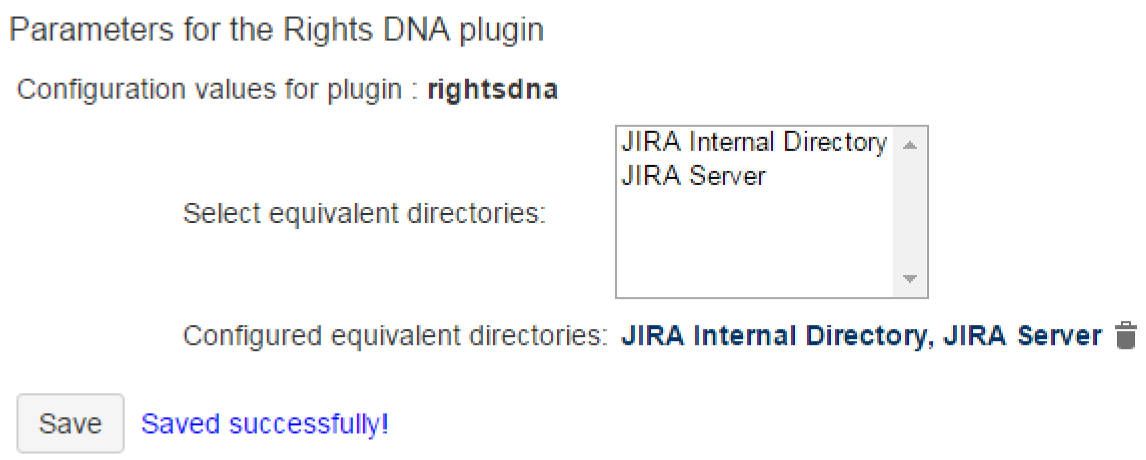
Parameters for the Rights DNA plugin

Configuration values for plugin : **rightsdna**

Select equivalent directories: JIRA Internal Directory, JIRA Server

No equivalent directories are configured!

Here you can configure your equivalent directories to avoid such problems. To do this, select the equivalent directories from the list using **Ctrl+Click**. After selecting the equivalent directories, click **Save.** You can add multiple sets of equivalent directories. Each time the configuration is saved, a new equivalence set is added.



Parameters for the Rights DNA plugin

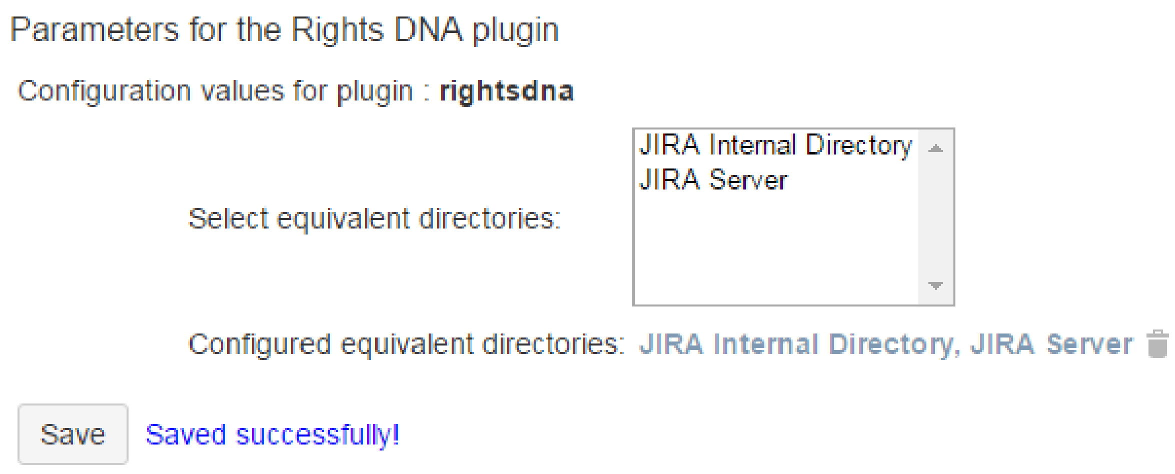
Configuration values for plugin : **rightsdna**

Select equivalent directories:

Configured equivalent directories: **JIRA Internal Directory, JIRA Server**

Saved successfully!

You can also remove a pair of equivalent directories by clicking the trash bin icon. The selected set will be greyed out and deleted once the configuration is saved. You must click **Save** in order for the changes to take place.



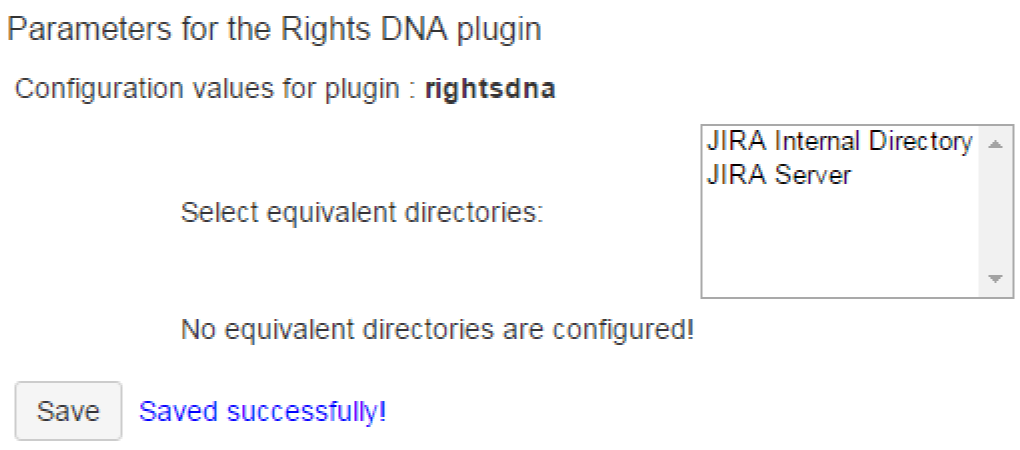
Parameters for the Rights DNA plugin

Configuration values for plugin : **rightsdna**

Select equivalent directories:

Configured equivalent directories: **JIRA Internal Directory, JIRA Server**

Saved successfully!



Parameters for the Rights DNA plugin

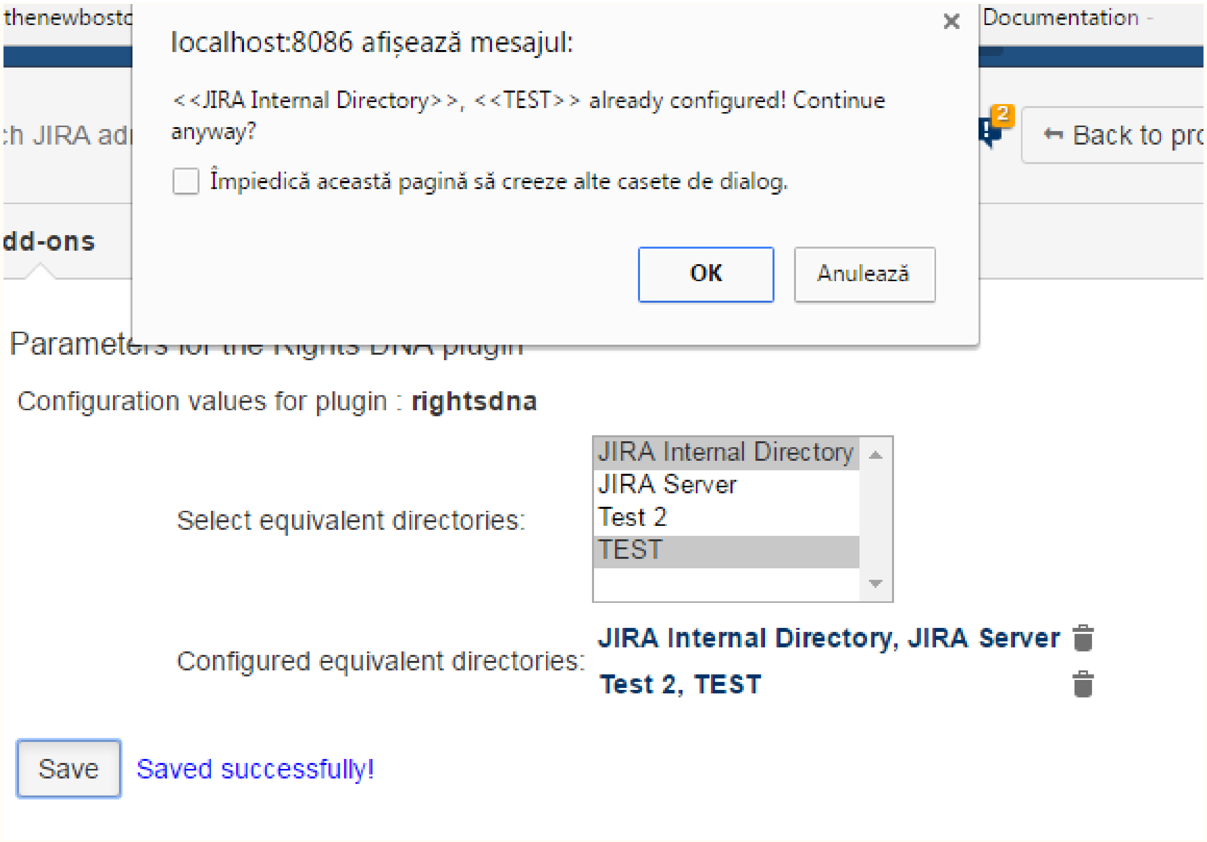
Configuration values for plugin : **rightsdna**

Select equivalent directories:

No equivalent directories are configured!

Saved successfully!

- A directory can't be in more than one pair at a time. If you try to configure a directory that is already configured, a message will be displayed.



localhost:8086 afișează mesajul:

<<JIRA Internal Directory>>, <<TEST>> already configured! Continue anyway?

□ Împiedică această pagină să creeze alte casete de dialog.

Parameters for the Rights DNA plugin

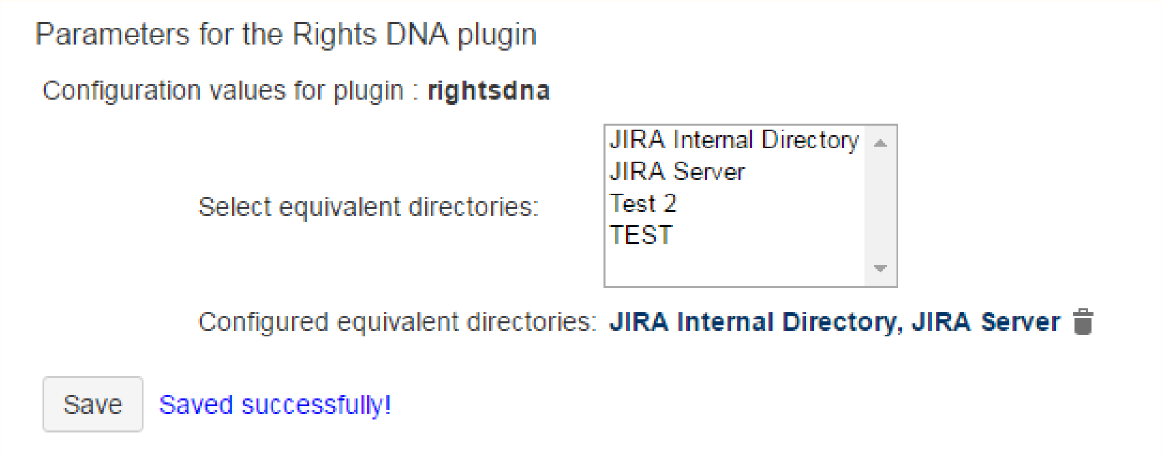
Configuration values for plugin : **rightsdna**

Select equivalent directories: JIRA Internal Directory, JIRA Server, Test 2, TEST

Configured equivalent directories: **JIRA Internal Directory, JIRA Server**
Test 2, TEST

Saved successfully!

If you choose to continue, the sets that contain at least one of the directories from the current selection will be removed and the new configuration will be saved:



Parameters for the Rights DNA plugin

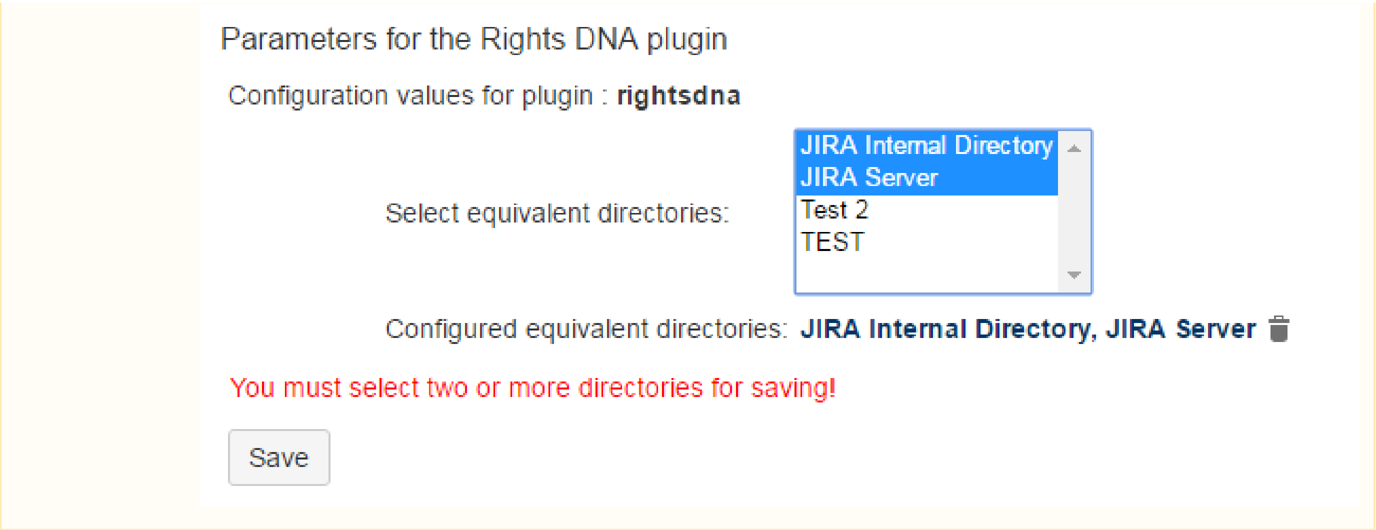
Configuration values for plugin : **rightsdna**

Select equivalent directories: JIRA Internal Directory, JIRA Server, Test 2, TEST

Configured equivalent directories: **JIRA Internal Directory, JIRA Server**

Saved successfully!

- You must select two or more directories for saving!



Parameters for the Rights DNA plugin

Configuration values for plugin : **rightsdna**

Select equivalent directories: JIRA Internal Directory, JIRA Server

Configured equivalent directories: **JIRA Internal Directory, JIRA Server**

You must select two or more directories for saving!

Installation

Installation via Atlassian Universal Plugin Manager

This page points the simple steps to follow for installing the plugin using the Universal Plugin Manager. This method requires an internet connection.

If you are not familiar with Universal Plugin Manager (UPM), please read **this document** before we begin.

Steps are simple:

1. Enter the administration screen and go to ***Administration Add-ons Find new add-ons***.

2. Search for **RightsDNA** plugin and install it

That's all.

Manual Install

It may seem more complicated, but a manual install is quite easy to do. After all, all you have to do is to copy some files. Here's how.

Manual Install

Do not worry, it's a simple task to install it manually:

1. Download the correct rightsdna obr file from Atlassian Marketplace.

2. Go to **Administration Add-ons Manage add-ons**. Install the previously downloaded obr file by using '**Upload add-on**' link.

3. Install a license for rights dna, which can either be provided as the **rightsdna.lic** file, or as the key generated via the Atlassian Marketplace. See more details about this in Licensing.

Install notes for JIRA 7

When upgrading from an older version of JIRA to JIRA 7, you must update all our plugins as well.

As you can see on this page, the versions compatible with JIRA 7 are the 3.1.x versions and the 4.0.x version compatible with the new katl-commons 4.0.x.

What should I do if I installed an incompatible version?

As we have said before, **3.0.x** versions are compatible with **JIRA 6.x** and **3.1.x** and **4.0.x** versions are compatible with **JIRA 7.x**.

If you have installed Rights DNA 3.0.x on JIRA 7.x or Rights DNA 3.1.x / Rights DNA 4.0.x on JIRA 6.x, you should do the next steps :

1. Uninstall warden
2. Uninstall katl-commons
3. Uninstall Rights DNA
4. Install the right version of Rights DNA (the one compatible with your JIRA)
5. katl-commons and warden should now have the right versions as well

> After you uninstall katl-commons and warden, some plugins may remain disabled, so you may need to re-enable them manually.

Licensing

Dual Licensing support

Versions 1.0.1+ and 2.0.1+ support both Kepler and Atlassian licenses, but you only need one valid license to run the plugin, which can is provided as the **rightsdna.lic** file, or as the key generated via the **Atlassian Marketplace**.

The order in which the licenses are checked is:

1. Atlassian License
2. Kepler License

It is **strongly recommended** that you do not install both licenses at once, as this might yield unwanted results. For example, consider that you have an Atlassian License with the support date expired and one valid Kepler License. In this case you cannot update the plugin, because the Atlassian License is checked first and its support date is expired.

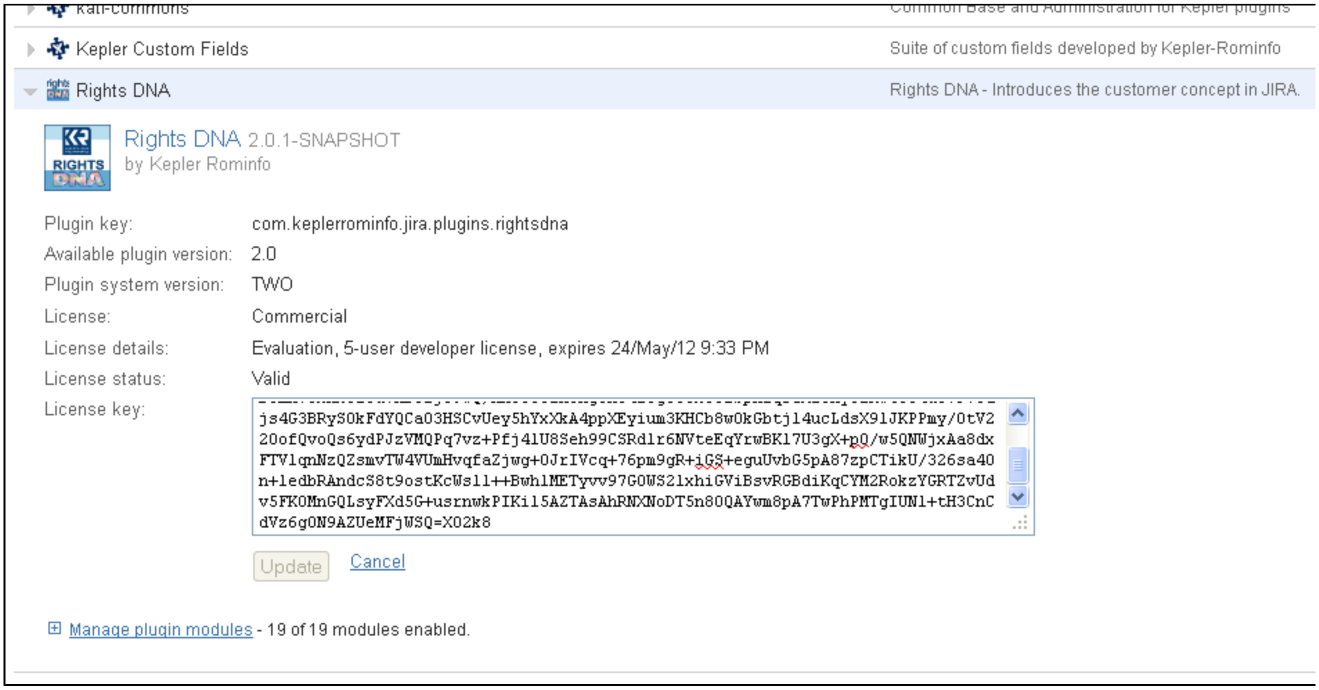
Atlassian Licensing

> To support Atlassian licenses you need to install **katl-commons 2.0.4+** (for JIRA 5) or **katl-commons 1.1.11+** (for JIRA 4.4.x) *before* installing Rights DNA.

The Atlassian Marketplace allows you to easily purchase or generate an evaluation license for Rights DNA.

Using Universal Plugin Manager 2.0.1+

After generating the license key, all you have to do is access the **Administration-> Plugins** section in your JIRA instance and paste the key into the corresponding plugin textbox.



katl-commons
Common Base and Administration for Kepler plugins
Kepler Custom Fields
Suite of custom fields developed by Kepler-Rominfo
Rights DNA
Rights DNA - Introduces the customer concept in JIRA.
RIGHTS DNA

Rights DNA 2.0.1-SNAPSHOT
by Kepler Rominfo

Plugin key:	com.keplerrominfo.jira.plugins.rightsdna
Available plugin version:	2.0
Plugin system version:	TWO
License:	Commercial
License details:	Evaluation, 5-user developer license, expires 24/May/12 9:33 PM
License status:	Valid
License key:	

```
js4G3BRyS0kFdYQCa03HSCvUey5hYxXkA4ppXEyium3KHCb8w0kGbtjl4ucLdsX9lJKPPmy/0tV2
20ofQvoQs6ydPJzVMQPq7vz+Pfj4lU8Seh99CSRdlr6NVteEqYrwBKl7U3gX+pQ/w5QNWjxAa8dx
FTVlqnNzQZsmvTW4VUmHvqfaZjwg+0JrIVcq+76pm9gR+iGS+eguUvbG5pA87zpCTikU/326sa40
n+ledbRAndcS8t9ostKcWsll++BwhlMETyvv97G0WS2lxhiGViBsvRGBdiKqCYM2RokzYGRTZvUd
v5FK0MnGQLsyFXd5G+usrnwkPIKil5AZTAsAhRNXNoDT5n80QAYwm8pA7TwPhPMTgIUNl+tH3CnC
dVz6g0N9AZUeMFjWSQ=X02k8
```

⊞ Manage plugin modules - 19 of 19 modules enabled.

Kepler Licensing

The Kepler license is a file (**rightsdna.lic**) which must be placed in the <JIRA_HOME>/kepler folder. You can either generate and download a free evaluation license by registering on **our site** and accessing the **Licenses** section, or you can purchase the plugin by following **these instructions**.

Reminder
Don't forget that you need only *one* valid license to run the plugin.

Technical info
Starting with **katl-commons version 2.5.5** an new plugin, called ***Warden***, will be automatically installed by any paid add-on (including Rights DNA). This plugin is responsible with the management of licenses (both JIRA and Kepler). Do not attempt to uninstall it without removing first all the Kepler paid add-ons.

Removing an unused license
If you want to remove a no longer used Atlassian license, this can be done in UPM (for UPM 2.0.1+) , by removing the old license key and clicking Update. To remove a Kepler license, you have to delete the correspondent .lic file from the kepler folder. Note that any change to the Kepler license requires a server restart.

Uninstall

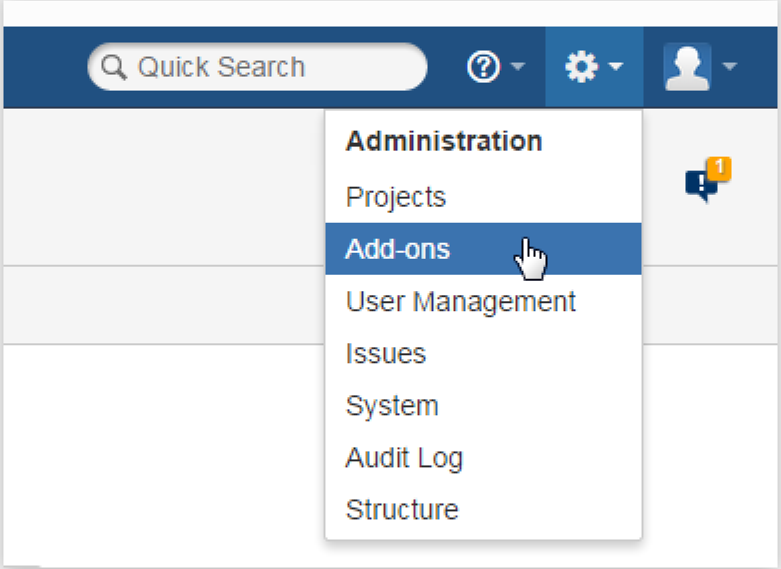
Uninstall via Atlassian Universal Plugin Manager

At first we will uninstall the plugin and finally we'll remove the corresponding tables in the internal database.

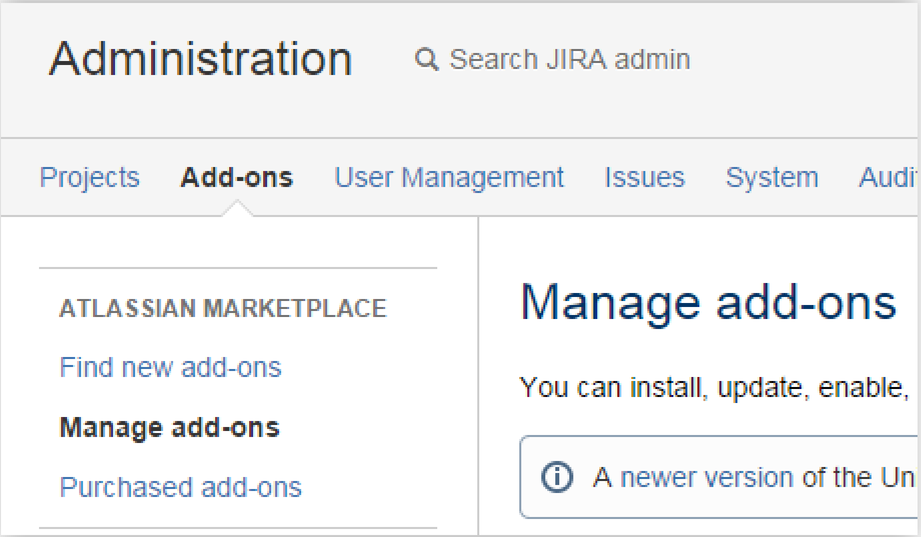
Uninstall the plugin

If you are not familiar with Universal Plugin Manager (UPM), please read this document before we begin.

1) Log in as administrator and go to Jira Administration in the right up section of the page



2) Click on the `Add-ons` menu link ant then the Manage Add-ons section

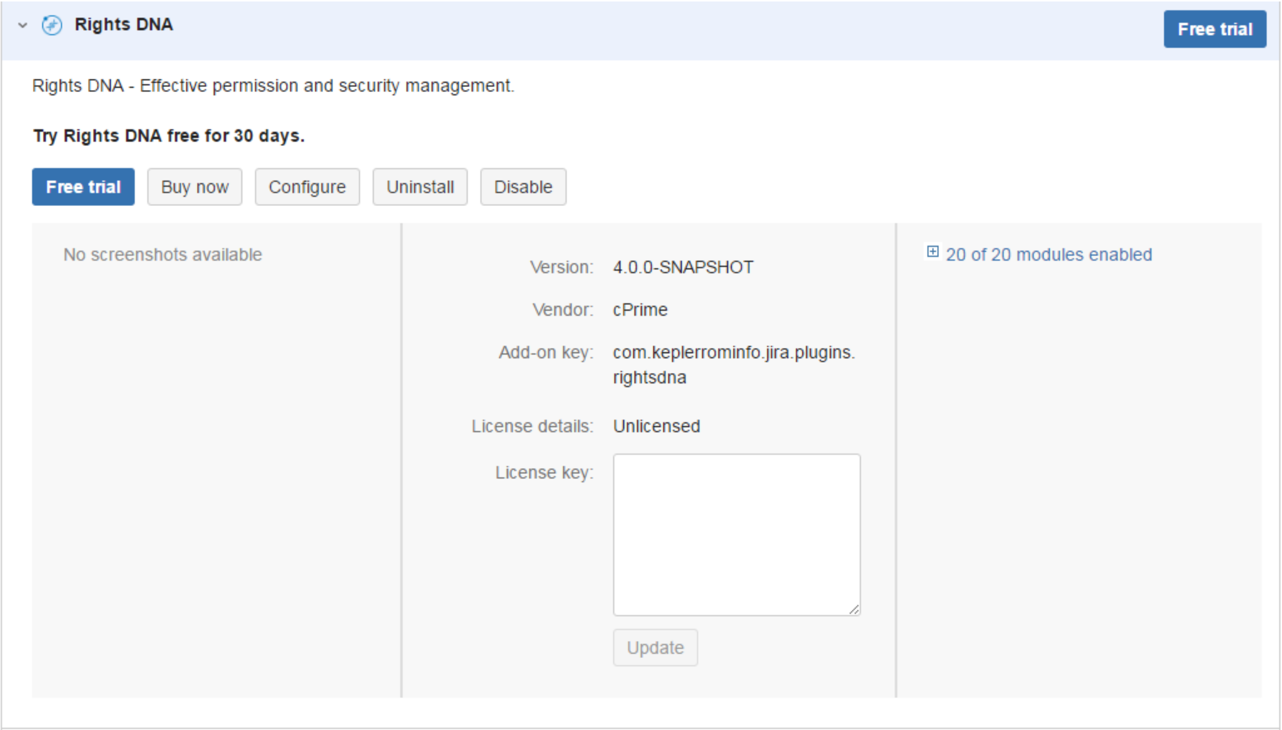


Manage add-ons

You can install, update, enable,

A newer version of the Un

3) Search for `Rights DNA` plugin in `User Installed Plugins` section and click on `Uninstall` button



Rights DNA - Effective permission and security management.

Try Rights DNA free for 30 days.

No screenshots available

Version: 4.0.0-SNAPSHOT

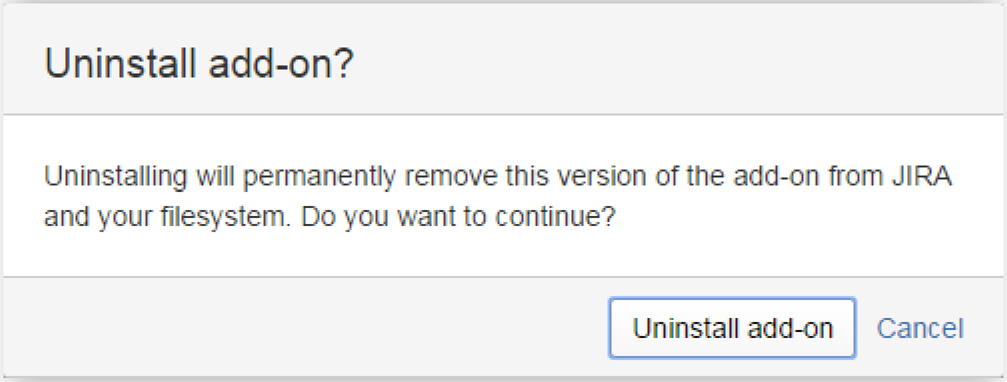
Vendor: cPrime

Add-on key: com.keplerrominfo.jira.plugins.rightsdna

License details: Unlicensed

License key:

4) Press `Continue` when the uninstall confirmation dialog box appears

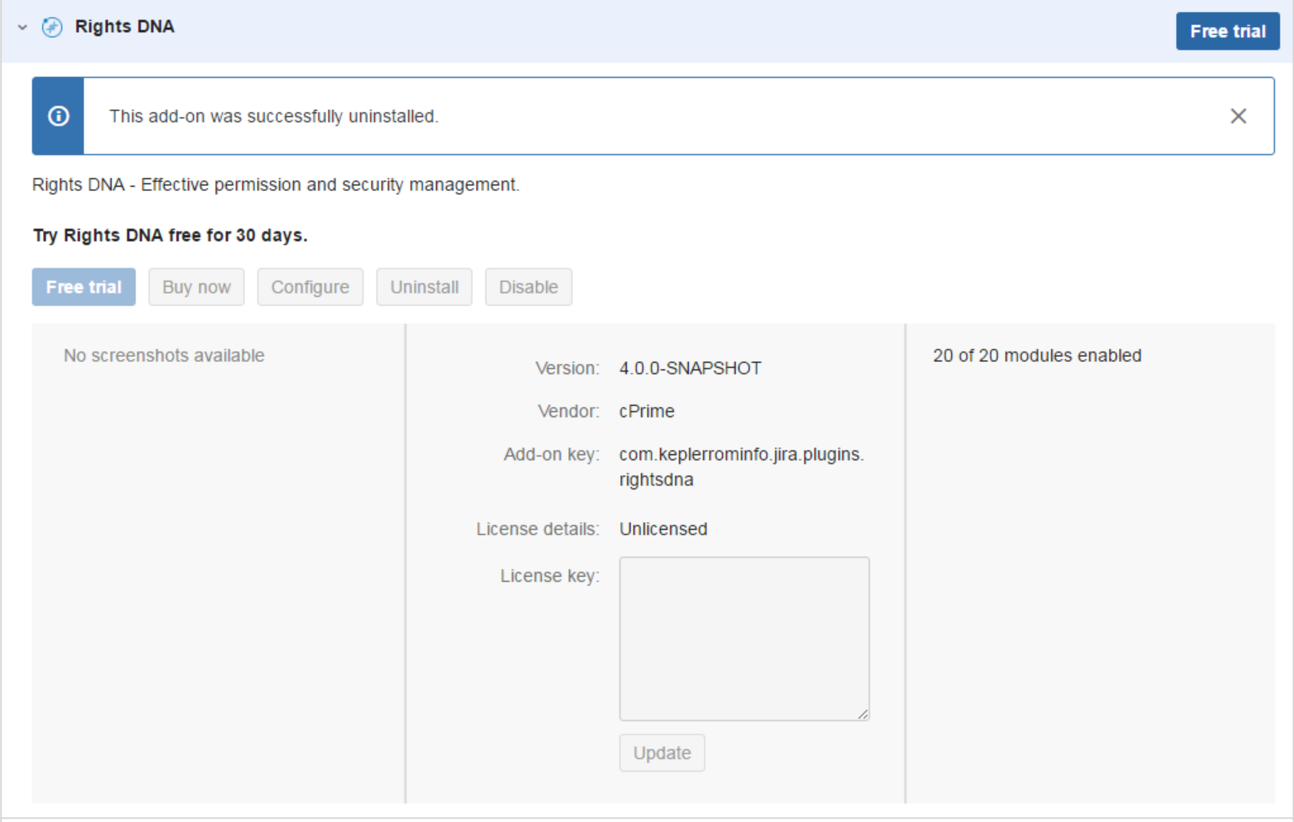


Uninstall add-on?

Uninstalling will permanently remove this version of the add-on from JIRA and your filesystem. Do you want to continue?

Optionally, you can provide feedback on the reason why you uninstalled the plugin.

5) A message "successfully uninstalled" should appear



This add-on was successfully uninstalled.

Rights DNA - Effective permission and security management.

Try Rights DNA free for 30 days.

No screenshots available

Version: 4.0.0-SNAPSHOT

Vendor: cPrime

Add-on key: com.keplerrominfo.jira.plugins.rightsdna

License details: Unlicensed

License key:

20 of 20 modules enabled

Now you can delete Rights DNA corresponding plugin tables.

Remove the tables

You can go to the internal database administration tool.

You can use a visual tool or a command line tool and remove the following tables in your database:

- krightsdnacustomer
- krightsdnapermset
- krightsdnapermsetusers
- krightsdnauserdefinedfamily

Manual Uninstall

At first we will uninstall the plugin manually and finally we'll remove the corresponding tables in the internal database.

Uninstall the plugin

You need to have access where the Jira server has been installed.

Goto the folder where Jira server has been installed.

Access **<JIRA_APPLICATION_DATA>/plugins/installed-plugins** and manually delete Rights DNA plugin.

Remove the tables

You can go to the internal database administration tool.

You can use a visual tool or a command line tool and remove the following tables in your database:

- krightsdnacustomer
- krightsdnapermset
- krightsdnapermsetusers
- krightsdnauserdefinedfamily

Restart the server

Now you can restart Jira server

User Guide

Years of careless maintenance can create a lot of external and internal users with strange permissions on some projects. In order to sort it out, we created RightsDNA to colorfully show our administrators where the problems are.

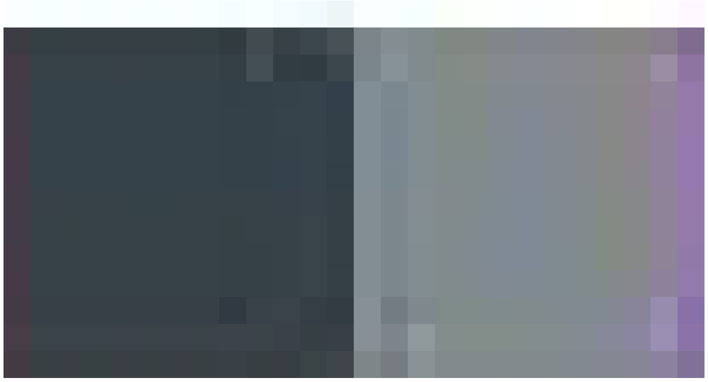
The plugin contains a suite of useful tools to help you maintain a clean and secure JIRA environment. Rights DNA focuses on calculating the effective rights and shows from where a permission is granted (the sources of the permission).

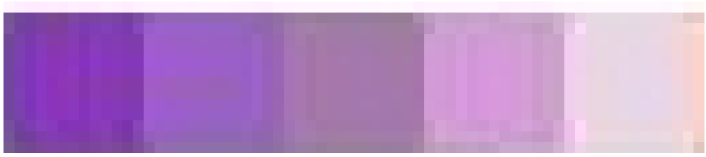
DNA ?

The permission (or rights) DNA of a user is represented as a colored bar.

This encapsulates all the permissions a user can have. Each permission is color-coded and is represented by a square on the bar. If the placeholder for a certain permission is white, it means that the user does not have that permission.

Different shades of the same color are used to represent permissions in a certain category.

Category	DNA segment
Global Permissions	
Project Permissions	
Issue Permissions	
Voters & Watchers Permissions	

Comments Permissions	
Attachments Permissions	
Time Tracking Permissions	

The permissions contained in each category are as follows:

Category	Permissions	Description	Color
Global Permissions	**JIRA Administrators**	Ability to perform most administration functions (excluding Import & Export, SMTP Configuration, etc.).	🟦
	Bulk Change	Ability to modify a collection of issues at once. For example, resolve multiple issues in one step.	🟦
	Create Shared Objects	Ability to share dashboards and filters with other users, groups and roles.	🟦
	Manage Group Filter Subscriptions	Ability to manage (create and delete) group filter subscriptions.	🟦
	JIRA System Administrators	Ability to perform all administration functions. There must be at least one group with this permission.	🟦
	JIRA Users	Ability to log in to JIRA. They are a 'user'. Any new users created will automatically join these groups, unless those groups have JIRA System Administrators or JIRA Administrators permissions.	🟦
	Browse Users	Ability to select a user or group from a popup window as well as the ability to use the 'share' issues feature. Users with this permission will also be able to see names of all users and groups in the system.	🟦
Project Permissions	**Administer Projects**	Ability to administer a project in JIRA.	🟩
	Browse Projects	Ability to browse projects and the issues within them.	🟩
	View Version Control	Ability to view Version Control commit information for issues.	🟩
	View Read-Only Workflow	Users with this permission may view a read-only version of a workflow.	🟩
Issue Permissions	**Assign Issues**	Ability to assign issues to other people.	🟥
	Assignable User	Users with this permission may be assigned to issues.	🟥
	Close Issues	Ability to close issues. Often useful where your developers resolve issues, and a QA department closes them.	🟥
	Create Issues	Ability to create issues.	🟥
	Delete Issues	Ability to delete issues.	🟥
	Edit Issues	Ability to edit issues.	🟧

	Link Issues	Ability to link issues together and create linked issues. Only useful if issue linking is turned on.	🟧
	Modify Reporter	Ability to modify the reporter when creating or editing an issue.	🟥
	Move Issues	Ability to move issues between projects or between workflows of the same project (if applicable). Note the user can only move issues to a project he or she has the create permission for.	🟥
	Resolve Issues	Ability to resolve and reopen issues. This includes the ability to set a fix version.	🟥
	Schedule Issues	Ability to set or edit an issue's due date.	🟥
	Set Issue Security	Ability to set the level of security on an issue so that only people in that security level can see the issue.	🟫
Voters & Watchers Permissions	**Manage Watchers**	Ability to manage the watchers of an issue.	⬛
	View Voters and Watchers	Ability to view the voters and watchers of an issue	⬛
Comments Permissions	**Delete All Comments**	Ability to delete all comments made on issues.	🟪
	Delete Own Comments	Ability to delete own comments made on issues.	🟪
	Edit All Comments	Ability to edit all comments made on issues.	🟪
	Edit Own Comments	Ability to edit own comments made on issues.	🟪
	Add Comments	Ability to comment on issues.	⬜
Attachments Permissions	**Delete All Attachments**	Users with this permission may delete all attachments.	🟫
	Delete Own Attachments	Users with this permission may delete own attachments.	🟫
	Create Attachments	Users with this permission may create attachments.	🟫
Time Tracking Permissions	**Work On Issues**	Ability to log work done against an issue. Only useful if Time Tracking is turned on.	🟨
	Delete All Worklogs	Ability to delete all worklogs made on issues.	🟨
	Delete Own Worklogs	Ability to delete own worklogs made on issues.	🟨
	Edit All Worklogs	Ability to edit all worklogs made on issues.	🟩
	Edit Own Worklogs	Ability to edit own worklogs made on issues.	🟩

Permission Sources

Each user can have a permission for a variety of reasons. He can have a permission because he is in a certain group or project role or because he is the Assignee on an issue, for example. We call these permission sources.

Each permission can have any (or none) of the permission sources listed below:

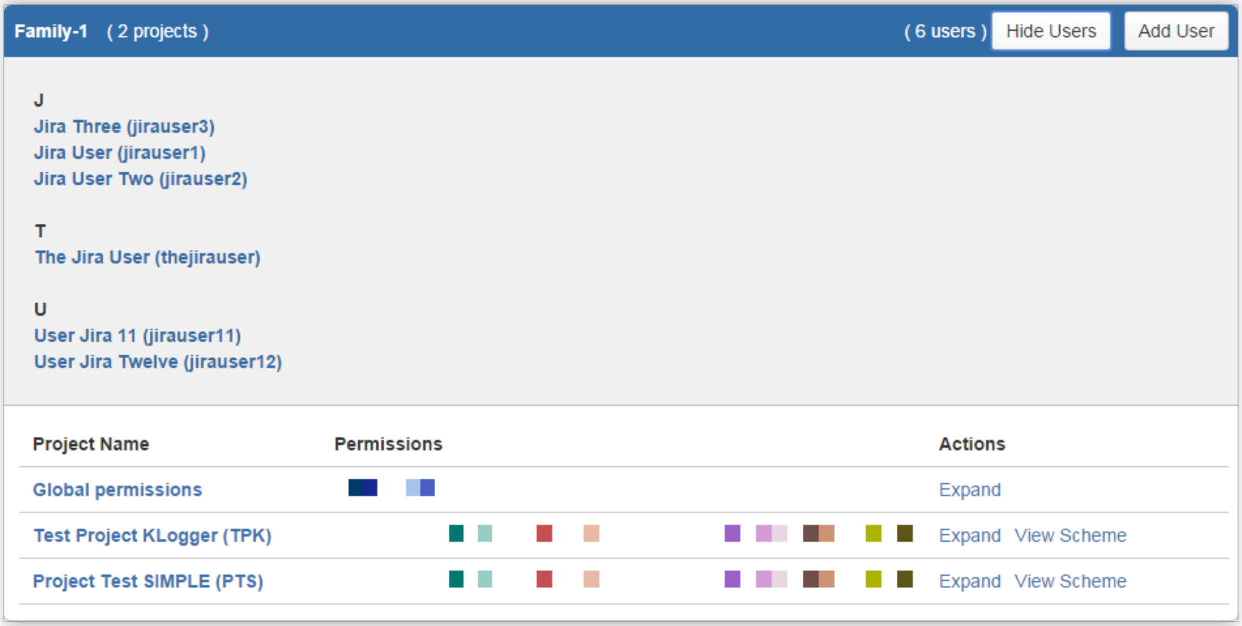
Permission Source	Variety	Description
Direct	None	The user is assigned the permission directly.
Group	Direct	One of the groups the user is member of is given the permission.
	Anyone	This is a pseudo-group which gives all users (registered or anonymous) a certain permission
Project Role	Direct	The user is assigned directly in the project role that has the permission
	From Group	The user belongs to one of the groups assigned to the project role that has the permission

Assignee	None	The user has the permission because he **CAN BE** the assignee on an issue. This does not necessarily mean that he does have the permission at the moment.
Reporter	None	The user has the permission because he **CAN BE** the reporter of an issue. This does not necessarily mean that he does have the permission at the moment.
Project Lead	None	The user has the permission because he is the project lead.

> **Assignee**, **Reporter** and **Project Lead** permission sources are not verified by default. We called these **dynamic permissions.** If you wish to include these in the analysis check the **Check Dynamic Permissions** check-box when scanning.

Families

Families are sets of users grouped by common DNA. This means that the users in a family may have different permissions for different projects, but for each project they all have the same permissions with identical permission sources.



Family-1 (2 projects) (6 users) Hide Users Add User

J
Jira Three (jirauser3)
Jira User (jirauser1)
Jira User Two (jirauser2)

T
The Jira User (thejirauser)

U
User Jira 11 (jirauser11)
User Jira Twelve (jirauser12)

Project Name	Permissions	Actions
Global permissions	■■ ■■	Expand
Test Project KLogger (TPK)	■ ■ ■ ■ ■ ■■ ■■ ■ ■	Expand View Scheme
Project Test SIMPLE (PTS)	■ ■ ■ ■ ■ ■■ ■■ ■ ■	Expand View Scheme

> For more information about families, see **Managing Families**.

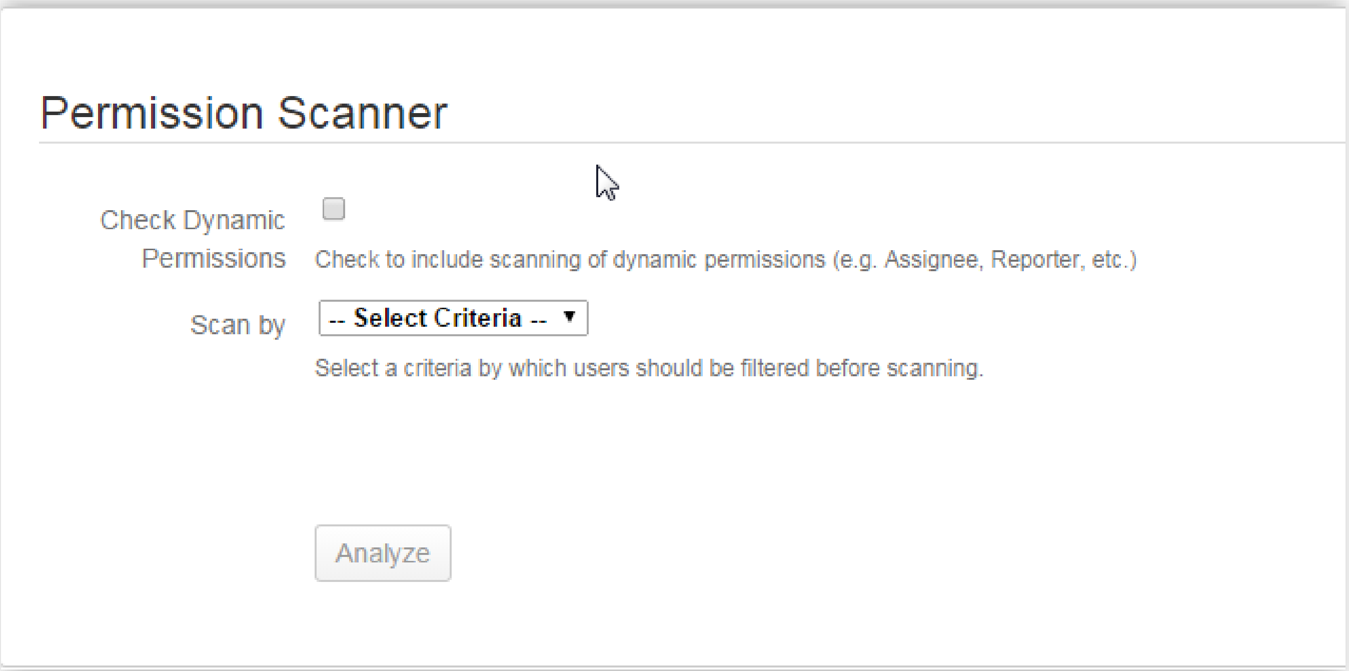
Families

- Family Scanner
- Family Browser
- Managing Families
- Define Families
- Predefined Families

Family Scanner

Scanning

Allows you to filter the JIRA users by a certain criteria (see below) and then group them by common permission DNA for better management.



Permission Scanner

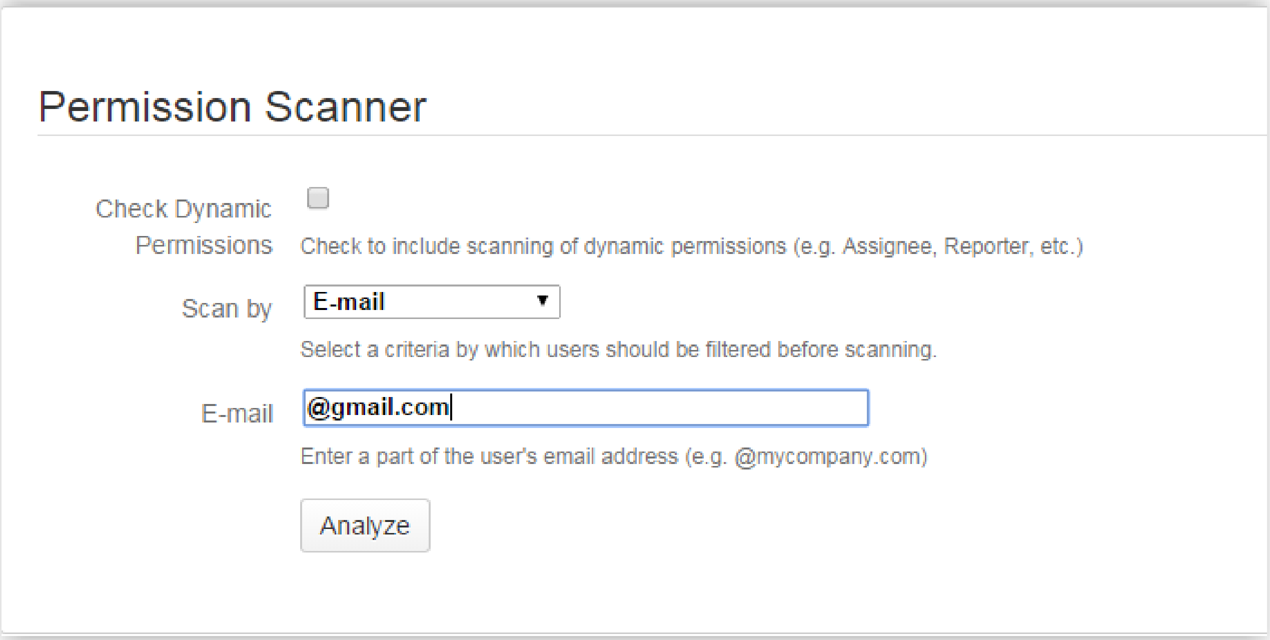
Check Dynamic Permissions □
Check to include scanning of dynamic permissions (e.g. Assignee, Reporter, etc.)

Scan by: -- Select Criteria -- ▾
Select a criteria by which users should be filtered before scanning.

Calculates effective rights for a set of users grouped by a certain criteria.

There are two criteria you can filter users by:

- **e-mail** - analyze effective rights starting from a set of users with common e-mail keywords.



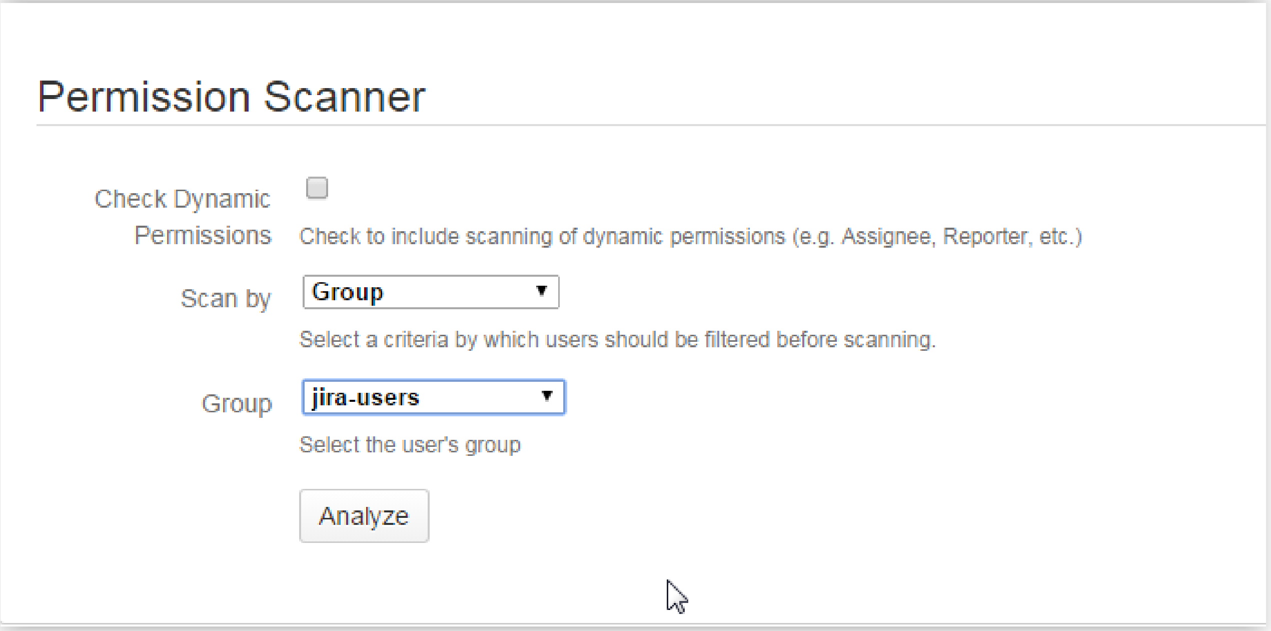
Permission Scanner

Check Dynamic Permissions □
Check to include scanning of dynamic permissions (e.g. Assignee, Reporter, etc.)

Scan by: E-mail ▾
Select a criteria by which users should be filtered before scanning.

E-mail: @gmail.com
Enter a part of the user's email address (e.g. @mycompany.com)

- **group** - analyze effective rights starting from a set of users who are members of the selected group.



Permission Scanner

Check Dynamic Permissions □ Check to include scanning of dynamic permissions (e.g. Assignee, Reporter, etc.)

Scan by: Group ▾ Select a criteria by which users should be filtered before scanning.

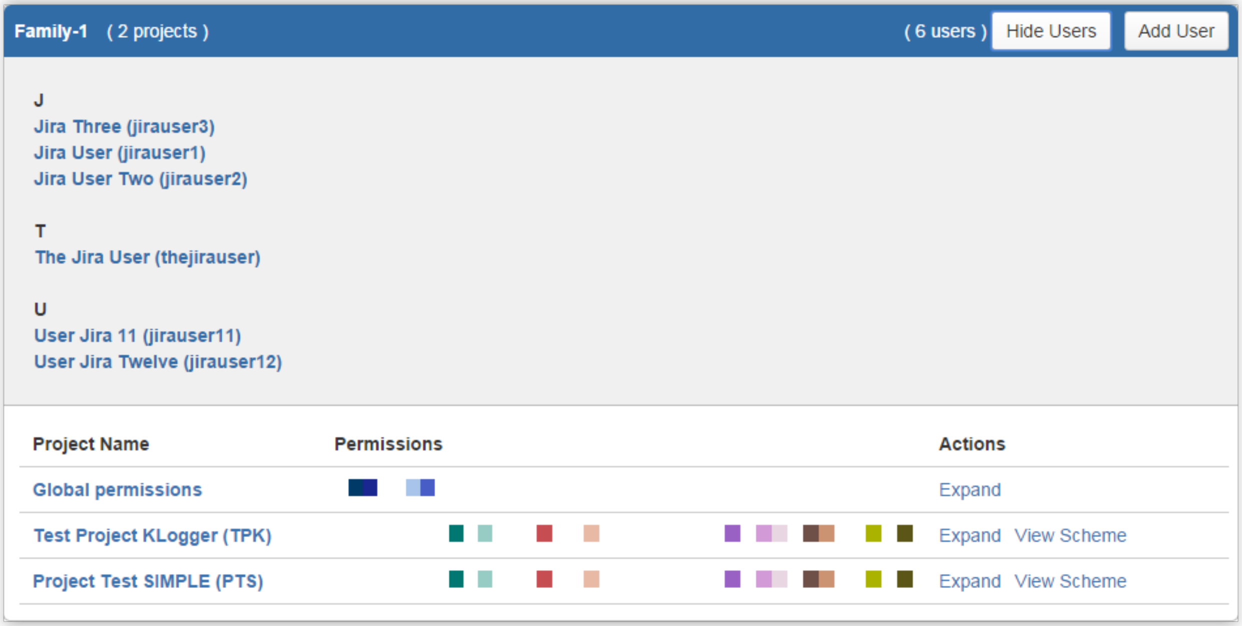
Group: jira-users ▾ Select the user's group

As always, you have the option to include or not the **dynamic permissions** in the scan.

Results

The result is a set of families.

Families are sets of users grouped by common DNA. This means that the users in a family may have different permissions for different projects, but for each project they all have the same permissions with identical permission sources.



Family-1 (2 projects) (6 users)

J
Jira Three (jirauser3)
Jira User (jirauser1)
Jira User Two (jirauser2)

T
The Jira User (thejirauser)

U
User Jira 11 (jirauser11)
User Jira Twelve (jirauser12)

Project Name	Permissions	Actions
Global permissions	■■ ■■	Expand
Test Project KLogger (TPK)	■ ■ ■ ■ ■ ■■ ■■ ■ ■	Expand View Scheme
Project Test SIMPLE (PTS)	■ ■ ■ ■ ■ ■■ ■■ ■ ■	Expand View Scheme

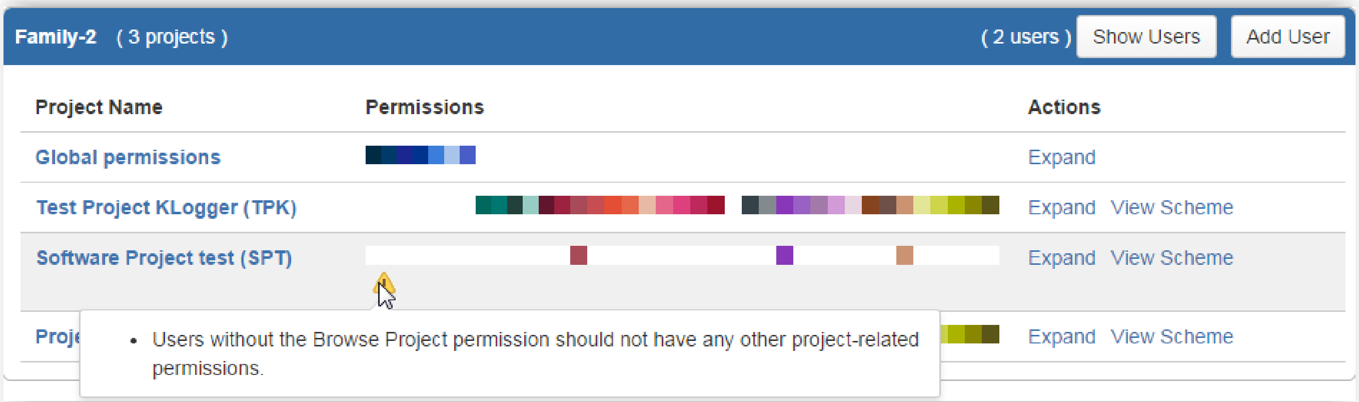
For more information about families, see **Managing Families**.

When moving a user into a family, the permissions of that users will change immediately.

Advices

Besides calculating effective rights for a set of users grouped by a certain criteria, this tab highlights some situations based on the DNA of each family. We call them advices.

Are displayed as a warning icon at the end of the project row they applied to or at the end of the global permissions row. To see the effective advice you have to hover over the warning icon.



Family-2 (3 projects) (2 users) Show Users Add User

Project Name	Permissions	Actions
Global permissions		Expand
Test Project KLogger (TPK)		Expand View Scheme
Software Project test (SPT)		Expand View Scheme
Proj…		Expand View Scheme

- Users without the Browse Project permission should not have any other project-related permissions.

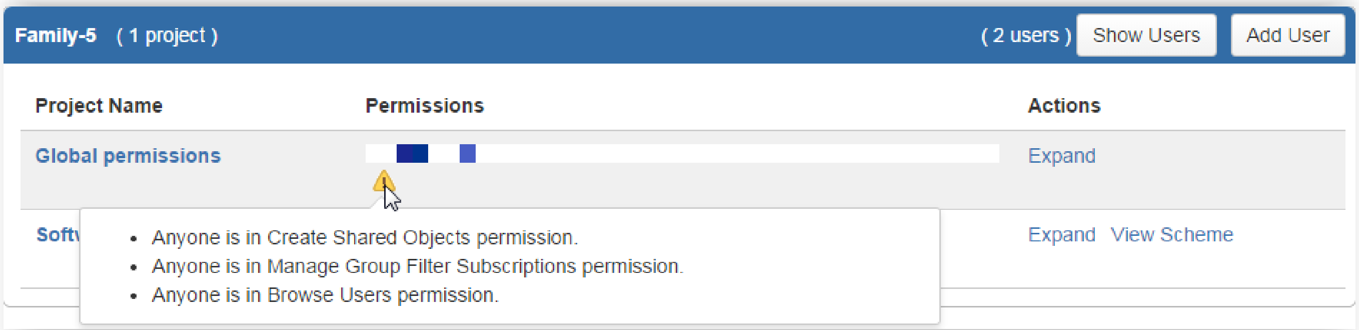
This helps you to see useless permissions within projects. For example, it warns you when:

-users don't have JIRA Users global permission, but they have some other permissions.

-users have some project-related permission but they don't have permission to browse that project.

-users can create issues on the project but they can't see them (they don't have the Browse Projects permission).

When JIRA is private , it warns you when group "Anyone" is in Global Permission. This means that even if you don't want public users to login, Anyone can take advantage of global permissions.



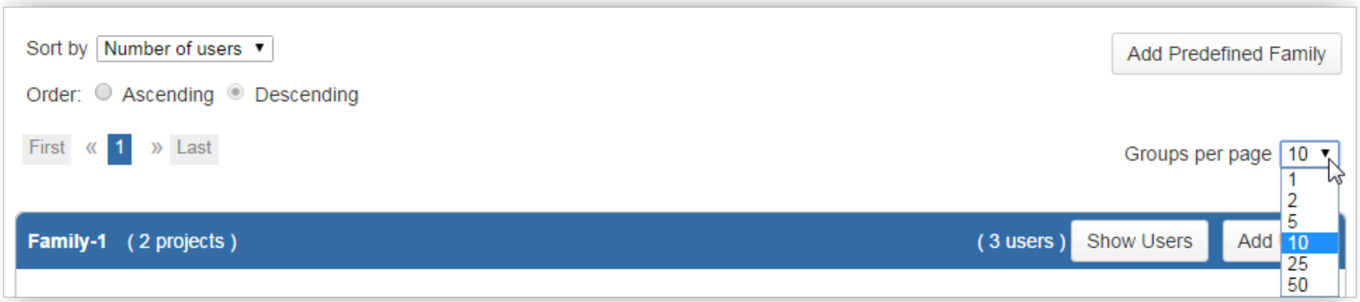
Family-5 (1 project) (2 users) Show Users Add User

Project Name	Permissions	Actions
Global permissions		Expand
Soft…		Expand View Scheme

- Anyone is in Create Shared Objects permission.
- Anyone is in Manage Group Filter Subscriptions permission.
- Anyone is in Browse Users permission.

The number of families per page is configurable. By default, a single family per page is displayed.

> Since rights-DNA 2.0.3, the default number of families displayed per page is 10.



Sort by Number of users ▼

Order: ○ Ascending ● Descending

Add Predefined Family

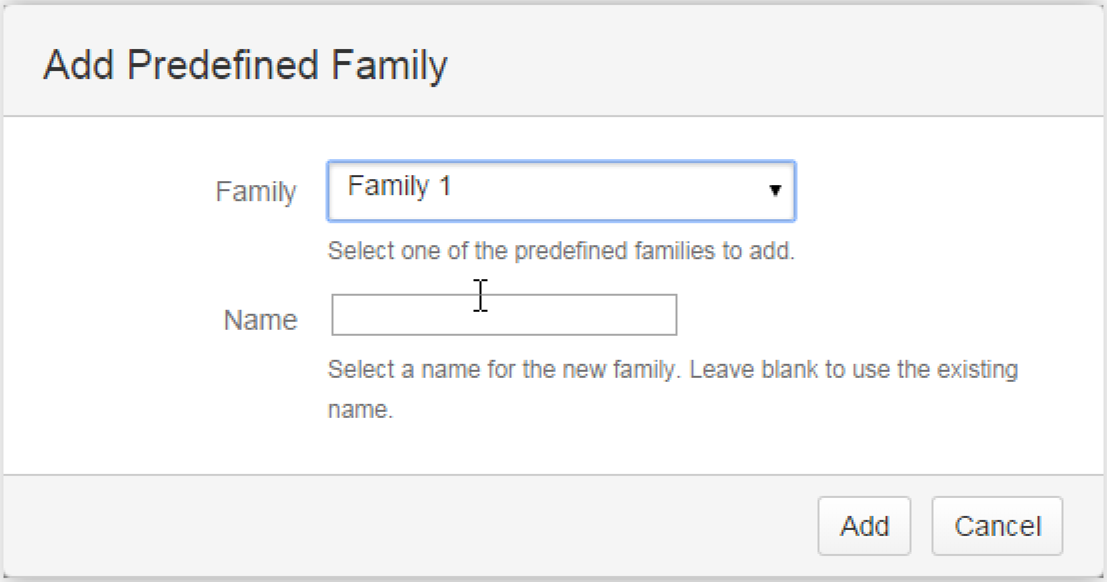
First « 1 » Last

Groups per page 10 ▼

Family-1 (2 projects) (3 users) Show Users Add

Adding Predefined Families

To add a predefined family to the family set you have to click the **Add Predefined Family** button in the upper-right corner of the family set. A dialog will be displayed allowing you to choose a family defined in the **Define Families** tab.



Add Predefined Family

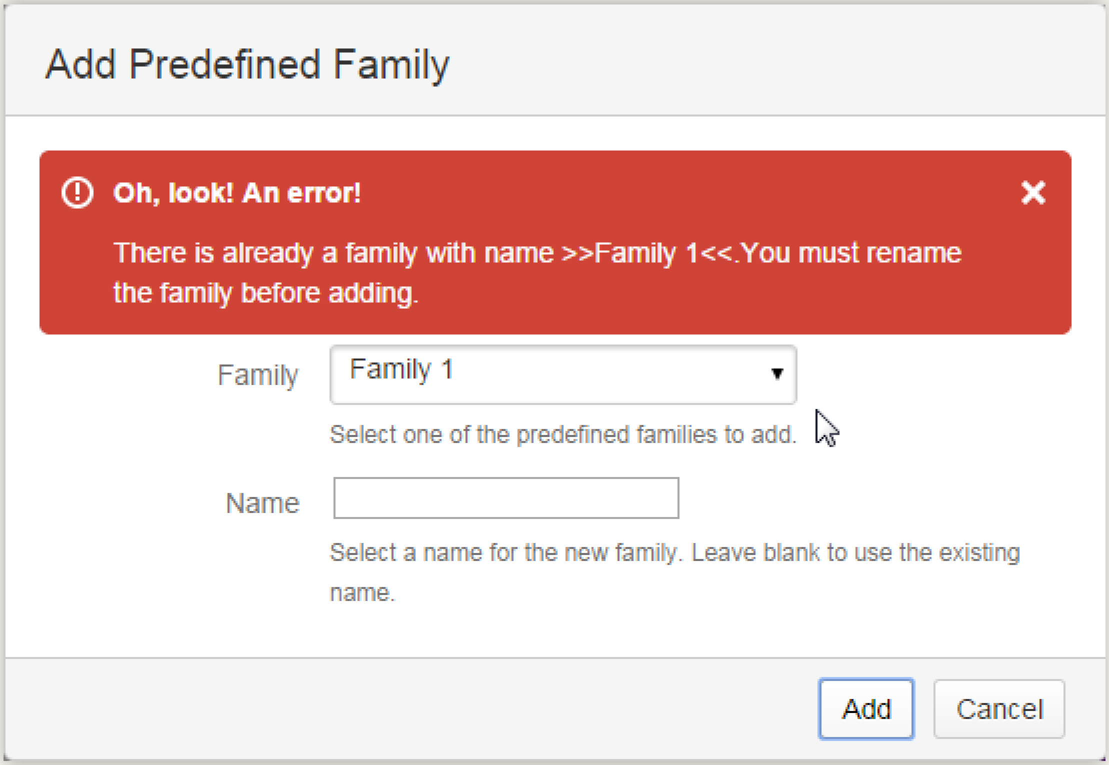
Family: Family 1

Select one of the predefined families to add.

Name:

Select a name for the new family. Leave blank to use the existing name.

You can change the name for the family before adding it. Bear in mind that you cannot have two families with the same name for the same scanning criteria.



Add Predefined Family

Oh, look! An error!

There is already a family with name >>Family 1<<.You must rename the family before adding.

Family: Family 1

Select one of the predefined families to add.

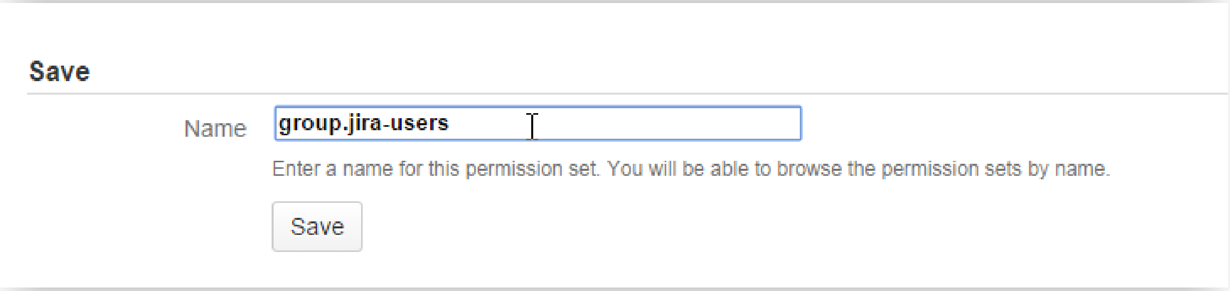
Name:

Select a name for the new family. Leave blank to use the existing name.

For more information about defining families, see **Define Families.**

Saving

You can save your families by clicking the **Save** button at the bottom of the page. In the **Family Browser** tab you will be able to browse through the saved sets of families by name.

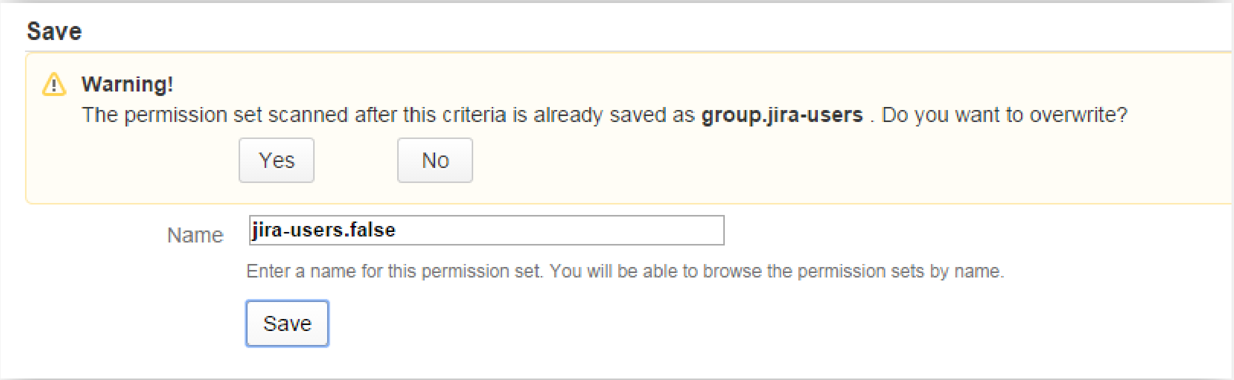


Save

Name: group.jira-users

Enter a name for this permission set. You will be able to browse the permission sets by name.

You must enter a name for the permission set before saving.



Save

Warning!
The permission set scanned after this criteria is already saved as **group.jira-users** . Do you want to overwrite?

Name: jira-users.false

Enter a name for this permission set. You will be able to browse the permission sets by name.

You can not save two sets of families scanned by the same criteria. The results will be overwritten.

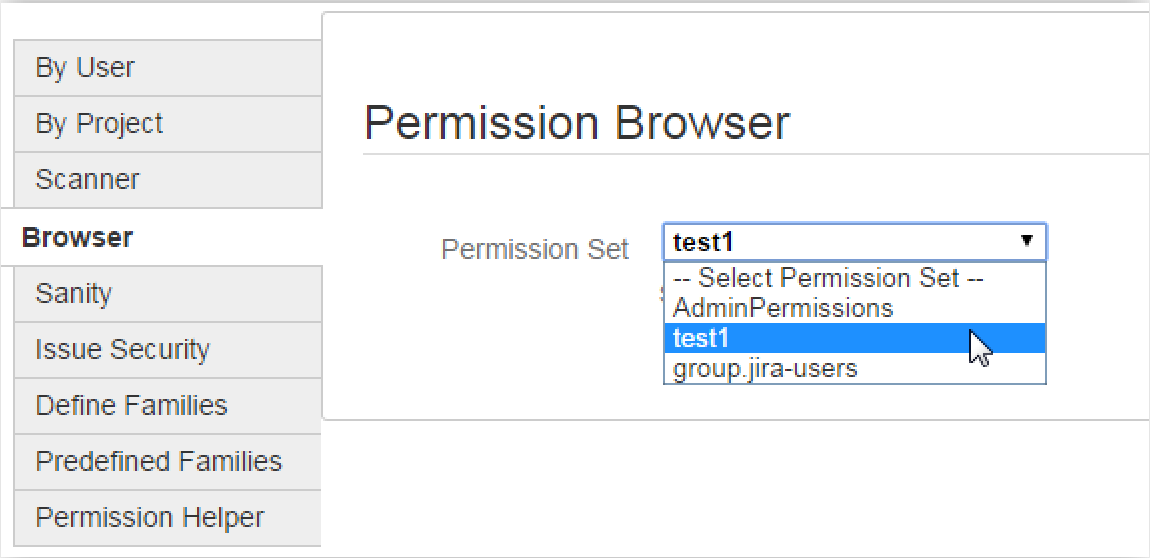
See Also:

Error formatting macro: contentbylabel: com.atlassian.confluence.api.service.exceptions.BadRequestException: Could not parse cql : null

Family Browser

Browser

The current tab displays all the sets of families saved using the **Family Scanner** tab.

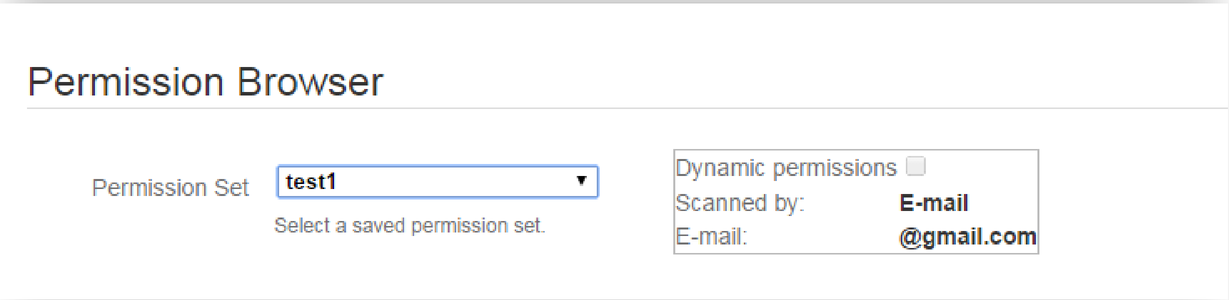


Permission Browser

Permission Set: test1

Results

After selecting a permission set, the scanning criteria and the families will be displayed.



Permission Browser

Permission Set: test1

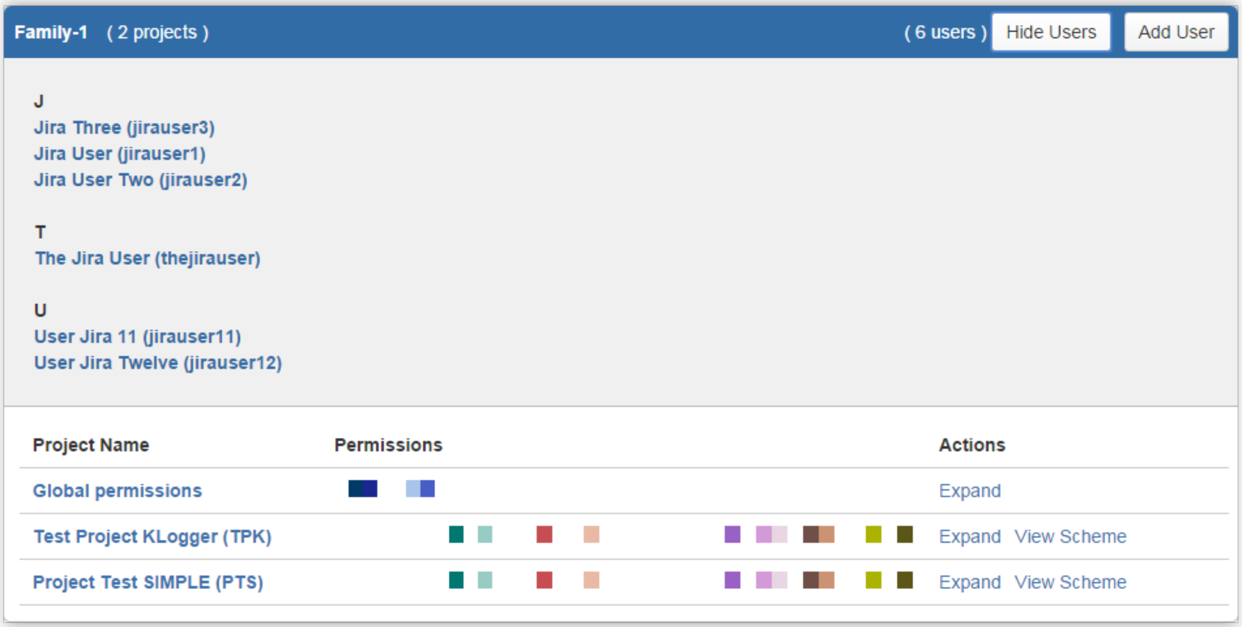
Select a saved permission set.

Dynamic permissions ☐

Scanned by: **E-mail**

E-mail: **@gmail.com**

Families are sets of users grouped by common DNA. This means that the users in a family may have different permissions for different projects, but for each project they all have the same permissions with identical permission sources.



Family-1 (2 projects) (6 users) Hide Users Add User

J
Jira Three (jirauser3)
Jira User (jirauser1)
Jira User Two (jirauser2)

T
The Jira User (thejirauser)

U
User Jira 11 (jirauser11)
User Jira Twelve (jirauser12)

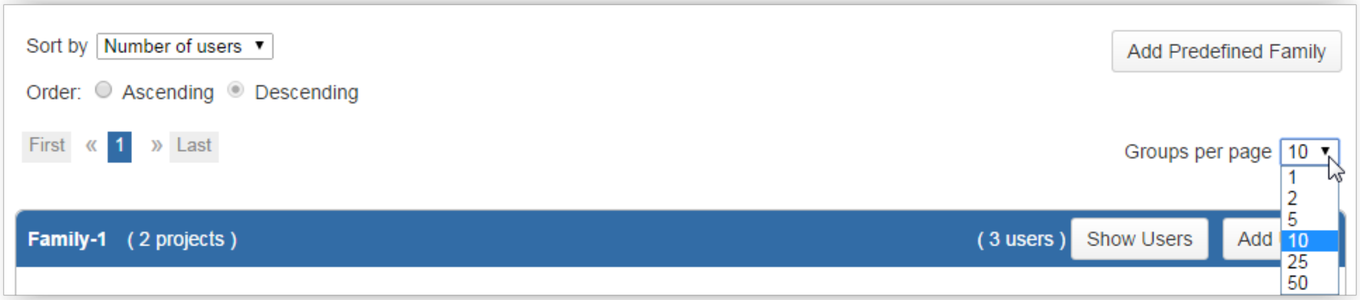
Project Name	Permissions	Actions
Global permissions	■■ ■■	Expand
Test Project KLogger (TPK)	■ ■ ■ ■ ■ ■■ ■■ ■ ■	Expand View Scheme
Project Test SIMPLE (PTS)	■ ■ ■ ■ ■ ■■ ■■ ■ ■	Expand View Scheme

For more information about families, see **Managing Families**.

All changes will be immediately saved.

The number of families per page is configurable. By default, a single family per page is displayed.

Since rights-DNA 2.0.3, the default number of families displayed per page is 10.



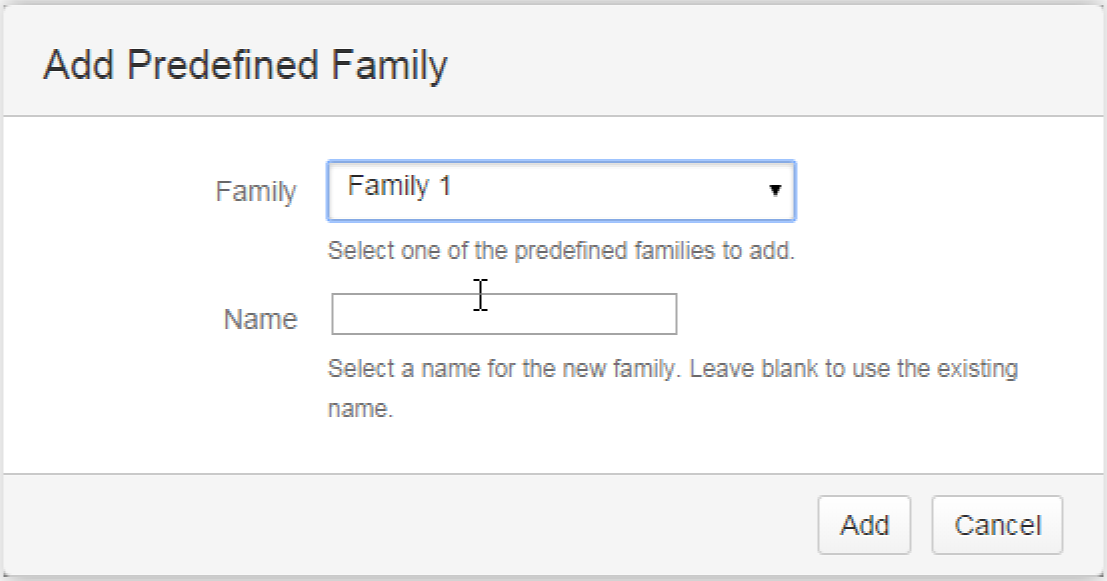
Sort by Number of users ▾

Order: ○ Ascending ● Descending

Family-1 (2 projects) (3 users) Show Users Add

Adding Predefined Families

To add a predefined family to the family set you have to click the **Add Predefined Family** button in the upper-right corner of the family set. A dialog will be displayed allowing you to choose a family defined in the **Define Families** tab.



Add Predefined Family

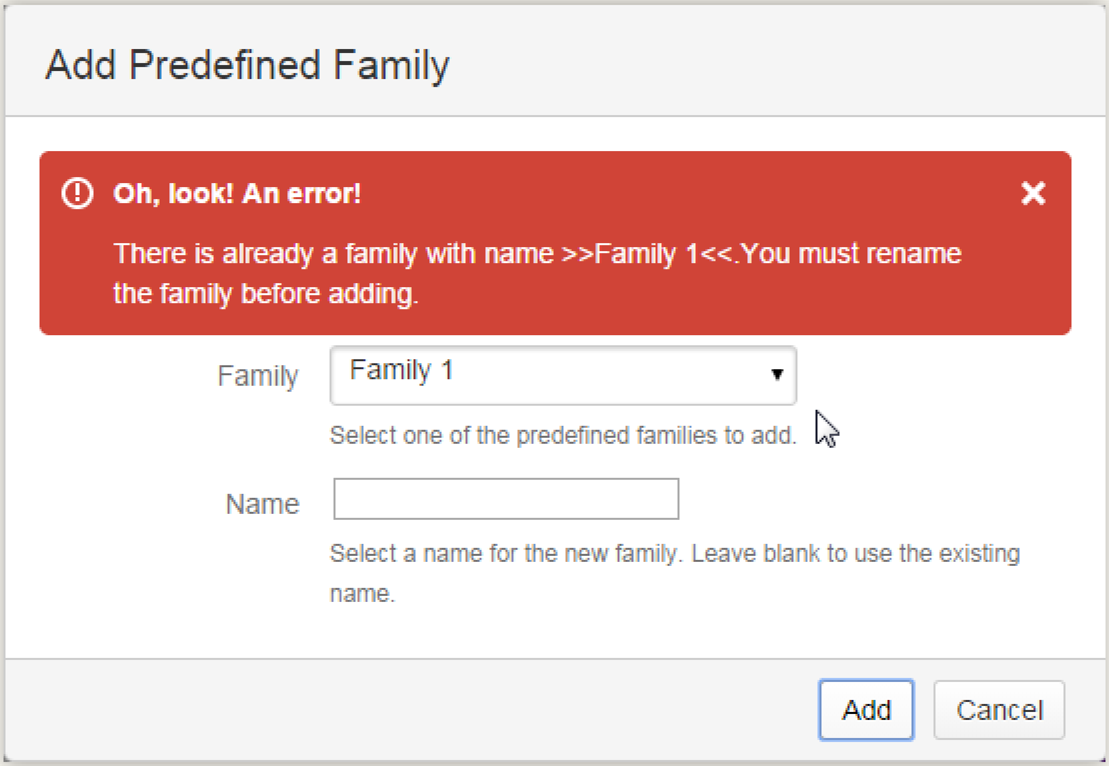
Family: Family 1

Select one of the predefined families to add.

Name:

Select a name for the new family. Leave blank to use the existing name.

You can change the name for the family before adding it. Bear in mind that you cannot have two families with the same name for the same scanning criteria.



Add Predefined Family

Oh, look! An error!

There is already a family with name >>Family 1<<.You must rename the family before adding.

Family: Family 1

Select one of the predefined families to add.

Name:

Select a name for the new family. Leave blank to use the existing name.

For more information about defining families, see **Define Families.**

When browsing, there is no guarantee that user's permissions have not changed in the meantime. For example, if you do a scan by email containing **tom.tom**, then save the family and change his permissions after saving, this won't be reflected in the already saved family.

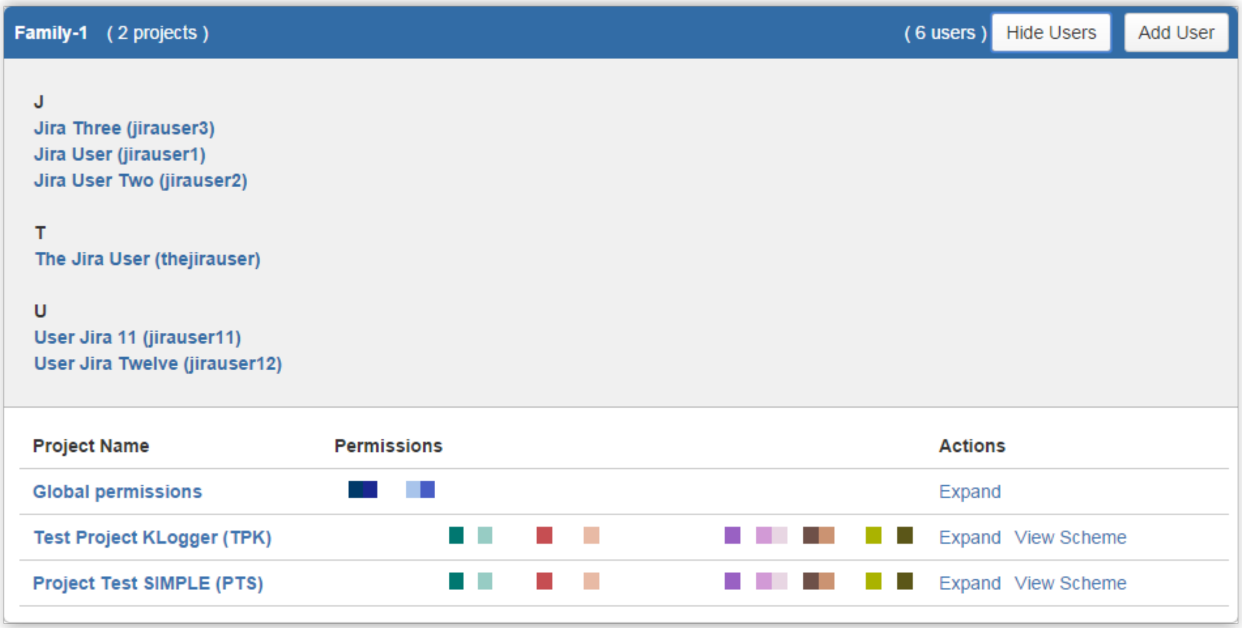
See Also:

Error formatting macro: contentbylabel: com.atlassian.confluence.api.service.exceptions.BadRequestException: Could not parse cql : null

Managing Families

Families

Families are sets of users grouped by common DNA. This means that the users in a family may have different permissions for different projects, but for each project they all have the same permissions with identical permission sources.



Family-1 (2 projects) (6 users)

J
Jira Three (jirauser3)
Jira User (jirauser1)
Jira User Two (jirauser2)

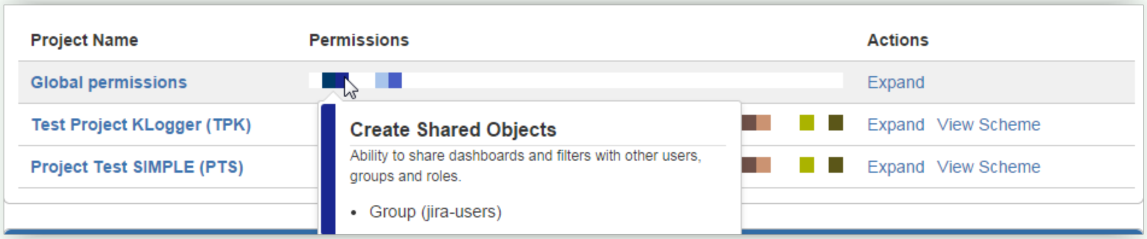
T
The Jira User (thejirauser)

U
User Jira 11 (jirauser11)
User Jira Twelve (jirauser12)

Project Name	Permissions	Actions
Global permissions	■■ ■■	Expand
Test Project KLogger (TPK)	■ ■ ■ ■ ■ ■■ ■■ ■ ■	Expand View Scheme
Project Test SIMPLE (PTS)	■ ■ ■ ■ ■ ■■ ■■ ■ ■	Expand View Scheme

As always, you can **expand** and **collapse** permission details per projects for easy access.

It is also possible to see the permission sources by hovering over the permission color you are interested in.



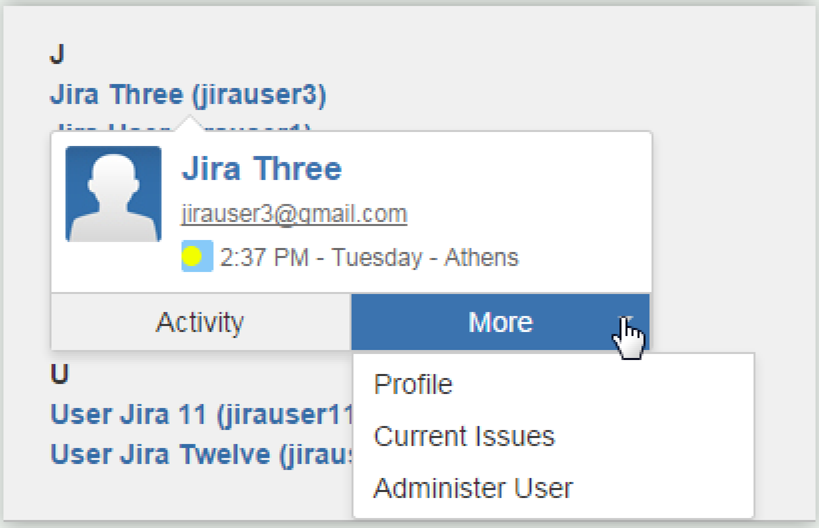
Project Name	Permissions	Actions
Global permissions	■■ ■■	Expand
Test Project KLogger (TPK)	■■ ■ ■	Expand View Scheme
Project Test SIMPLE (PTS)	■■ ■ ■	Expand View Scheme

Create Shared Objects

Ability to share dashboards and filters with other users, groups and roles.

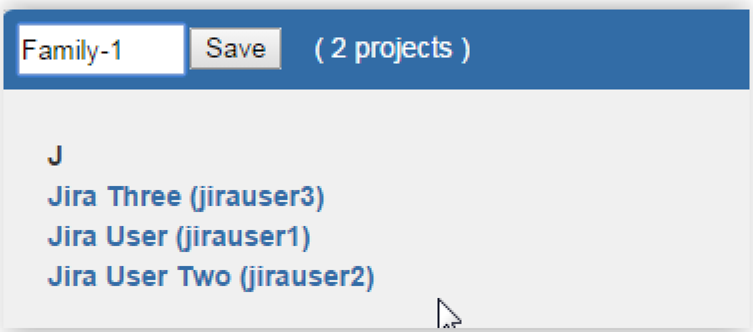
- Group (jira-users)

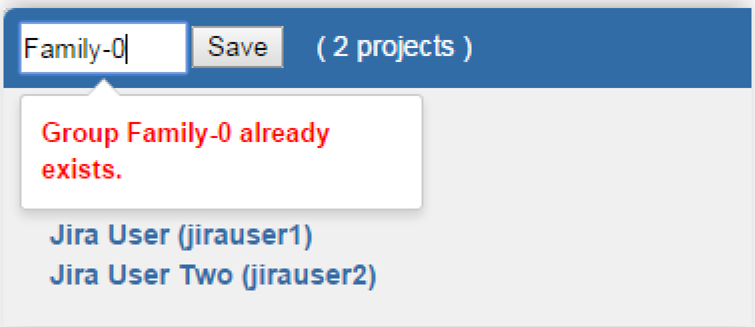
If you hover over the user names, a menu will appear with detailed information about the user and other useful links.



Renaming a Family

To rename a family, all you need to do is click its name. The name will change to a text box allowing you to change it.


Family-1

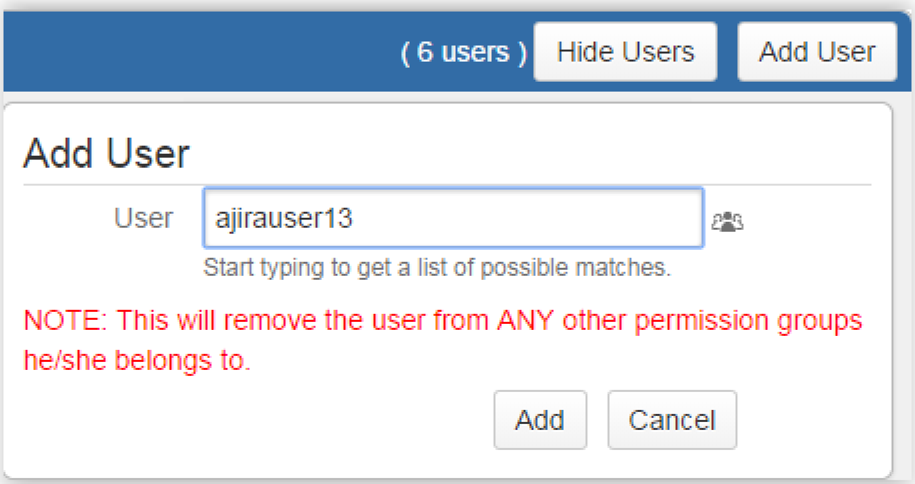

Family-0

Group Family-0 already exists.

> You cannot have two families with the same name for the same scanning criteria.

Adding a User

You can also move users from one family to another or even add users who were not covered by the scan. To do this, click the **Add User** button in the top-right corner of the family panel. A box will appear prompting for the username you wish to add to the family.



Add User

User ajirauser13

Start typing to get a list of possible matches.

NOTE: This will remove the user from ANY other permission groups he/she belongs to.

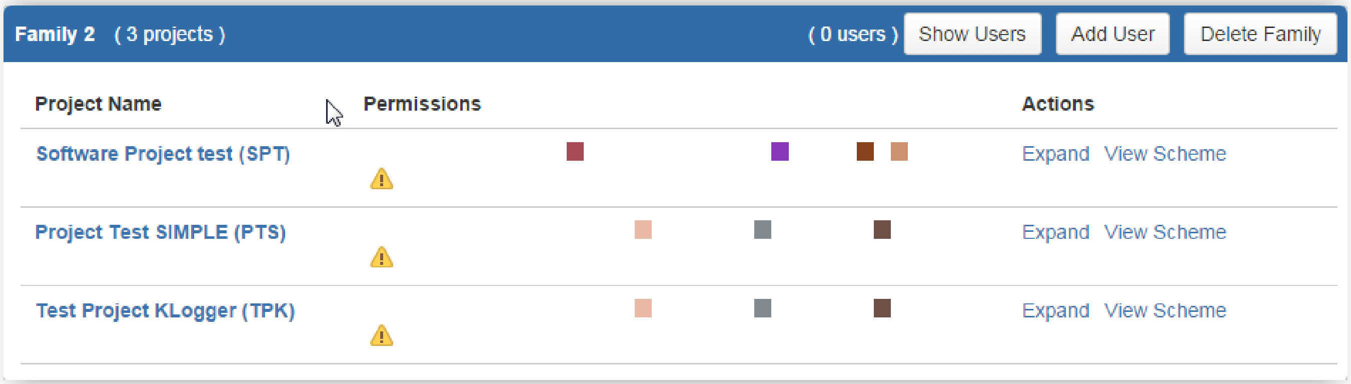
If the user is already in a family of the same scan, he will be removed from that family and added to this one.

When you choose to move a user into a family, you have to be aware that the permissions of that user will change. He will have the same permissions, with identical permission sources, as the users in the family he was moved into.

Deleting Empty Families

After moving users from one family to another is possible that some families remain without any user and you can delete them.

When a family is left with no users, the **Delete Family** button will be displayed in the upper-right corner of the family panel, which will allow you to remove it from the set.



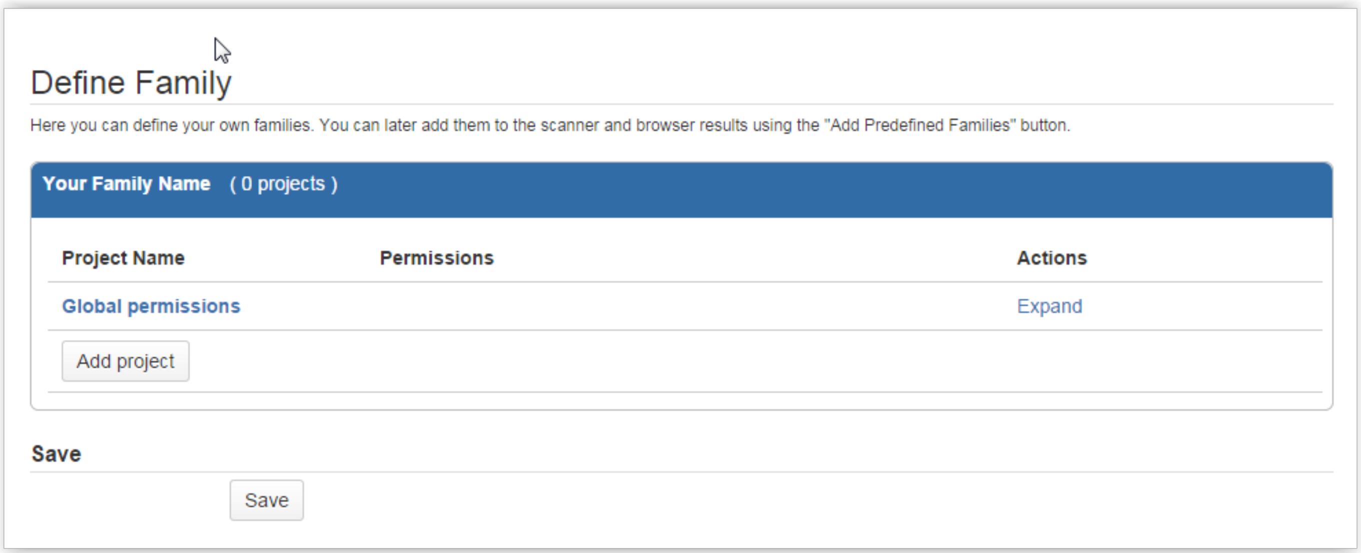
Project Name	Permissions	Actions
Software Project test (SPT)	⚠ ■ ■ ■ ■	Expand View Scheme
Project Test SIMPLE (PTS)	⚠ ■ ■ ■	Expand View Scheme
Test Project KLogger (TPK)	⚠ ■ ■ ■	Expand View Scheme

See Also:

Error formatting macro: contentbylabel: com.atlassian.confluence.api.service.exceptions.BadRequestException: Could not parse cql : null

Define Families

This tab allows you to define your own families. You can later add them to the **Family Scanner** and **Family Browser** results using the "Add Predefined Families" button at the upper-right corner of the family set.



Define Family

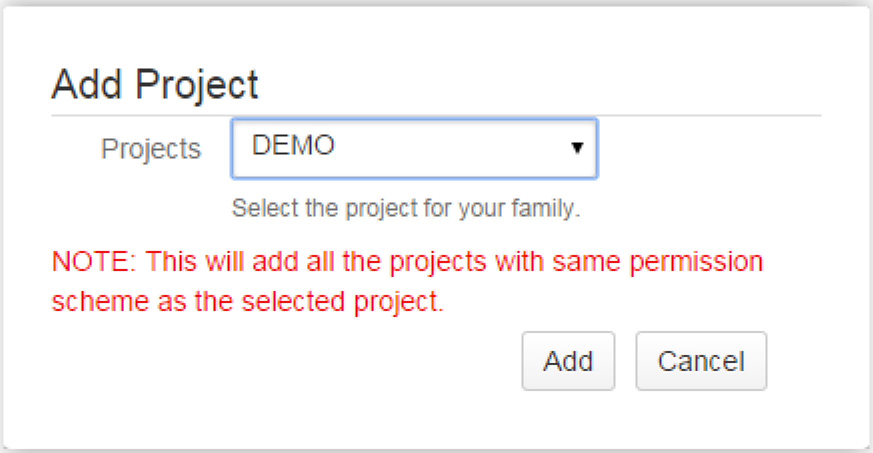
Here you can define your own families. You can later add them to the scanner and browser results using the "Add Predefined Families" button.

Your Family Name (0 projects)

Project Name	Permissions	Actions
Global permissions		Expand

Adding a project

To add a project to your family you have to click the **Add project** button in the bottom-right corner of the family panel. A box will be displayed allowing you to choose a project from a list of projects.



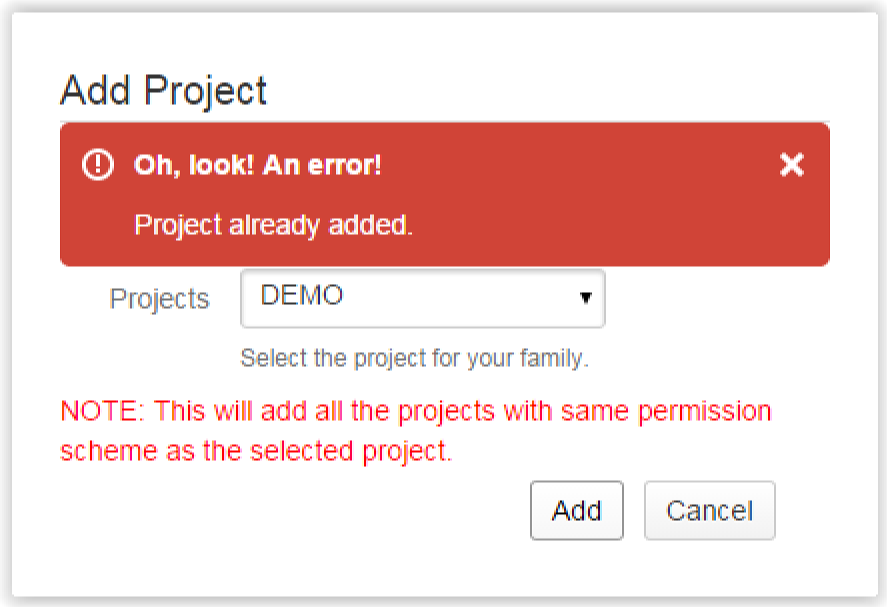
Add Project

Projects: DEMO

Select the project for your family.

NOTE: This will add all the projects with same permission scheme as the selected project.

You cannot add the same project multiple times. If you try to add the same project another time the following error message will appear in the dialog box:



Add Project

Oh, look! An error!

Project already added.

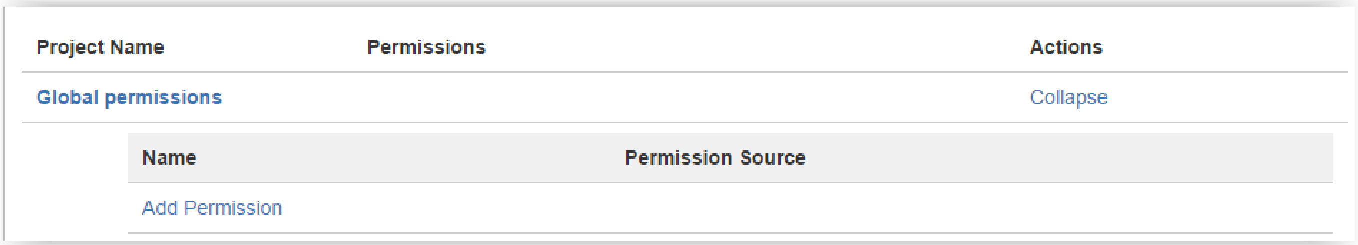
Projects: DEMO

Select the project for your family.

NOTE: This will add all the projects with same permission scheme as the selected project.

Adding a permission

On **Expand** you can see the permissions and sources of the family and add permissions and sources for those permissions.

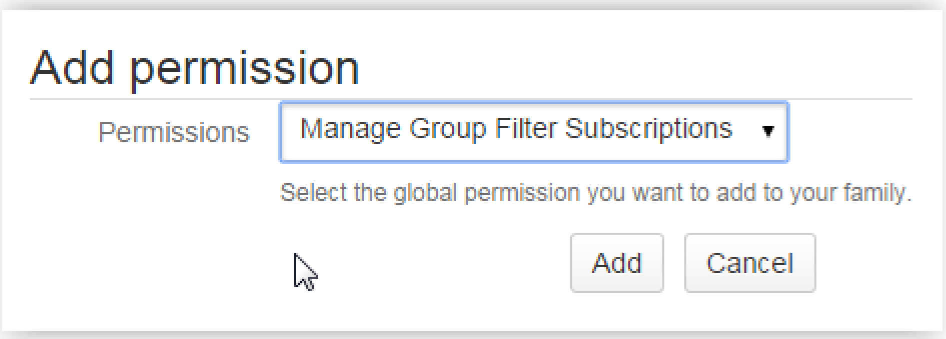


Project Name	Permissions	Actions
Global permissions		Collapse

Name	Permission Source
Add Permission	

To add a permission, click the **Add Permission** link in the expanded table for the corresponding project or global permissions. A dialog will be displayed allowing you to choose a permission.

The dialog for adding **Global Permissions** will look like this:

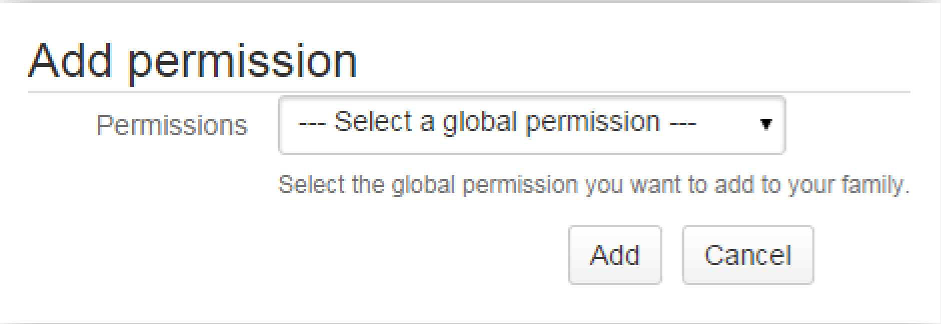


Add permission

Permissions: Manage Group Filter Subscriptions

Select the global permission you want to add to your family.

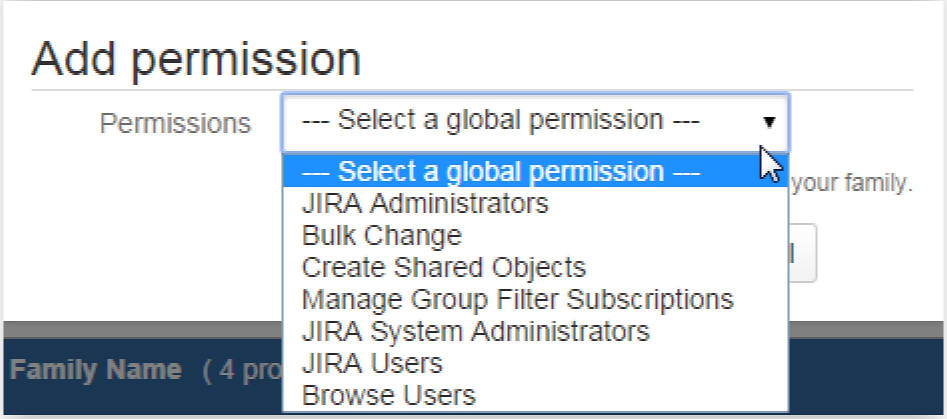
The **Project Permissions** dialog allows you to choose the permission category and then a permission from that category.



Add permission

Permissions: --- Select a global permission --- ▾

Select the global permission you want to add to your family.

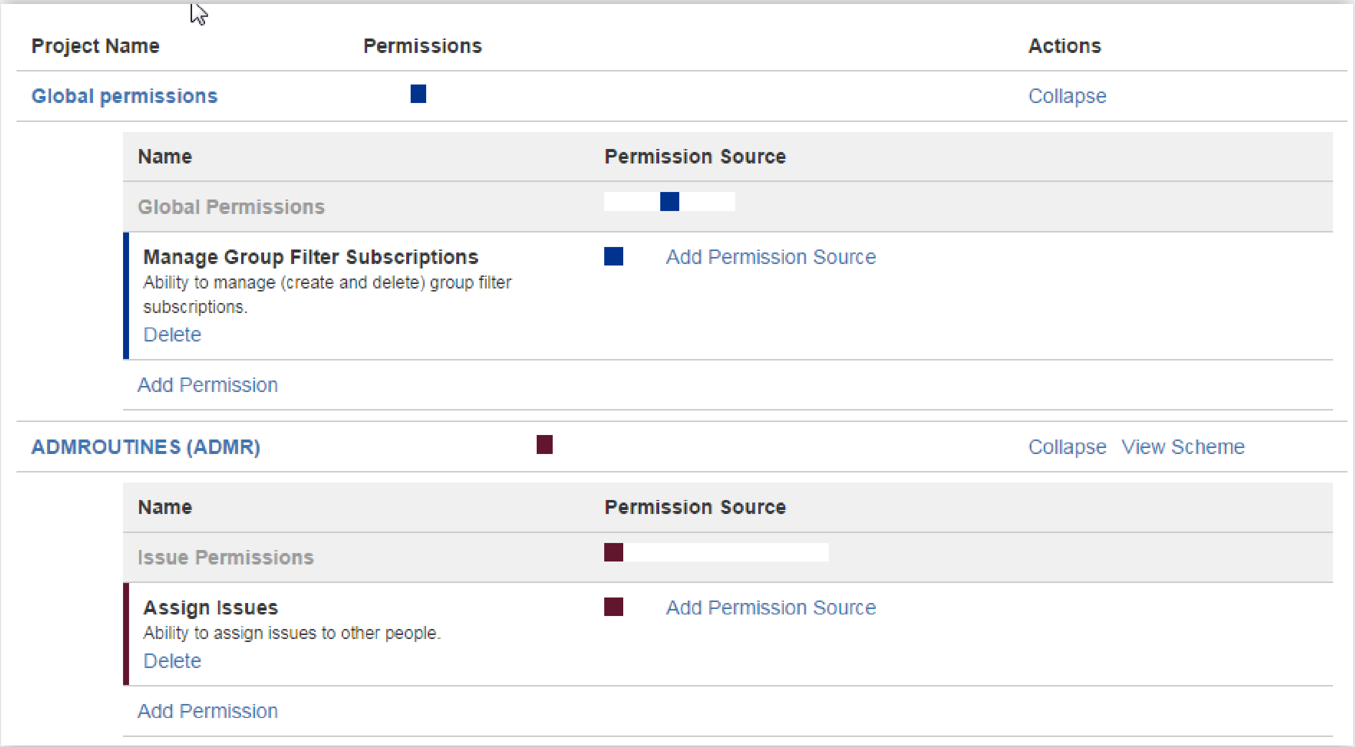


Add permission

Permissions: --- Select a global permission --- ▾

- --- Select a global permission ---
- JIRA Administrators
- Bulk Change
- Create Shared Objects
- Manage Group Filter Subscriptions
- JIRA System Administrators
- JIRA Users
- Browse Users

After adding the permissions, the family will look like this:



Project Name	Permissions	Actions
Global permissions	🟦	Collapse

Name	Permission Source
Global Permissions	🟦
Manage Group Filter Subscriptions Ability to manage (create and delete) group filter subscriptions. Delete	🟦 Add Permission Source
Add Permission	

Project Name	Permissions	Actions
ADMROUTINES (ADMR)	🟥	Collapse View Scheme

Name	Permission Source
Issue Permissions	🟥
Assign Issues Ability to assign issues to other people. Delete	🟥 Add Permission Source
Add Permission	

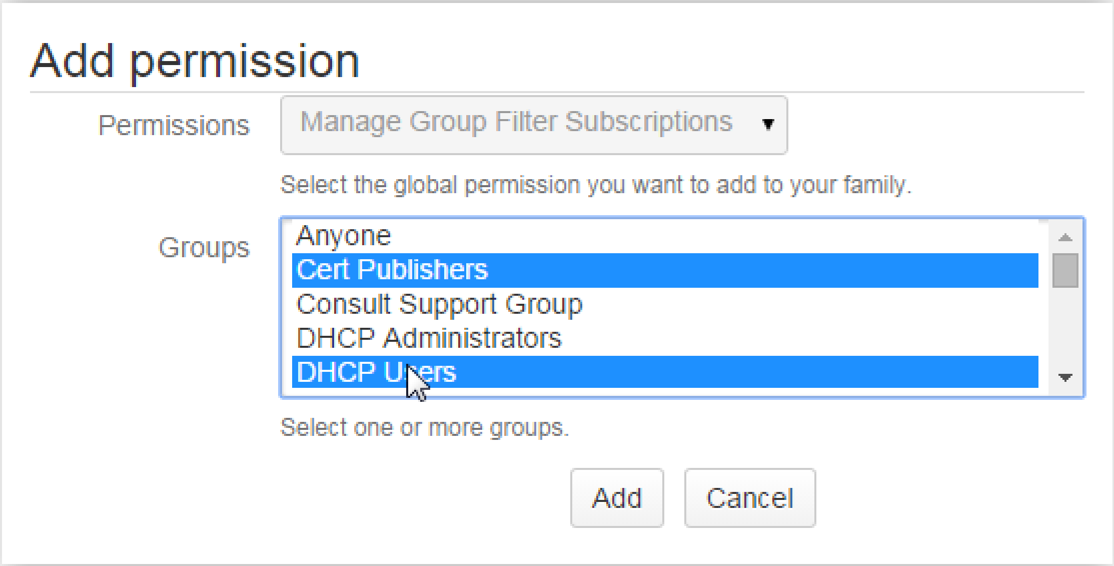
For each permission you have two options:

- **Delete** - will remove the permission with all its sources.
- **Add Permission Source -** add more sources for that permission.

Adding permission sources

You can choose the sources for permissions from the dialog which will be displayed when clicking the **Add Permission Source** link.

The dialog for Global Permissions will look like this:



Add permission

Permissions: Manage Group Filter Subscriptions

Select the global permission you want to add to your family.

Groups: Anyone, Cert Publishers, Consult Support Group, DHCP Administrators, DHCP Users

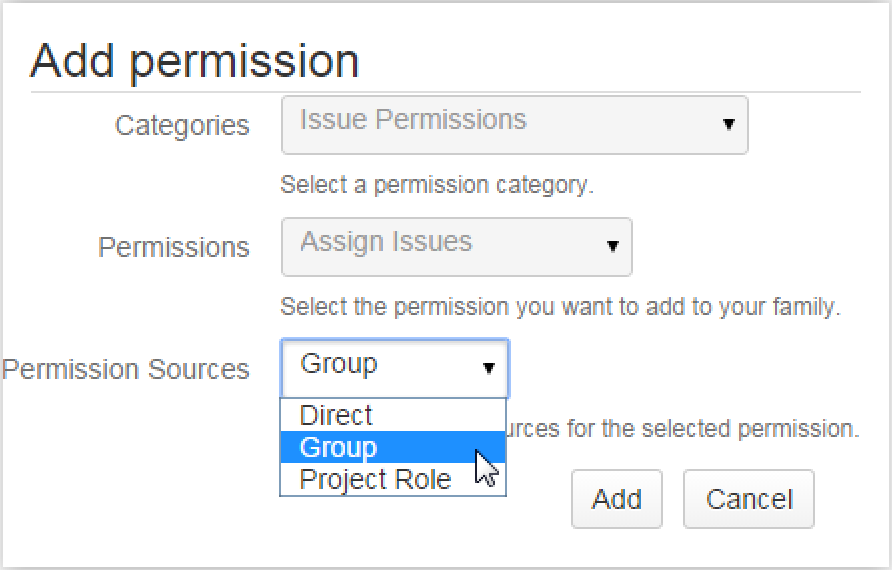
Select one or more groups.

Global permissions are only granted to groups of users.

Project permissions can be granted to:

- Individual users (Direct permission source)
- Groups
- Project Roles

The dialog which allows you to add sources for the project permissions is presented below:



Add permission

Categories: Issue Permissions

Select a permission category.

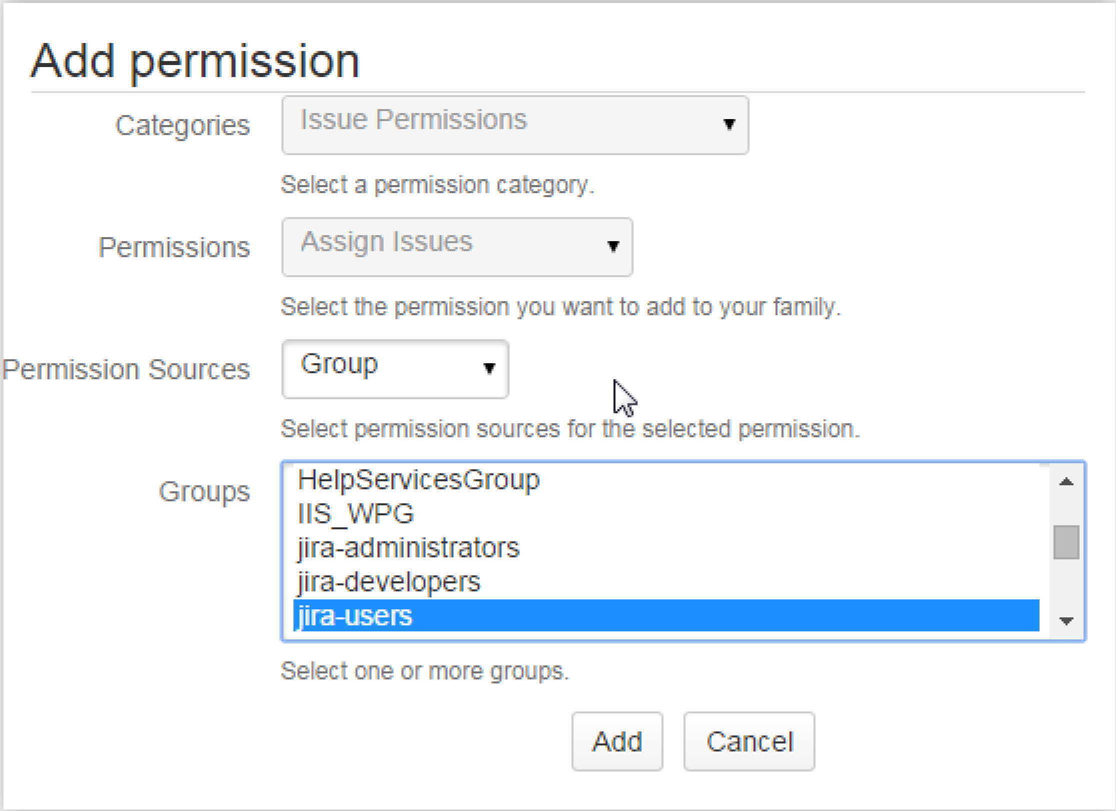
Permissions: Assign Issues

Select the permission you want to add to your family.

Permission Sources: Group

Direct, Group, Project Role

...urces for the selected permission.



Add permission

Categories: Issue Permissions

Select a permission category.

Permissions: Assign Issues

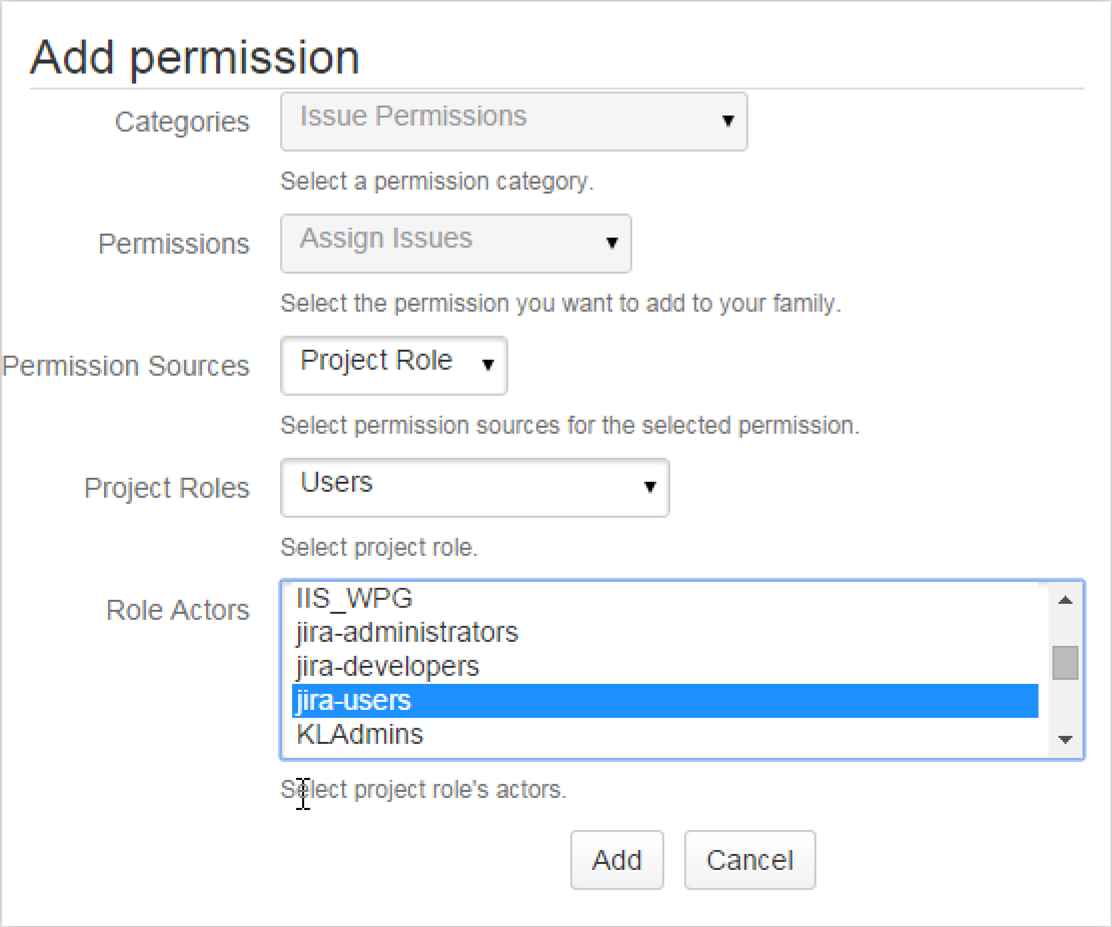
Select the permission you want to add to your family.

Permission Sources: Group

Select permission sources for the selected permission.

Groups:
- HelpServicesGroup
- IIS_WPG
- jira-administrators
- jira-developers
- jira-users

Select one or more groups.



Add permission

Categories: Issue Permissions

Select a permission category.

Permissions: Assign Issues

Select the permission you want to add to your family.

Permission Sources: Project Role

Select permission sources for the selected permission.

Project Roles: Users

Select project role.

Role Actors:
- IIS_WPG
- jira-administrators
- jira-developers
- jira-users
- KLAdmins

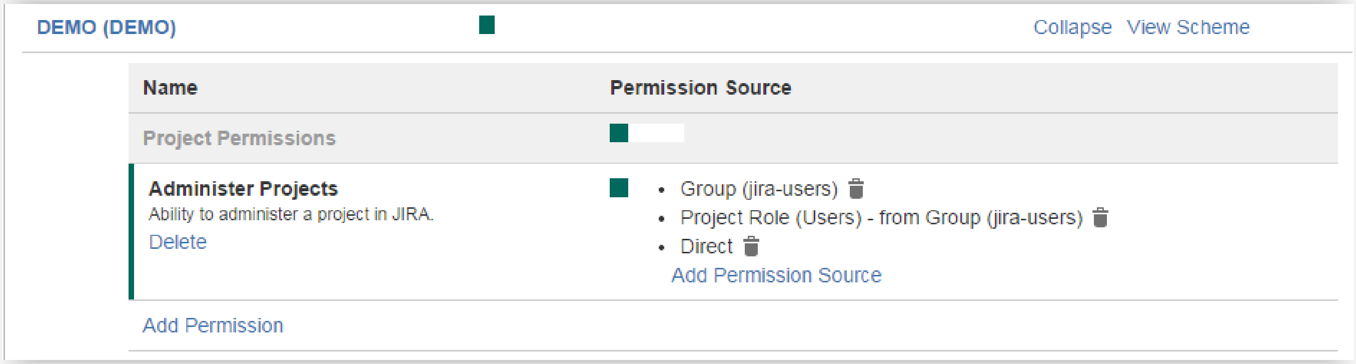
Select project role's actors.

Adding permission sources that do not already exist will be granted when the first user is assigned to the family. For example, say we have no **Browse Project** permission for the **jira-users** group. However, we add this as a permission source in one of the predefined families. When a user is assigned to the family, the **Browse Project** permission will be granted to the **jira-users** group. This can impact other families and permissions, so use this feature with care.

For more information about permission sources, see **User Guide**.

Deleting a permission source

After adding the permission sources the family will look similar to the one below:



DEMO (DEMO) 🟩 Collapse View Scheme

Name	Permission Source
Project Permissions	🟩
Administer Projects Ability to administer a project in JIRA. Delete	🟩 • Group (jira-users) • Project Role (Users) - from Group (jira-users) • Direct Add Permission Source

Add Permission

You can remove a permission source by clicking the red marker.

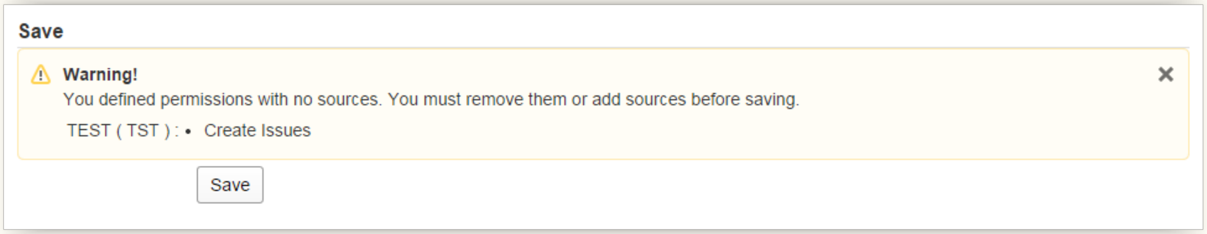
Saving your family

You can save your families by clicking the **Save** button at the bottom of the page. In the **Family Scanner** and **Family Browser** tab you will be able to add them using the "Add Predefined Families" button at the upper-right corner of the family set .

The family will be saved with the default name ("Your family name"), unless you change it before saving.

To rename a family, see **Managing Families.**

You cannot save the family if you defined permissions with no sources.



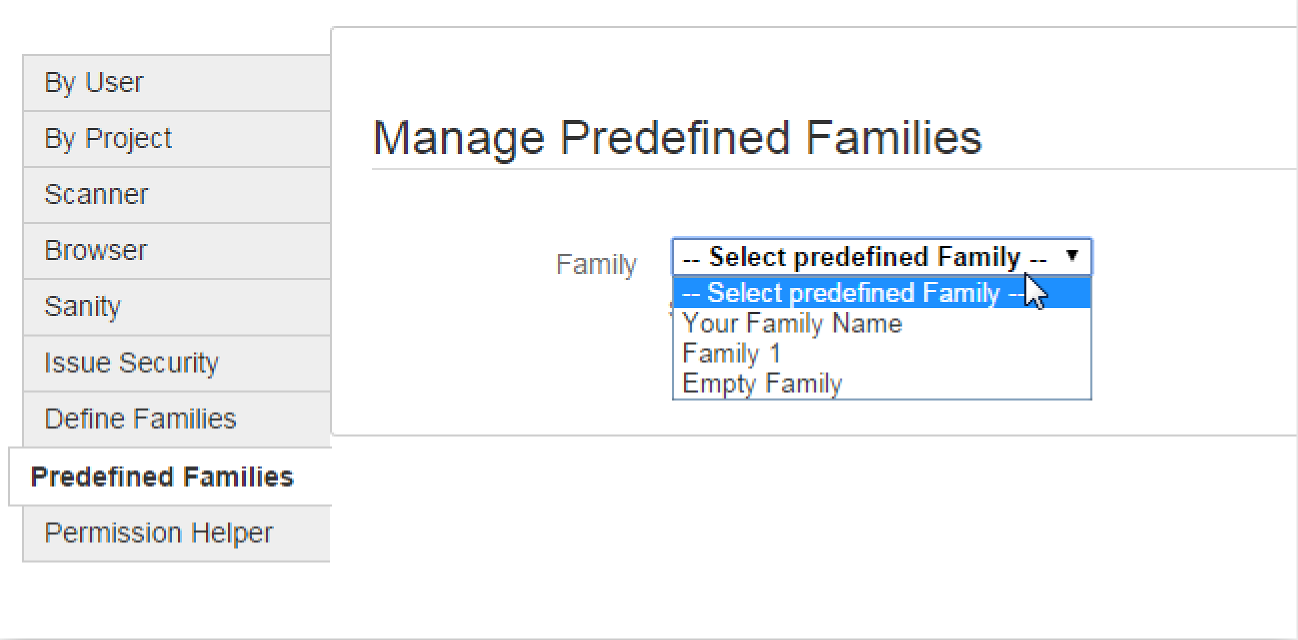
Save

⚠ **Warning!**
You defined permissions with no sources. You must remove them or add sources before saving.
TEST (TST) : • Create Issues

Predefined Families

Since rights-DNA 2.0.3

This tab allows you to manage the families you defined in the **Define Families** tab. You can delete them or modify their permissions and sources.



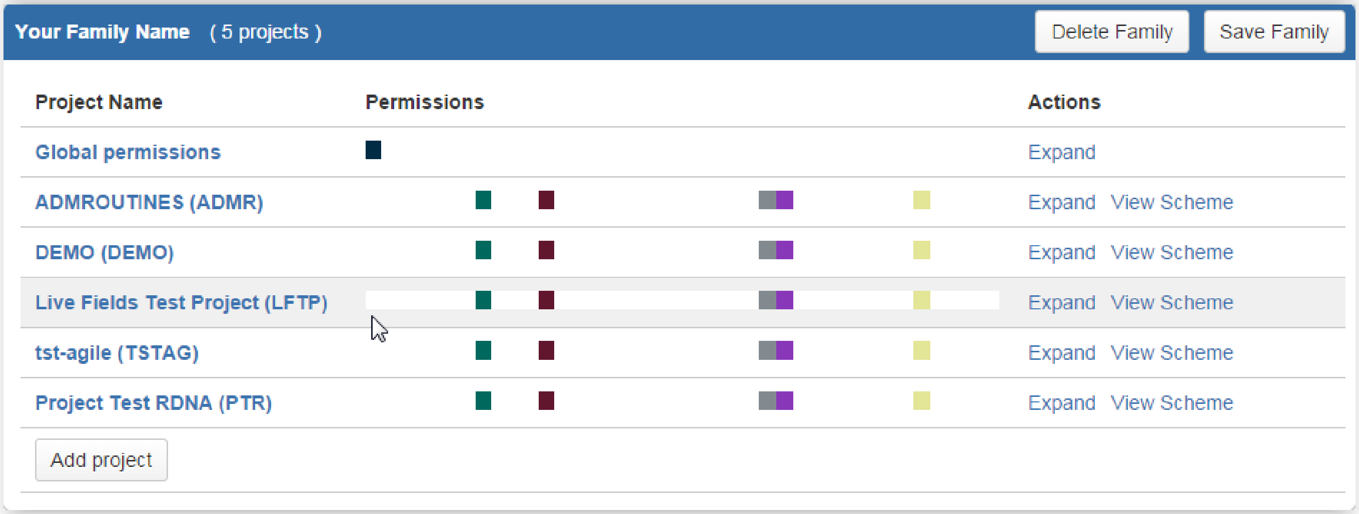
- By User
- By Project
- Scanner
- Browser
- Sanity
- Issue Security
- Define Families
- **Predefined Families**
- Permission Helper

Manage Predefined Families

Family: -- Select predefined Family --

- -- Select predefined Family --
- Your Family Name
- Family 1
- Empty Family

A family will be displayed after selecting a family name from the select box.



Your Family Name (5 projects) — Delete Family — Save Family

Project Name	Permissions	Actions
Global permissions	🟦	Expand
ADMROUTINES (ADMR)	🟩 🟥 ⬛🟪 🟨	Expand View Scheme
DEMO (DEMO)	🟩 🟥 ⬛🟪 🟨	Expand View Scheme
Live Fields Test Project (LFTP)	🟩 🟥 ⬛🟪 🟨	Expand View Scheme
tst-agile (TSTAG)	🟩 🟥 ⬛🟪 🟨	Expand View Scheme
Project Test RDNA (PTR)	🟩 🟥 ⬛🟪 🟨	Expand View Scheme

Add project

Delete family

To delete a family you have to click the Delete Family button in the top-right corner of the family panel.

Modify family

For more information about adding/removing permissions and sources, see **Define Families**.

Save family

After you modify your family you can save the changes by clicking the **Save Family** button in the top-right corner of the family panel. In the **Famil**

y Scanner and **Family Browser** tab you will be able to add them using the "Add Predefined Families" button at the upper-right corner of the family set .

The family name will remain the same, unless you change it before saving.

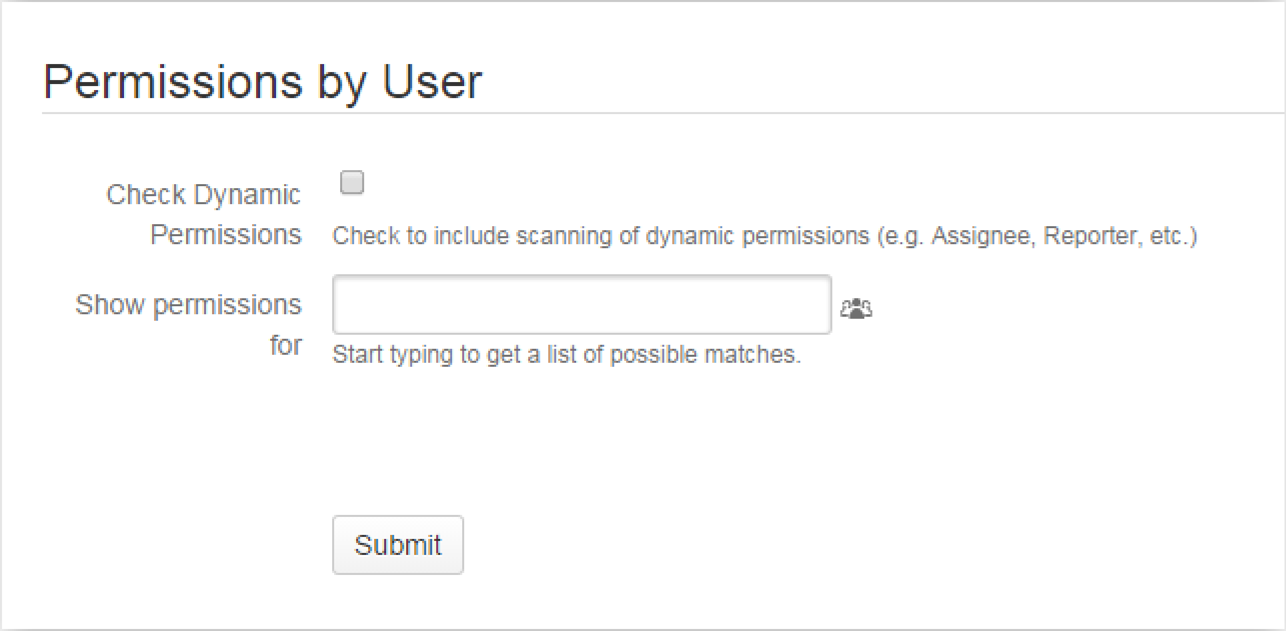
> You cannot save two families with same name or same permission sources.

> To rename a family see, **Managing Families**.

Permissions By User

Scanning

This tab allows you to select a user and see what permissions are granted to him across the projects.



Permissions by User

Check Dynamic Permissions □
Check to include scanning of dynamic permissions (e.g. Assignee, Reporter, etc.)

Show permissions for
Start typing to get a list of possible matches.

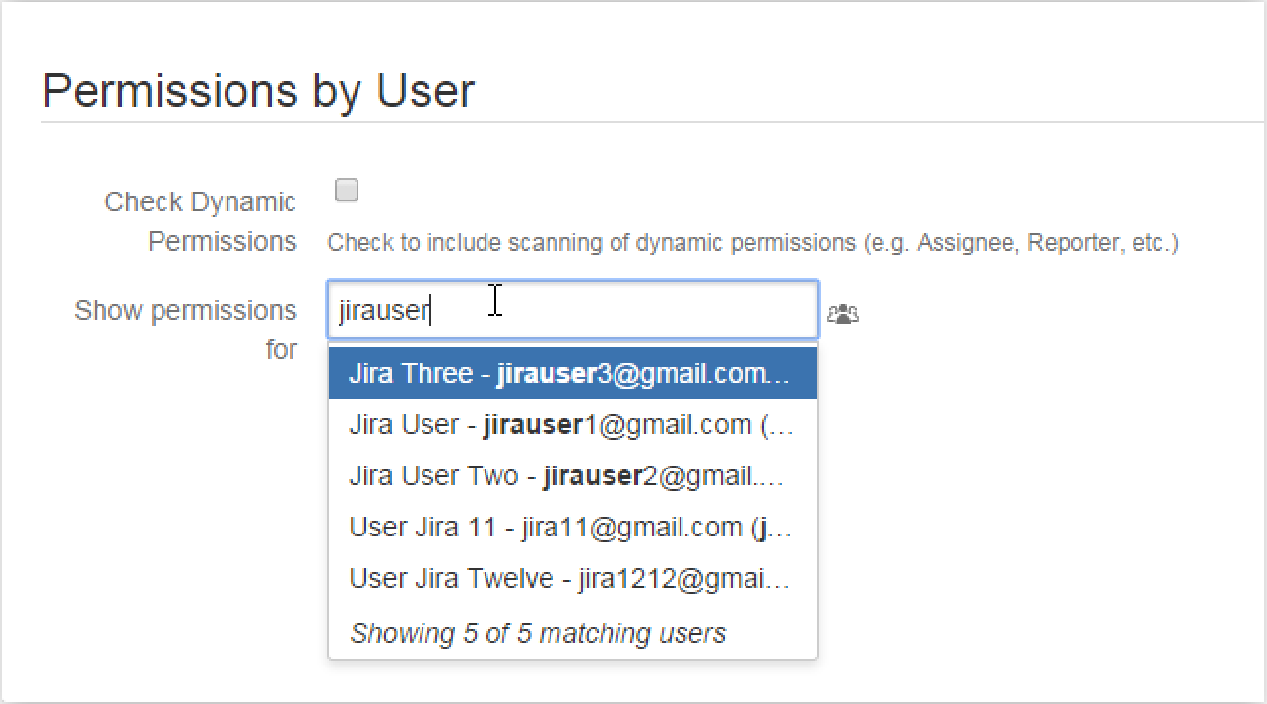
Submit

The **Check Dynamic Permissions** option permits the inclusion of dynamic permissions like Assignee and Reporter in the scan.

> For more information about dynamic permission sources see **User Guide**.

There are two ways of selecting a user :

- through **autocomplete** - you will get a list of possible matches after starting typing.



Permissions by User

Check Dynamic Permissions □ Check to include scanning of dynamic permissions (e.g. Assignee, Reporter, etc.)

Show permissions for jirauser

Jira Three - **jirauser**3@gmail.com…

Jira User - **jirauser**1@gmail.com (…

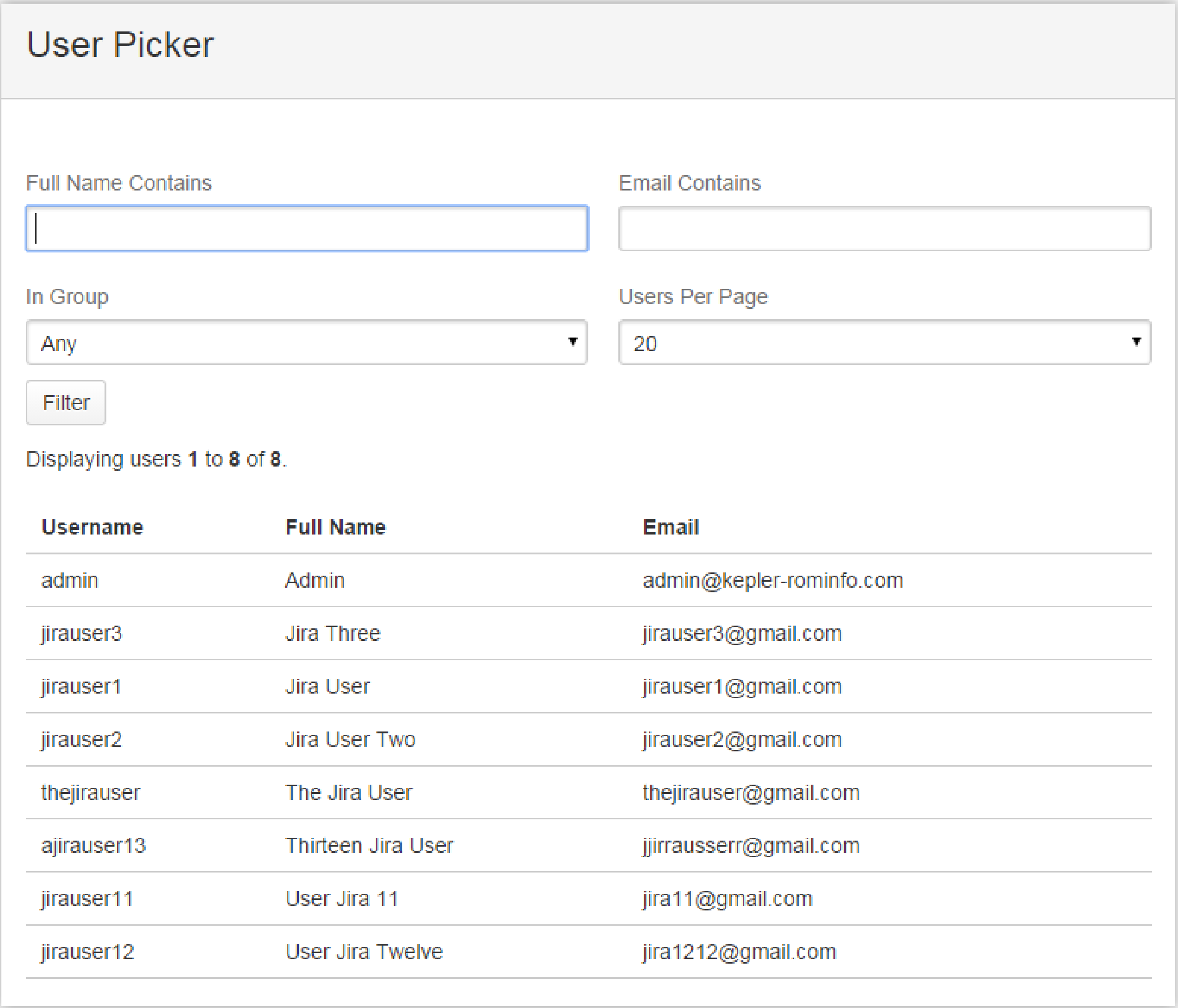
Jira User Two - **jirauser**2@gmail.…

User Jira 11 - jira11@gmail.com (**j**…

User Jira Twelve - jira1212@gmai…

Showing 5 of 5 matching users

- using the **pop-up** - clicking on the icon will show the dialog you are accustomed to, like in the image below:



User Picker

Full Name Contains

Email Contains

In Group: Any

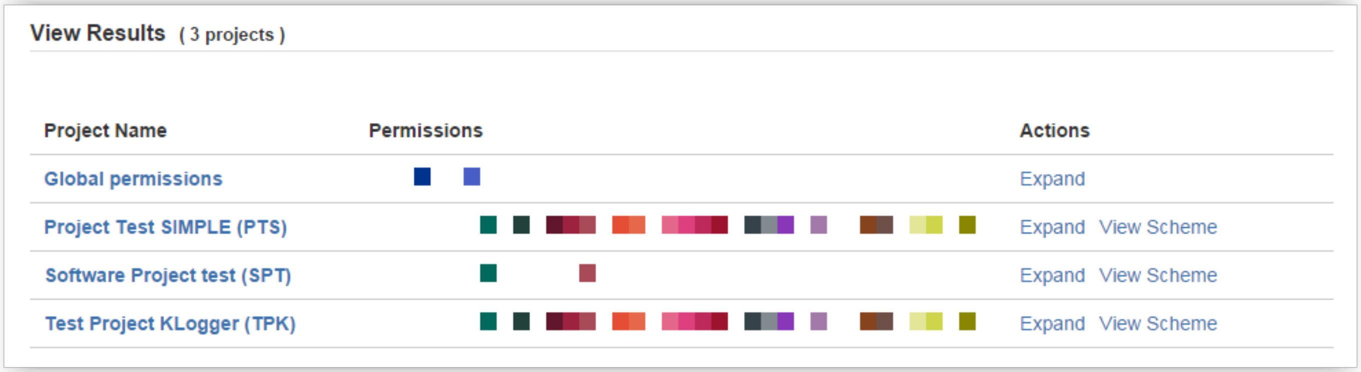
Users Per Page: 20

Displaying users **1** to **8** of **8**.

Username	Full Name	Email
admin	Admin	admin@kepler-rominfo.com
jirauser3	Jira Three	jirauser3@gmail.com
jirauser1	Jira User	jirauser1@gmail.com
jirauser2	Jira User Two	jirauser2@gmail.com
thejirauser	The Jira User	thejirauser@gmail.com
ajirauser13	Thirteen Jira User	jjirrausserr@gmail.com
jirauser11	User Jira 11	jira11@gmail.com
jirauser12	User Jira Twelve	jira1212@gmail.com

Results

After setting your options and clicking submit the results will be displayed as:



View Results (3 projects)

Project Name	Permissions	Actions
Global permissions	■ ■	Expand
Project Test SIMPLE (PTS)	■ ■ ■■■ ■■ ■■■■ ■■■ ■ ■■ ■■ ■	Expand View Scheme
Software Project test (SPT)	■ ■	Expand View Scheme
Test Project KLogger (TPK)	■ ■ ■■■ ■■ ■■■■ ■■■ ■ ■■ ■■ ■	Expand View Scheme

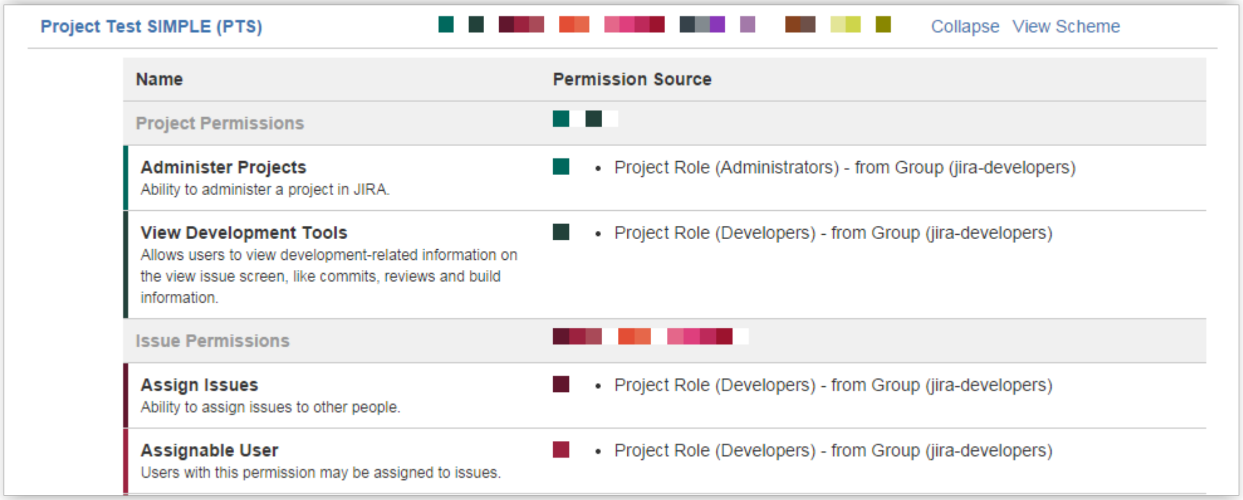
The table shows the projects the user has permissions on and, in the first row, the global permissions the user has. Each color represents a permission (for more information see User Guide) and shows that the user has that permission. If the placeholder for a certain permission is white, it means that the user does not have that permission.

For each project there is a link to the project page (by clicking on the name of the project) and to the permission scheme of that project.

Details

On **Expand** you can see from where a permission is granted (the sources of the permission).

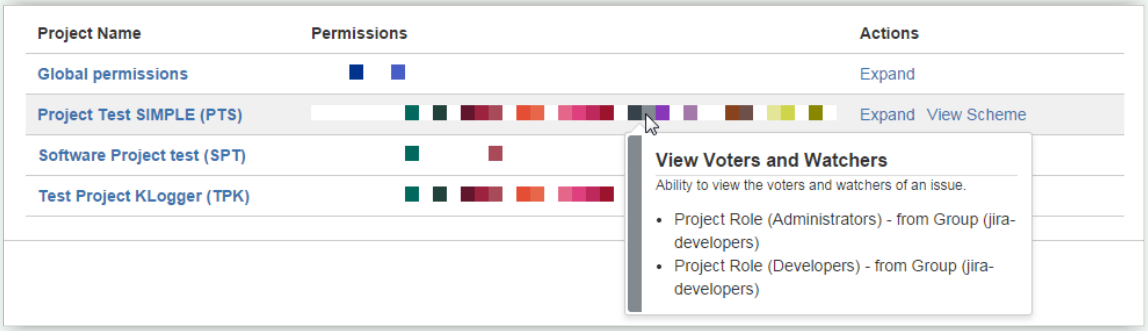
> For more information about permission sources see **User Guide**.



Project Test SIMPLE (PTS) ■ ■ ■■■ ■■ ■■■■ ■■■ ■ ■■ ■■ ■ Collapse View Scheme

Name	Permission Source
Project Permissions	■ ■
Administer Projects Ability to administer a project in JIRA.	■ • Project Role (Administrators) - from Group (jira-developers)
View Development Tools Allows users to view development-related information on the view issue screen, like commits, reviews and build information.	■ • Project Role (Developers) - from Group (jira-developers)
Issue Permissions	■■■ ■■ ■■■■
Assign Issues Ability to assign issues to other people.	■ • Project Role (Developers) - from Group (jira-developers)
Assignable User Users with this permission may be assigned to issues.	■ • Project Role (Developers) - from Group (jira-developers)

> It is also possible to see the permission sources by hovering over the permission color you are interested in.



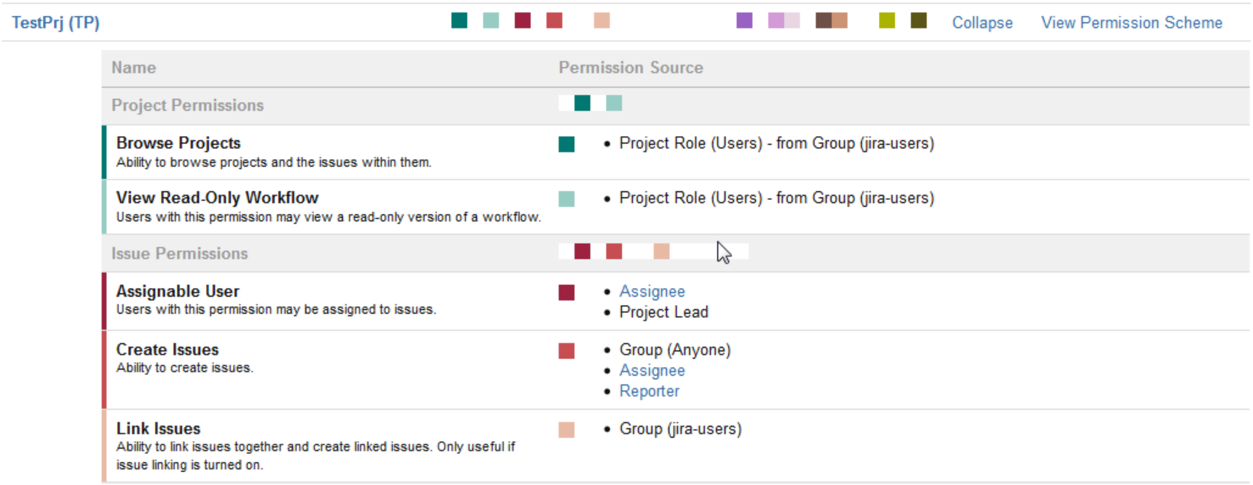
Project Name	Permissions	Actions
Global permissions	■ ■	Expand
Project Test SIMPLE (PTS)	■ ■ ■■■ ■■ ■■■■ ■■■ ■ ■■ ■■ ■	Expand View Scheme
Software Project test (SPT)	■ ■	
Test Project KLogger (TPK)	■ ■ ■■■ ■■ ■■■■	

View Voters and Watchers

Ability to view the voters and watchers of an issue.

- Project Role (Administrators) - from Group (jira-developers)
- Project Role (Developers) - from Group (jira-developers)

Check Dynamic Permissions filter will display in result set the permissions that user **CAN** have as Assignee or Reporter of an issue.



TestPrj (TP) ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ Collapse View Permission Scheme

Name		Permission Source
Project Permissions	■ ■	
Browse Projects Ability to browse projects and the issues within them.	■	- Project Role (Users) - from Group (jira-users)
View Read-Only Workflow Users with this permission may view a read-only version of a workflow.	■	- Project Role (Users) - from Group (jira-users)
Issue Permissions	■ ■ ■	
Assignable User Users with this permission may be assigned to issues.	■	- Assignee - Project Lead
Create Issues Ability to create issues.	■	- Group (Anyone) - Assignee - Reporter
Link Issues Ability to link issues together and create linked issues. Only useful if issue linking is turned on.	■	- Group (jira-users)

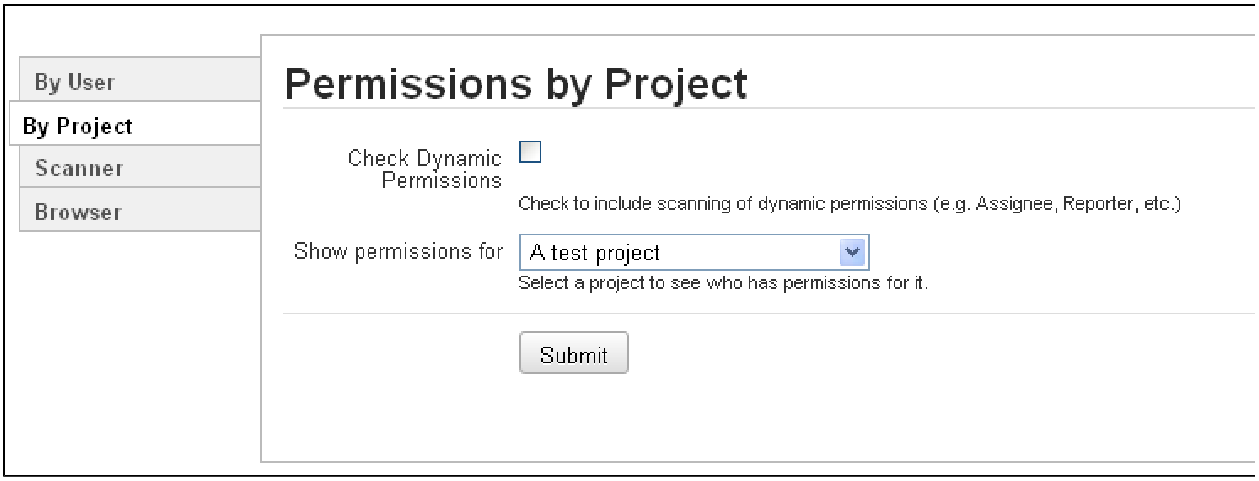
See Also:

Error formatting macro: contentbylabel: com.atlassian.confluence.api.service.exceptions.BadRequestException: Could not parse cql : null

Permissions By Project

Scanning

This tab shows all the users who are granted permissions for the selected project.



Permissions by Project

Check Dynamic Permissions □
Check to include scanning of dynamic permissions (e.g. Assignee, Reporter, etc.)

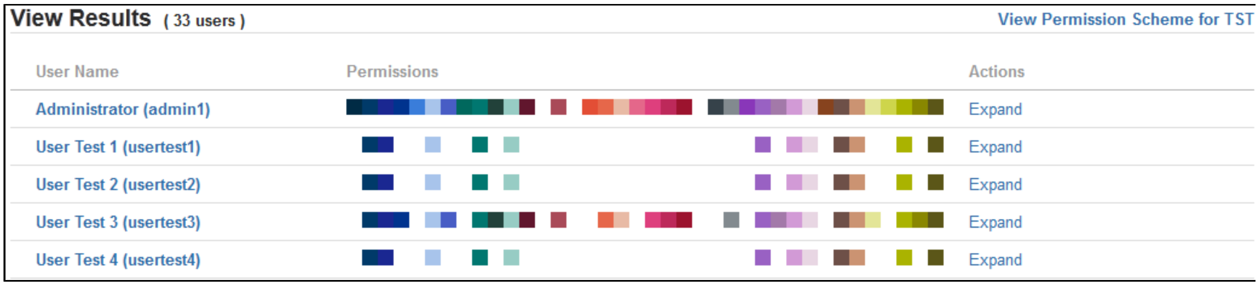
Show permissions for: A test project
Select a project to see who has permissions for it.

The **Check Dynamic Permissions** option allows the inclusion of dynamic permissions like Assignee and Reporter in the scan.

For more information about dynamic permission sources see **User Guide**.

Results

The results will be displayed as:



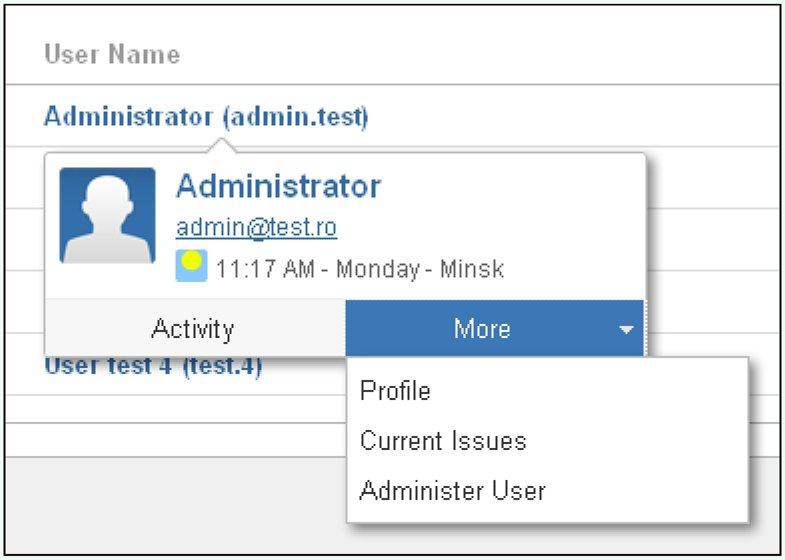
View Results (33 users) — View Permission Scheme for TST

User Name	Permissions	Actions
Administrator (admin1)		Expand
User Test 1 (usertest1)		Expand
User Test 2 (usertest2)		Expand
User Test 3 (usertest3)		Expand
User Test 4 (usertest4)		Expand

The table shows the users who have permissions on the selected project and the global permissions each user has. Each color represents a permission and shows that the user has that permission. If the placeholder for a certain permission is white, it means that the user does not have that permission. (for more information see **User Guide**)

Each user has a link to the user details page (by clicking on the user's name). To see the **Permission Scheme** of the selected project you can click the link displayed in the top-right corner of the results table.

By hovering over the user names, a menu will appear with detailed information about the user and other useful links.

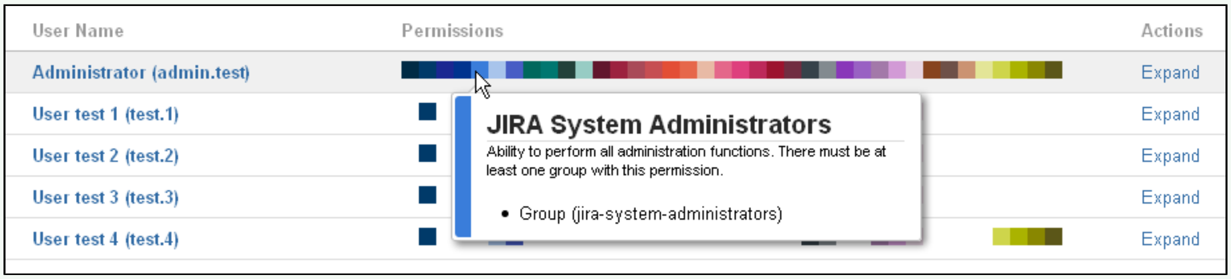


User Name

Administrator (admin.test)

Administrator
admin@test.ro
11:17 AM - Monday - Minsk

By hovering over a color, the permission's description and sources will be displayed.

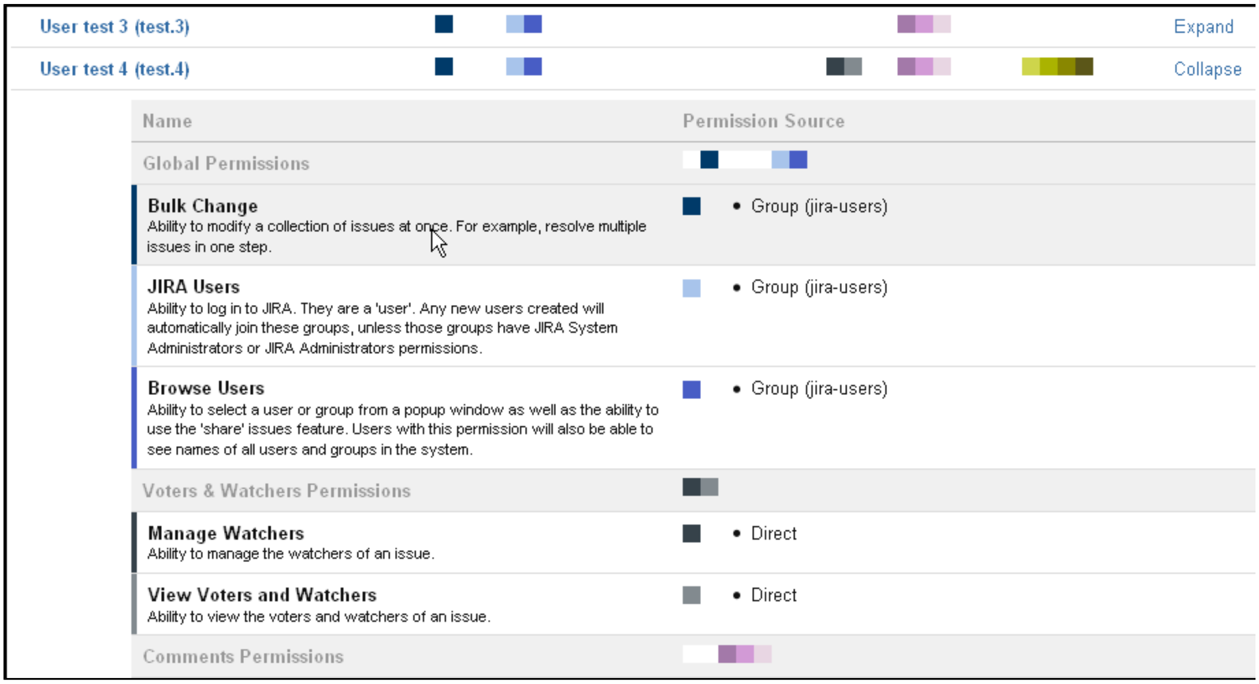


User Name	Permissions	Actions
Administrator (admin.test)		Expand
User test 1 (test.1)		Expand
User test 2 (test.2)		Expand
User test 3 (test.3)		Expand
User test 4 (test.4)		Expand

JIRA System Administrators
Ability to perform all administration functions. There must be at least one group with this permission.

- Group (jira-system-administrators)

Expanding a row will display the user's permissions details.



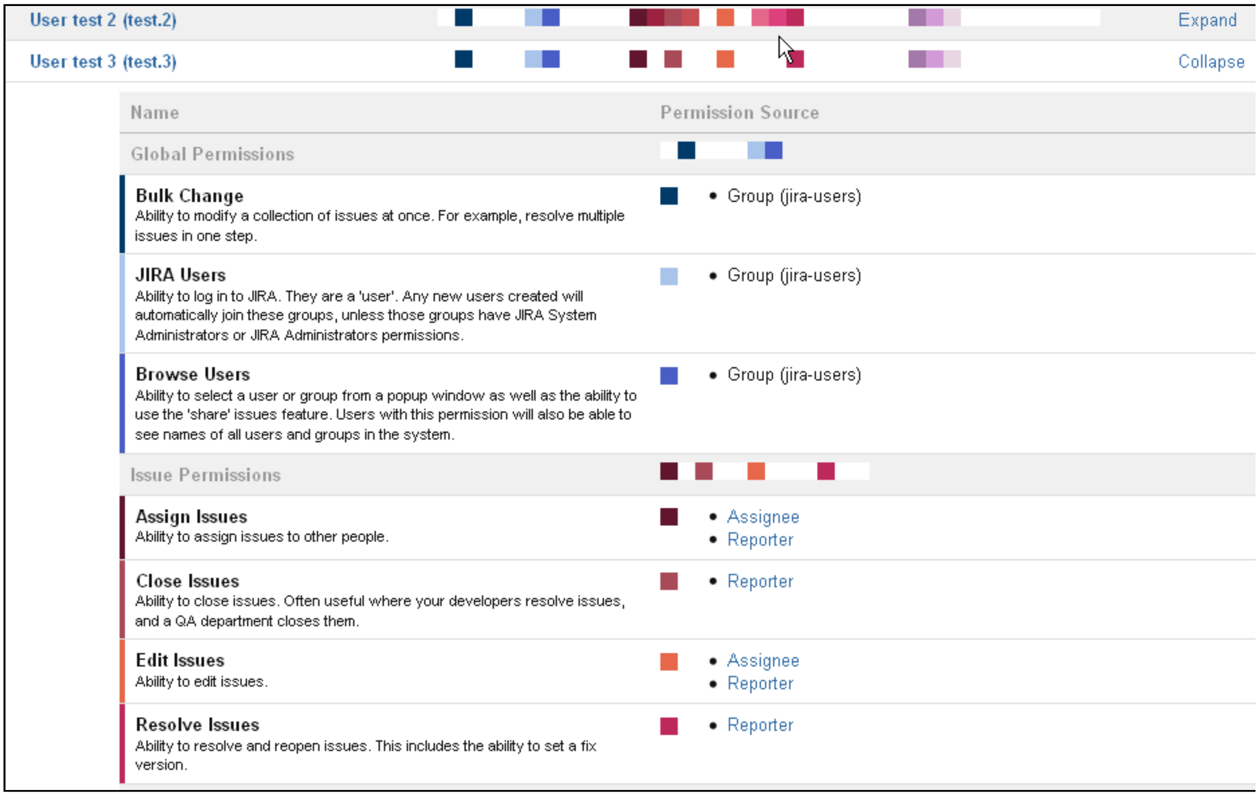
User test 3 (test.3) ■ ■■ ■■■ Expand

User test 4 (test.4) ■ ■■ ■■ ■■■ ■■■■ Collapse

Name	Permission Source
Global Permissions	■ ■■
Bulk Change Ability to modify a collection of issues at once. For example, resolve multiple issues in one step.	■ • Group (jira-users)
JIRA Users Ability to log in to JIRA. They are a 'user'. Any new users created will automatically join these groups, unless those groups have JIRA System Administrators or JIRA Administrators permissions.	■ • Group (jira-users)
Browse Users Ability to select a user or group from a popup window as well as the ability to use the 'share' issues feature. Users with this permission will also be able to see names of all users and groups in the system.	■ • Group (jira-users)
Voters & Watchers Permissions	■■
Manage Watchers Ability to manage the watchers of an issue.	■ • Direct
View Voters and Watchers Ability to view the voters and watchers of an issue.	■ • Direct
Comments Permissions	■■■

For more information about the permissions and sources, see **User Guide.**

Check Dynamic Permissions filter will display in result set the permissions that user **CAN** have as Assignee or Reporter of an issue.



User test 2 (test.2) ■ ■■ ■■■■ ■ ■■■ ■■■ Expand

User test 3 (test.3) ■ ■■ ■ ■ ■ ■ ■■■ Collapse

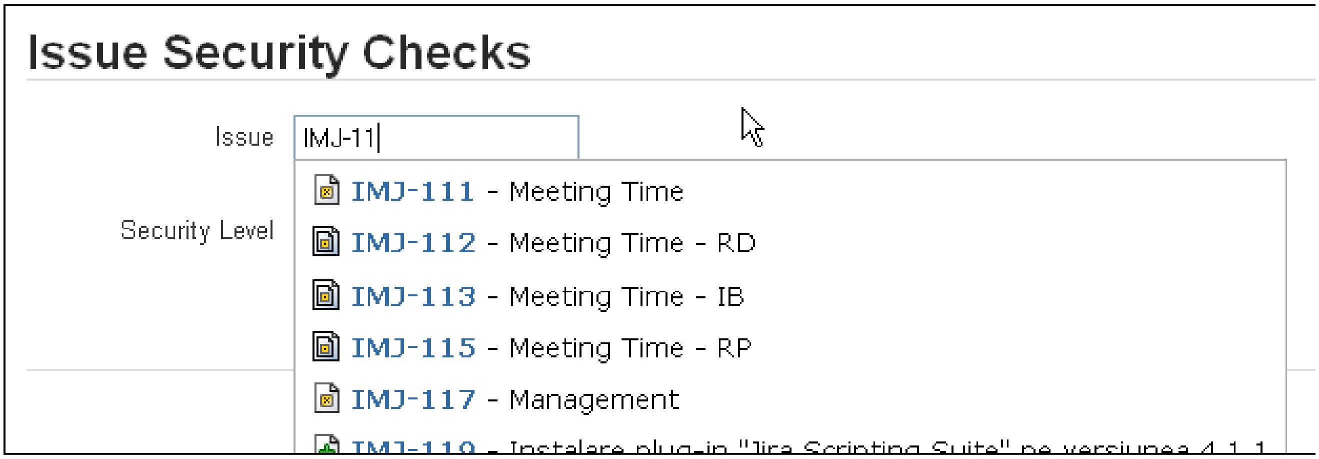
Name	Permission Source
Global Permissions	■ ■■
Bulk Change Ability to modify a collection of issues at once. For example, resolve multiple issues in one step.	■ • Group (jira-users)
JIRA Users Ability to log in to JIRA. They are a 'user'. Any new users created will automatically join these groups, unless those groups have JIRA System Administrators or JIRA Administrators permissions.	■ • Group (jira-users)
Browse Users Ability to select a user or group from a popup window as well as the ability to use the 'share' issues feature. Users with this permission will also be able to see names of all users and groups in the system.	■ • Group (jira-users)
Issue Permissions	■ ■ ■ ■
Assign Issues Ability to assign issues to other people.	■ • Assignee • Reporter
Close Issues Ability to close issues. Often useful where your developers resolve issues, and a QA department closes them.	■ • Reporter
Edit Issues Ability to edit issues.	■ • Assignee • Reporter
Resolve Issues Ability to resolve and reopen issues. This includes the ability to set a fix version.	■ • Reporter

See Also:

Error formatting macro: contentbylabel: com.atlassian.confluence.api.service.exceptions.BadRequestException: Could not parse cql : null

Issue Security

The **Issue Security** tab shows the users who can see a specific issue based on its **security level**.

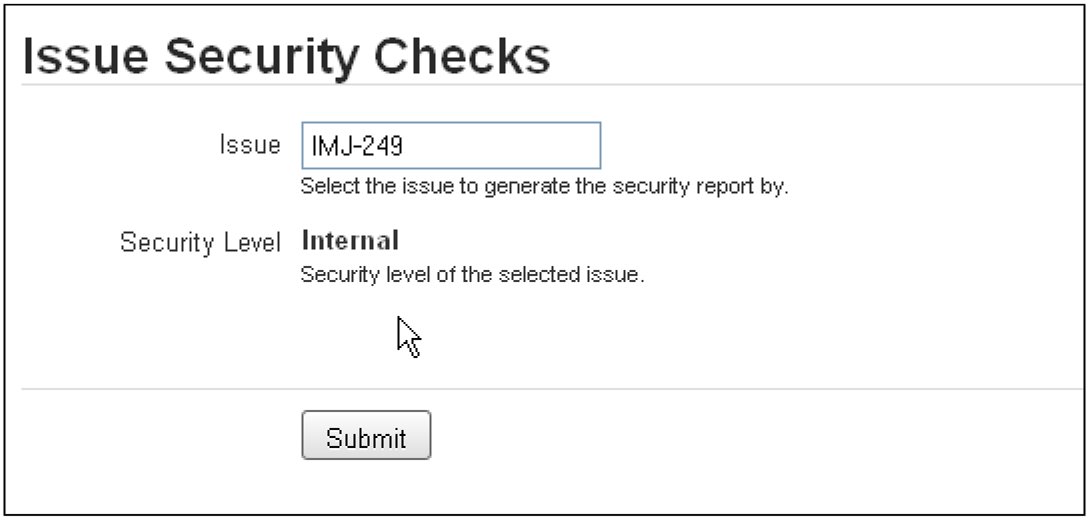


Issue Security Checks

Issue: IMJ-11

IMJ-111 - Meeting Time
IMJ-112 - Meeting Time - RD
IMJ-113 - Meeting Time - IB
IMJ-115 - Meeting Time - RP
IMJ-117 - Management
IMJ-119 - Instalare plug-in "Jira Scripting Suite" pe versiunea 4.1.1

Once you have selected the issue, the **Security Level** field below it will immediately update to show the security level of the specified issue.

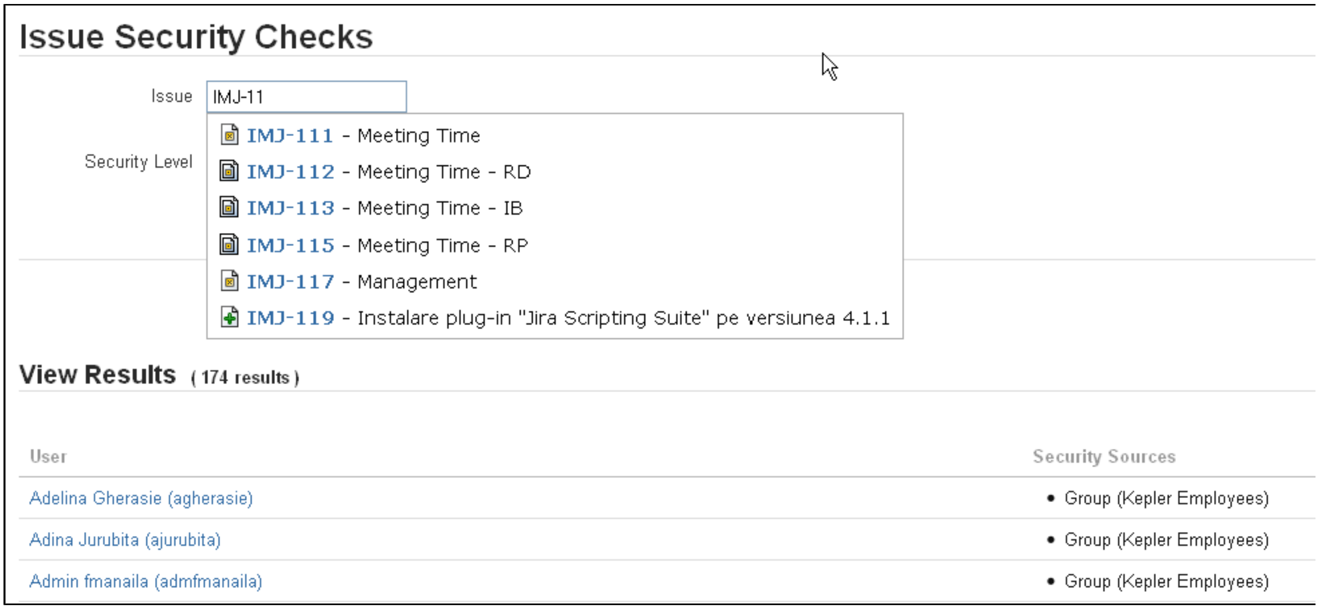


Issue Security Checks

Issue: IMJ-249
Select the issue to generate the security report by.

Security Level: **Internal**
Security level of the selected issue.

To see the users who have the security level required for the issue, click **Submit.**



Issue Security Checks

Issue: IMJ-11

Security Level

- IMJ-111 - Meeting Time
- IMJ-112 - Meeting Time - RD
- IMJ-113 - Meeting Time - IB
- IMJ-115 - Meeting Time - RP
- IMJ-117 - Management
- IMJ-119 - Instalare plug-in "Jira Scripting Suite" pe versiunea 4.1.1

View Results (174 results)

User	Security Sources
Adelina Gherasie (agherasie)	• Group (Kepler Employees)
Adina Jurubita (ajurubita)	• Group (Kepler Employees)
Admin fmanaila (admfmanaila)	• Group (Kepler Employees)

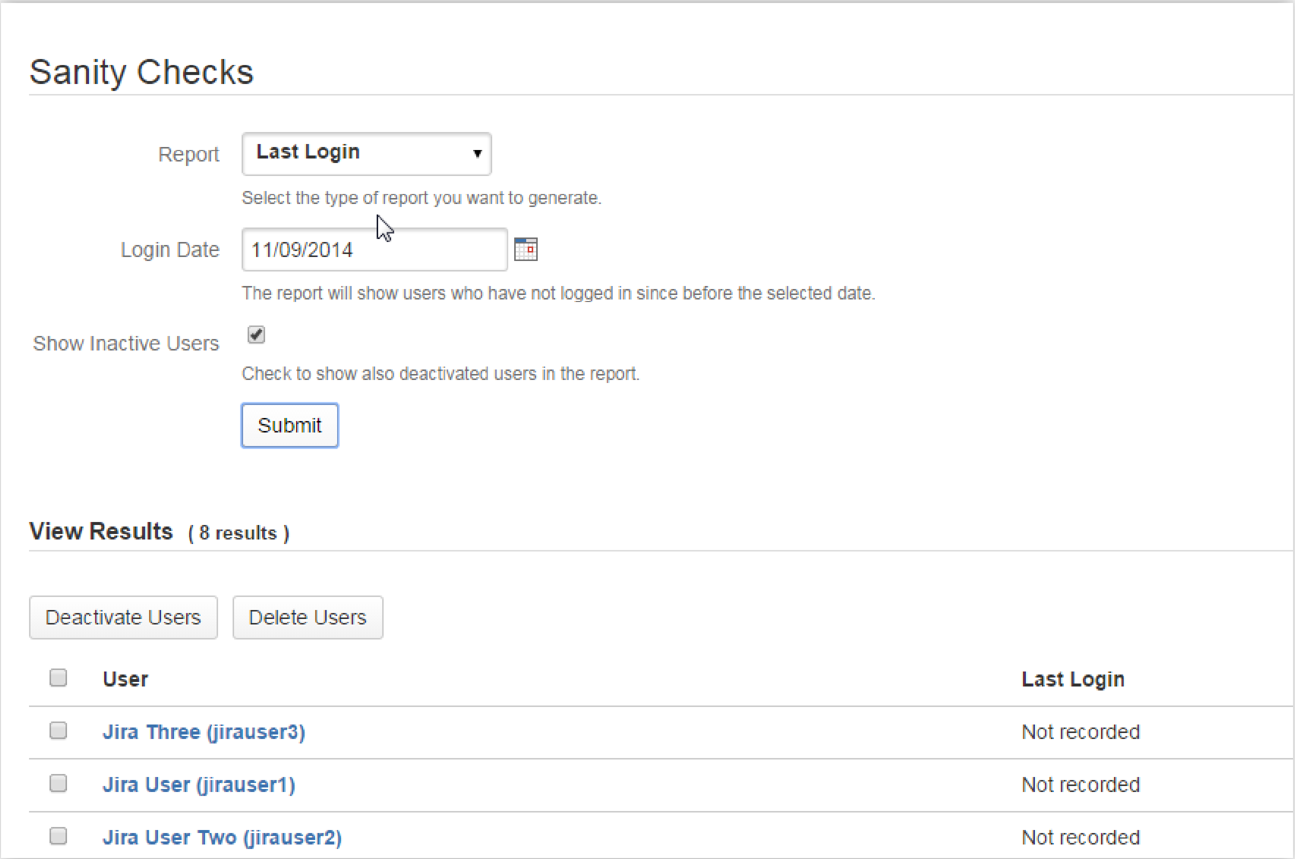
Not all users shown in the report can actually see the specified issue unless they also have the **Browse Project** permission for the project the issue belongs to.

Other Sanity Checks

The **Sanity** tab allows you to generate reports useful for maintaining a clean JIRA environment.

Last Login

The **Last Login** report will show all users who have not logged in since before a specified date. This helps keep track of inactive users that still count toward your license limit.



Sanity Checks

Report: Last Login ▾

Select the type of report you want to generate.

Login Date: 11/09/2014

The report will show users who have not logged in since before the selected date.

Show Inactive Users: ☑

Check to show also deactivated users in the report.

View Results (8 results)

☐	User	Last Login
☐	Jira Three (jirauser3)	Not recorded
☐	Jira User (jirauser1)	Not recorded
☐	Jira User Two (jirauser2)	Not recorded

You can **deactivate** (remove user from all the groups he/she belongs to and remove him/her from any project roles he/she might have) and **delete** (remove user from JIRA) users. To do that you have to check the users you want to deactivate/delete.

To select all users click the checkbox from the **User** column.

On **Delete Users** a box will be displayed to confirm if you want to continue with this action.

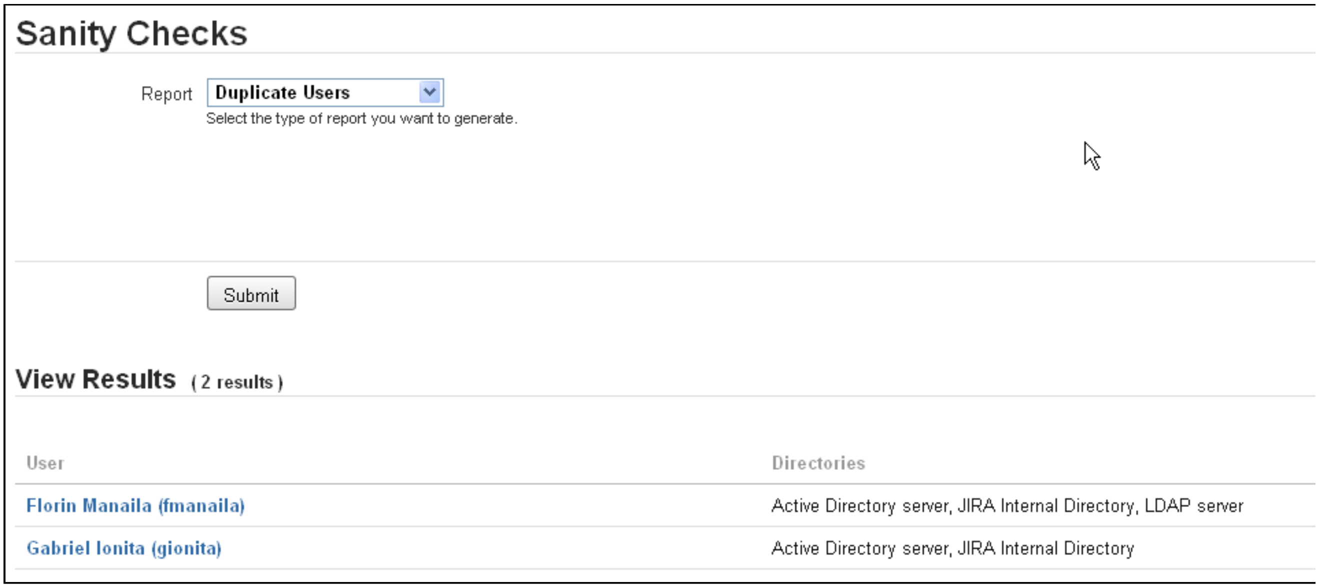
> Since version 2.6.3, deactivating a user just sets its active flag to false, keeping its associated groups and project roles.
>
> In case of users from read-only LDAP directory, for which the active flag can't be updated, they are only removed from all groups associated to the following global permissions: JIRA Users, Administrators and System Administrators.

Duplicate Users

The **Duplicate Users** report shows the users who can be found in multiple directories.

The report is aware of the mirrored user directories from the **Configuration** page. This means that users who only belong to pairs of directories which are already configured are ignored from the report.

The **Duplicate Users** report is helpful because the users shown might not belong to the same groups in both directories and if the primary crashes, those users might end up having different permissions than expected.



Sanity Checks

Report: Duplicate Users

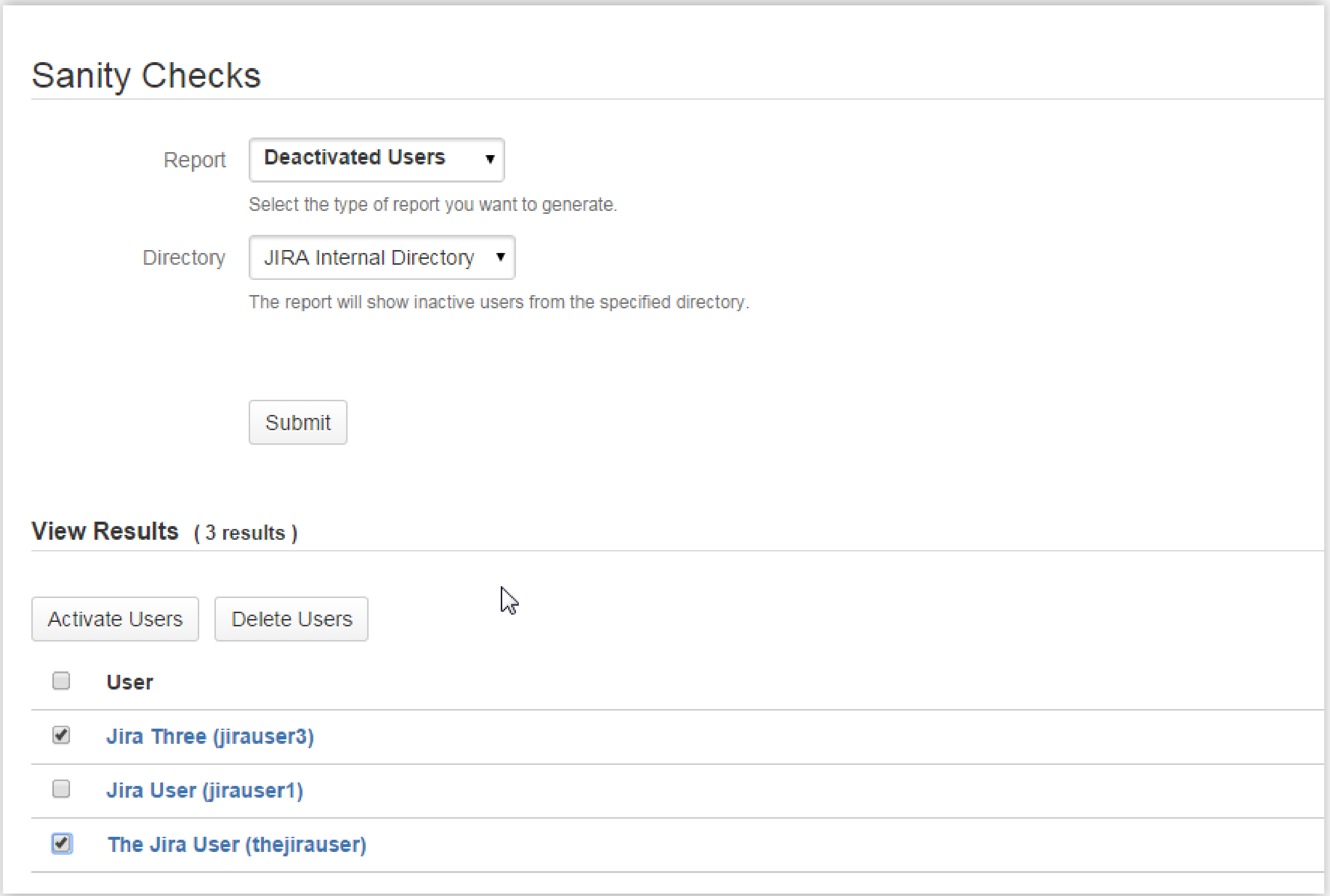
Select the type of report you want to generate.

View Results (2 results)

User	Directories
Florin Manaila (fmanaila)	Active Directory server, JIRA Internal Directory, LDAP server
Gabriel Ionita (gionita)	Active Directory server, JIRA Internal Directory

Deactivated Users

The **Deactivated Users** report shows the users who do not belong to any group nor do they have any roles on any projects. Deactivated users will not be able to log in and will not count towards your license limit. The report shows deactivated users from the specified directory.



Sanity Checks

Report: Deactivated Users

Select the type of report you want to generate.

Directory: JIRA Internal Directory

The report will show inactive users from the specified directory.

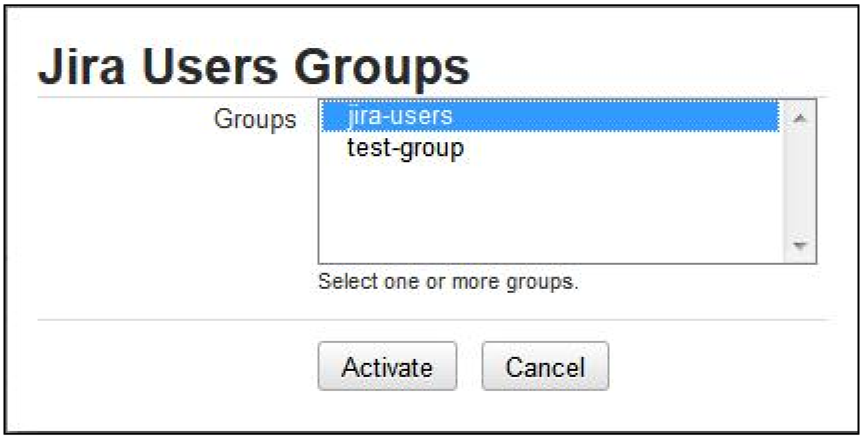
View Results (3 results)

☐	User
☑	Jira Three (jirauser3)
☐	Jira User (jirauser1)
☑	The Jira User (thejirauser)

You can **activate** (add user as member of a group - only groups who have JIRA User permission) and **delete** (remove user from JIRA) users. To do that you have to check the users you want to activate/delete.

To select all users click the checkbox from the **User** column.

To **Activate Users**, a dialog will be displayed from where you can choose the groups to add the selected users to.



Jira Users Groups

Groups: jira-users, test-group

Select one or more groups.

To **Delete Users** a dialog will be displayed to confirm the action.

> Since version 2.6.3, activating a user which has its active flag set to false just sets it back to true, preserving the user's associated groups.
>
> In case of users which have the active flag set to true, but aren't members of any of the groups associated to the following global permissions: JIRA Users, Administrators or System Administrators, they will be added to the selected groups from the JIRA Users Groups dialog.

Permission Helper

Scanning

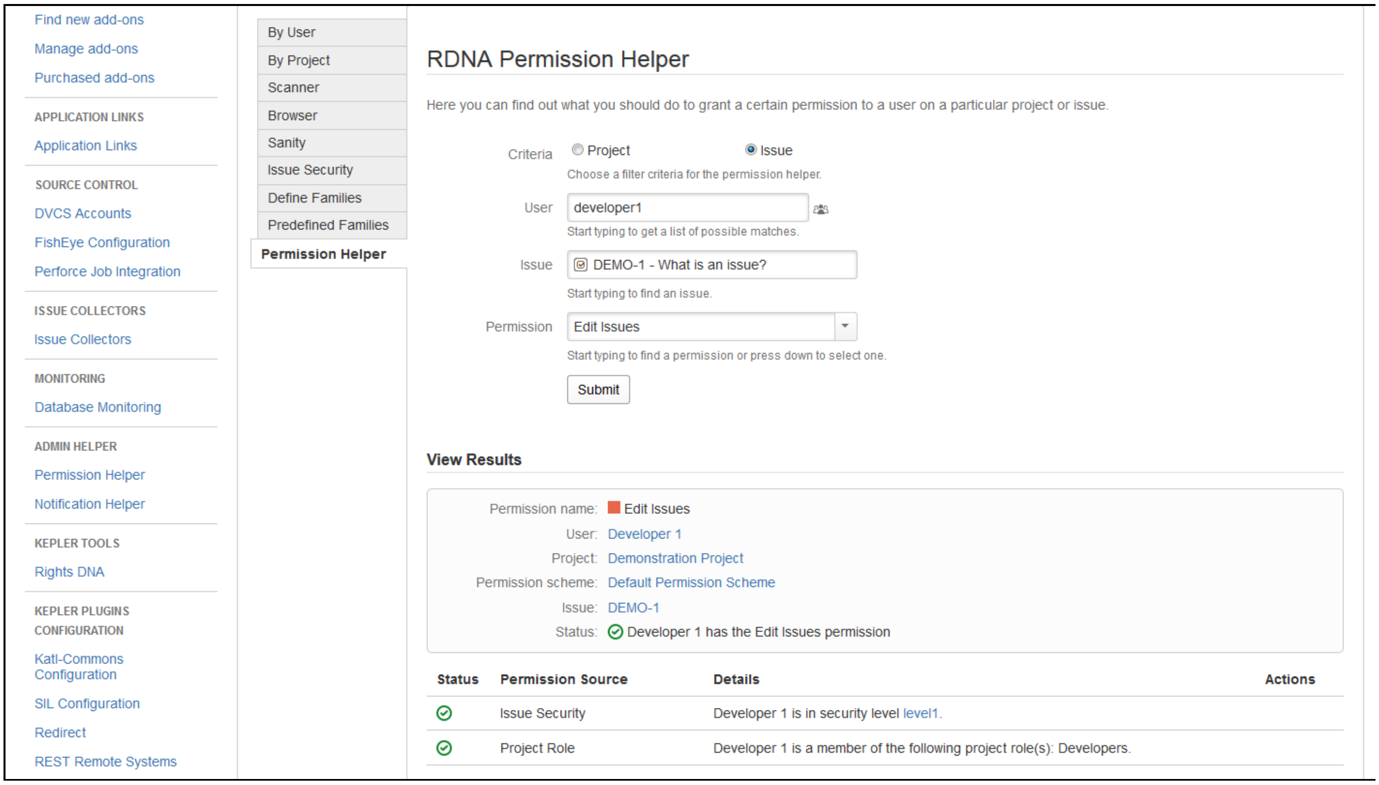
This tab gives you advices regarding what you should do to grant a certain permission to a user on a particular project or issue, or, if the user has that permission, it shows the permission sources.

You can choose a filter criteria for the permission helper, either by project or by issue.

Note that scanning a permission by project only checks non-dynamic permission sources (direct, by group, by project role or by project lead).

The dynamic permissions (eg. assignee, reporter) are only relevant in an issue context, so you can switch to this criteria and choose a particular issue for scanning.

You can select the user for which to check the permission either through autocomplete - you will get a list of possible matches after starting typing, or by using the user picker pop-up lunched from the right of the input.



RDNA Permission Helper

Here you can find out what you should do to grant a certain permission to a user on a particular project or issue.

Criteria ○ Project ● Issue
Choose a filter criteria for the permission helper.

User: developer1
Start typing to get a list of possible matches.

Issue: DEMO-1 - What is an issue?
Start typing to find an issue.

Permission: Edit Issues
Start typing to find a permission or press down to select one.

View Results

Permission name: 🟥 Edit Issues
User: Developer 1
Project: Demonstration Project
Permission scheme: Default Permission Scheme
Issue: DEMO-1
Status: ✓ Developer 1 has the Edit Issues permission

Status	Permission Source	Details	Actions
✓	Issue Security	Developer 1 is in security level level1.	
✓	Project Role	Developer 1 is a member of the following project role(s): Developers.	

After selecting your criteria (project or issue), the user and the permission to analyze you can click submit.If any of the project, issue or user are invalid, an error message will be displayed.

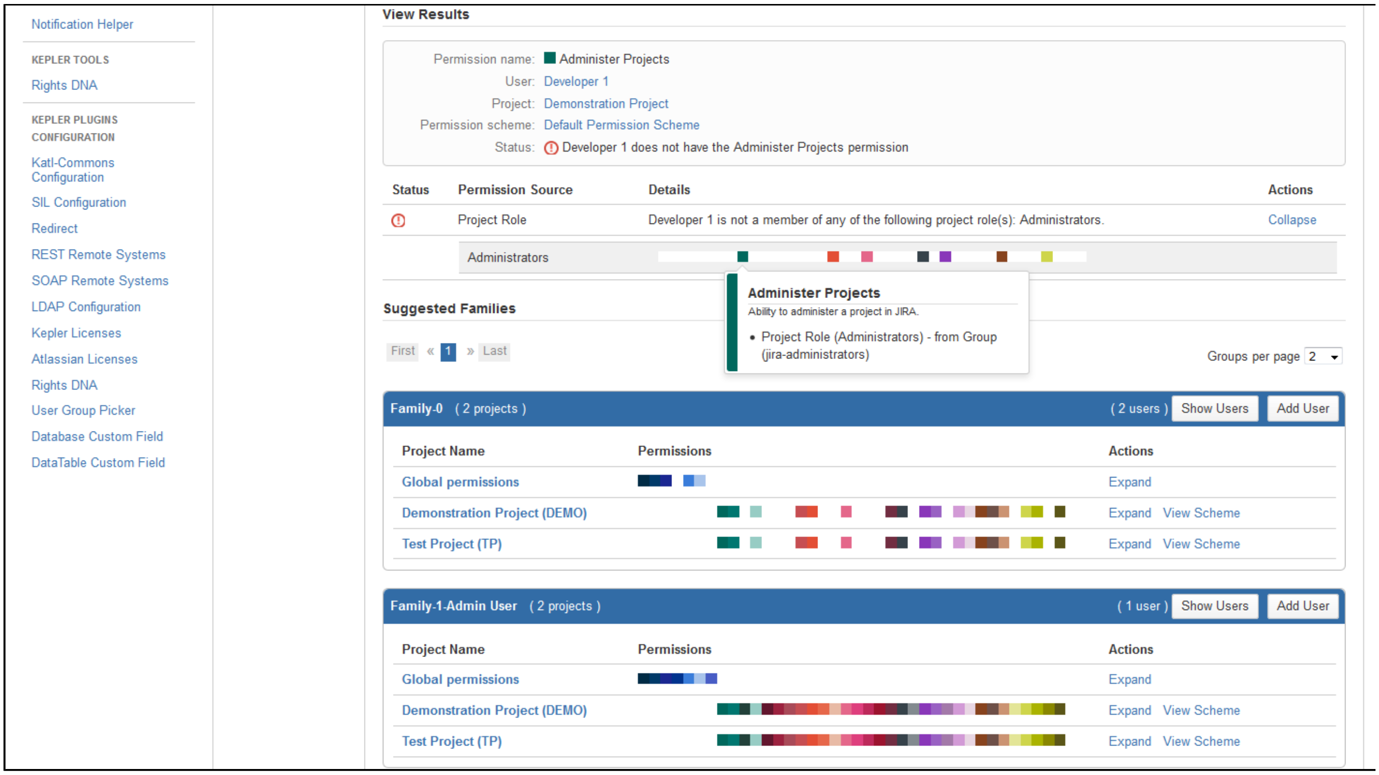
If validation passes, you will be shown the analysis result. If the user has the given permission for the given project/issue context, the permission sources will be displayed.

For the issue criteria, the issue security is also checked.

Results

If the user doesn't have the given permission for the given project / issue context, you will be shown all the failed permission sources.

For project role or group permission sources, you can expand the result and see in what groups / project roles should the user be included to be granted the permission, as well as what other permissions he will be granted by joining that groups / roles.



Notification Helper

KEPLER TOOLS

Rights DNA

KEPLER PLUGINS CONFIGURATION

Katl-Commons Configuration

SIL Configuration

Redirect

REST Remote Systems

SOAP Remote Systems

LDAP Configuration

Kepler Licenses

Atlassian Licenses

Rights DNA

User Group Picker

Database Custom Field

DataTable Custom Field

View Results

Permission name: Administer Projects
User: Developer 1
Project: Demonstration Project
Permission scheme: Default Permission Scheme
Status: Developer 1 does not have the Administer Projects permission

Status	Permission Source	Details	Actions
ⓘ	Project Role	Developer 1 is not a member of any of the following project role(s): Administrators.	Collapse
	Administrators		

Administer Projects
Ability to administer a project in JIRA.

- Project Role (Administrators) - from Group (jira-administrators)

Suggested Families

First « 1 » Last

Groups per page 2

Family-0 (2 projects) (2 users) Show Users Add User

Project Name	Permissions	Actions
Global permissions		Expand
Demonstration Project (DEMO)		Expand View Scheme
Test Project (TP)		Expand View Scheme

Family-1-Admin User (2 projects) (1 user) Show Users Add User

Project Name	Permissions	Actions
Global permissions		Expand
Demonstration Project (DEMO)		Expand View Scheme
Test Project (TP)		Expand View Scheme

You can also try the "Suggest Families" button, that will scan for all permission families that contain the searched permission on the given issue / project.

Suggested families will be ordered by the number of users from the same email domain as the searched user. Clicking on the "Show Users" button will show all the family members.

You can add the user to any of the suggested families by clicking the "Add User" button of the specific family.

You can then do a rescan of the user permission and you will see the user was granted the permission.

> Adding the user to a suggested permissions family will remove the user from ANY other permission groups he/she belongs to.

Backup and restore

At Restore: install first the plugins

Mundane operations as backup and restore may pose some problems to the unsuspecting JIRA administrator. Since all the Kepler plugins create some tables in the JIRA schema - we created this mechanism long before Active Objects was introduced into Atlassian's framework - you need to take some precautions at restore.

Specifically, at restore you need to create the tables used by our plugins. You do not need to copy schema from the previous JIRA or fill it with data, you just need to **simply install the plugins into JIRA before restoring** (enabling the plugins would create the needed tables).

Rights DNA has two dependencies:

1. katl-commons (core support)
2. warden (used for licensing)

For reference, these are the tables created by each add-on

Plugin	Tables

Rights DNA	krightsdnacustomer krightsdnapermset krightsdnapermsetusers krightsdnauserdefinedfamily
katl-commons	kplugins kpluginscfg kissuestate kstatevalues
warden	-