

Contents

1. Manual:Extension/BlueSpiceNSFileRepo	2
2. Manual:Extension/BlueSpicePermissionManager	3
3. Reference:NSFileRepo	9

Manual:Extension/BlueSpiceNSFileRepo

Contents

1	What is NSFileRepo?	3
2	How?	3
3	Related info	3

What is NSFileRepo?

NSFileRepo enables uploading files to namespaces. Extension restricts access to upload and read files and images to a given set of user groups associated with protected namespaces. Using this extension (within the security limitations noted above), you can protect not only pages and areas of your wiki, but also any uploaded images or files within those namespaces. Provide namespace based features to uploaded files.

How?

File is still being uploaded to the File namespace (NS_FILE) but it contains additional namespace that can be any of the namespaces defined on the wiki. Such files have structure:

```
[[{FILE_NS}:{Namespace}:{Filename}]]
```

For example file can be uploaded to "Private" namespace and access to that namespace can be restricted to particular user group(s):

```
[[File:Private:Filename.jpg]]
```

The standard for accessing files is generally:

```
[[File:Filename.jpg]]  
[[Image:Filename.jpg]]  
[[Media:Filename.jpg]]
```

Access to particular namespace can be managed by using [BlueSpicePermissionManager](#) extension.

Technical details on how this extension works can be found [here](#).

Related info

- [Reference:NSFileRepo](#)

Permission manager

A [quality version](#) of this page, [approved](#) on *20 November 2020*, was based off this revision.

Contents

1	Accessing the Permission manager	5
2	Role-based permissions	5
3	The roles matrix	5
3.1	Role inheritance	7
4	Default roles	7
5	Technical info	8
5.1	Logging	8
6	Configuration	8
7	Related info	8

Accessing the Permission manager

To manage permissions, you use the Permission manager. It is located under *Global actions > Management > Permission manager*. This links to the page `Special:PermissionManager`.

Role-based permissions

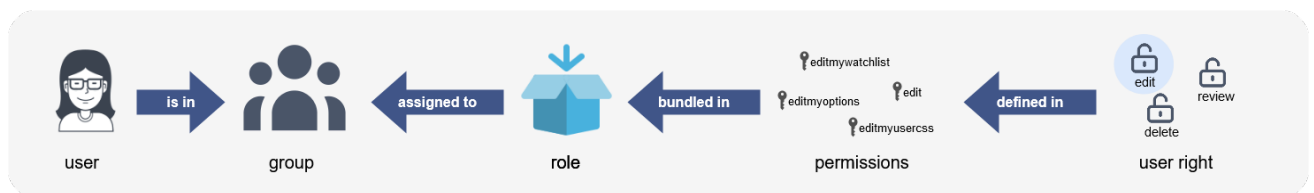
In BlueSpice 3, roles were introduced as a way to manage user rights. The main intention of using roles is to simplify rights management.

Roles represent a **collection of individual permissions** that are necessary to perform certain functions on the wiki. For example, for a user who is supposed to only read the wiki, many permissions in addition to the "read" permission are needed: The ability to change their own settings, to search the wiki, to view page ratings, and so on.



All permissions that make up a logical group are encapsulated in a role, in this example the role "reader". If wiki admins want to grant read-only rights to a user group, they only need to assign that group the "reader" role, instead of assigning many individual permissions that are needed to create a "read"-user.

By assigning roles to a group, all users belonging to that group receive the rights of these roles. Roles are never assigned directly to users, but always to groups instead. Users are then assigned to one or more groups.



The roles matrix

The permission manager consists of the group tree (1) and the role matrix (2):

Special

Permission manager

Save Reset Show system groups Export table

Role	Wiki	Namespaces					
		(Pages)	QM	Portal	Staff	Minutes	
bot	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
maintenanceadmin	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
admin	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
author	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
editor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
reviewer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
accountmanager	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
structuremanager	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
reader	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
accountsselfcreate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
commenter	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Columns > Namespaces >

- User
- User talk
- BlueSpice
- BlueSpice talk
- File
- File talk
- MediaWiki
- MediaWiki talk
- Template
- Template talk
- Help
- Help talk
- Category
- Category talk
- Property
- Property talk
- Form
- Form talk
- Concept
- Concept Talk
- Widget
- Widget talk
- Module
- Module talk
- Blog
- Book

Associating groups with roles in namespaces

The **group tree** shows all existing groups:

- **Group "*"':** all non-logged-in (anonymous) users
- **Group "user":** all logged-in users, the default group for all users
- **Subgroups of group "user":** all groups that are defined on the wiki, either by default, by MediaWiki, or custom groups created by an administrator. System groups, created by MediaWiki, can be hidden by unchecking the "Show system groups" checkbox above the tree.

The columns in the **role matrix** are:

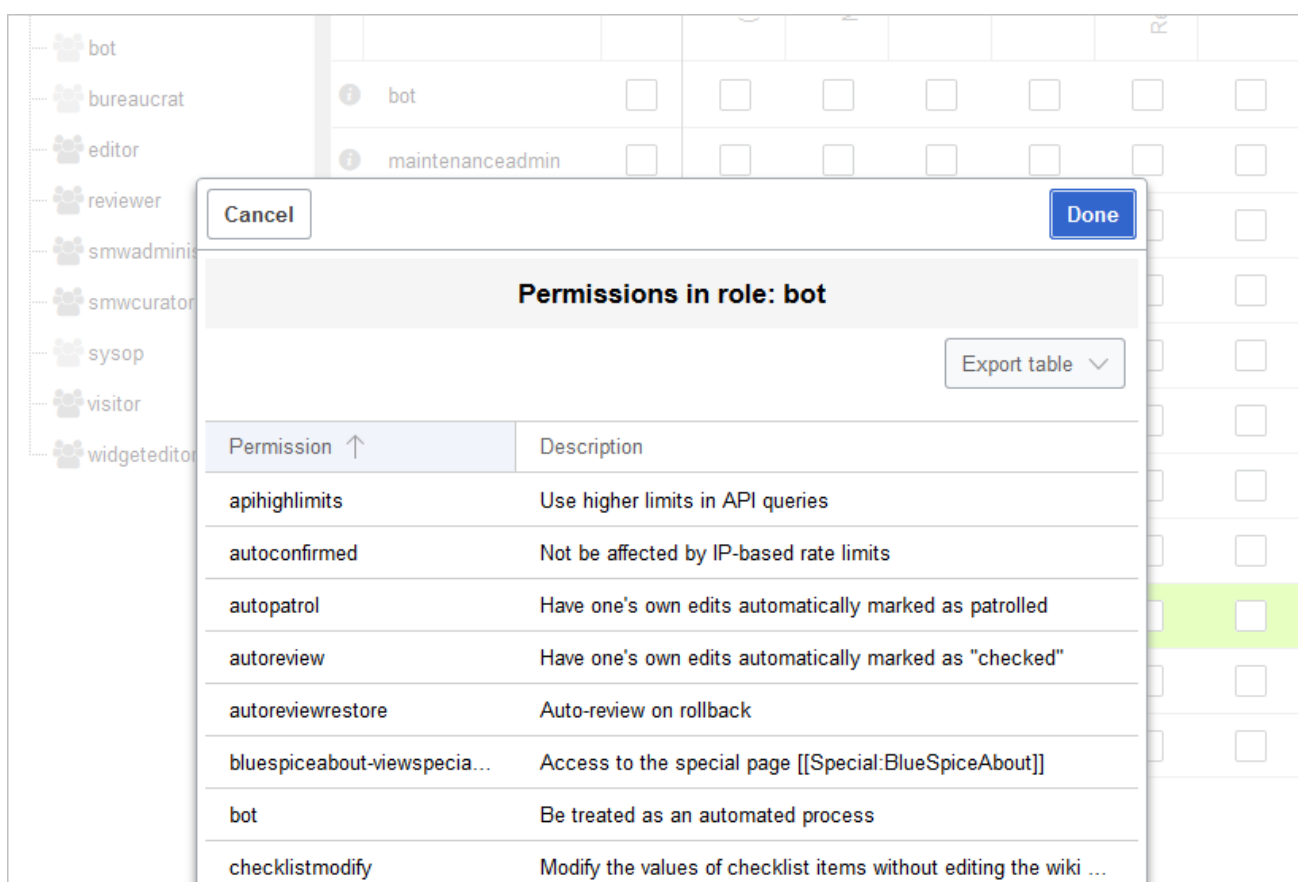
- **Role information** (info icon): Clicking the icon shows all the permissions in a role. This list is exportable.
- **Role name**
- **Wiki:** Assignment of a role to the entire wiki. By assigning the role in this column, a user group gets permissions in this role on the wiki (all namespaces).
- **Individual namespaces:** The following columns list every (applicable) namespace on the wiki.
 - Roles can be assigned to individual namespaces. For example, the group *user* can get the *editor* role only in the namespace *Public*. *Users in this group cannot edit content in any other*. By granting a role to a particular group in a particular namespace, means that all other groups will lose permissions from this role, eg. granting role "reader" in namespace "Private" to group "sysop" means that all users in any other groups won't be able to read pages in "Private" namespace, even if they have "reader" role granted on the wiki level ("Wiki" column).
 - The same role can be granted to multiple groups for the same namespace.
 - Additional namespaces can be added in the matrix by clicking on the arrow in table header, then "Columns". Then the namespaces can be selected.

Role inheritance

By default, all roles granted to the (*) group will be granted to the *user* group, and all roles granted to the *user* group are granted to its subgroups. If a group inherits the role from an upper-level group field, this is indicated in the role matrix with a green background, but the checkbox is empty.

Default roles

By default, the Permission manager includes a number of predefined roles that serve most user needs. The individual permissions contained in a role can be seen by clicking the info icon in front of the role name. It opens a dialog with a permissions list for the role.



The screenshot shows a dialog box titled "Permissions in role: bot" overlaid on a permission matrix. The dialog has "Cancel" and "Done" buttons. It contains a table with the following permissions and descriptions:

Permission ↑	Description
apihighlimits	Use higher limits in API queries
autoconfirmed	Not be affected by IP-based rate limits
autopatrol	Have one's own edits automatically marked as patrolled
autoreview	Have one's own edits automatically marked as "checked"
autoreviewrestore	Auto-review on rollback
bluespiceabout-viewspecia...	Access to the special page [[Special:BlueSpiceAbout]]
bot	Be treated as an automated process
checklistmodify	Modify the values of checklist items without editing the wiki ...

- **bot:** exists to achieve recurring system actions. This role is assigned to the user BSMaintenance in Bluespice via the group bot. The group bot should not be changed.
- **admin:** Grants access to all administrative special pages and to all typical administrative features.
- **maintenanceadmin:** Similar to the *admin* role, but with extended admin rights for maintaining wiki integrity.
- **author:** all permissions necessary for creating content on the wiki. Editing, moving, or deleting pages is not possible.
- **editor:** create content, edit and delete content.

- **reviewer:** If you have activated the review function and, therefore, work draft pages in a namespace, there must be at least one group with the role of reviewer. By default, the group “reviewer” is available for this purpose. Only users in the reviewer role can approve draft pages. Reviewers generally need read, write and review rights via the corresponding three roles of reader, editor and reviewer. However, if you have not activated the review function in any namespace, you do not need this role in your wiki.
- **accountmanager:** enables the administration of user accounts. Since user accounts are managed independently of namespaces in the wiki, this role cannot be restricted to individual namespaces. Grayed-out namespaces have no meaning here as long as the role in the wiki itself is highlighted in green.
- **structuremanager:** allows some actions for wiki maintenance such as moving pages, mass deleting pages or searching and replacing text, as well as renaming namespaces.
- **accountselfcreate:** enables the automatic creation of new user accounts and is required for single-sign-on. You can assign this role, for example, to anonymous users who can create their own account.
- **commenter:** allows the creation of discussion contributions and page ratings, but not of the pages themselves. The editor role includes all the rights of the commenter role. If a group has editor rights, it does not need special commenter rights.
- **reader:** Basic read access. Users can also edit their personal settings.

Important! The default roles and related permissions are different in the [BlueSpice pro Cloud permission manager](#).

Technical info

Logging

Every change to the roles is logged in `Special:Log`, in the `Permission Manager log`. These logs are available only to wiki administrators (users in groups with the role *admin*).

Configuration

All changes to the role matrix are backed up. By default, the last 5 backups are kept. This limit can be changed in [Config manager](#), under extension `BlueSpicePermissionManager`.

- **Backup limit:** Sets the number of backups for the permissions manager. Each time the page *Special:PermissionManager* is saved, a backup is created. If the backup limit is set to 5, the last five versions of the permissions configuration are saved as backups.

Related info

- [Reference:PermissionManager](#)
- [Managing groups](#)
- [Rights concepts](#)

Reference:NSFileRepo

Extension\ : NSFileRepo

Overview			
Description:	implements per-namespace group permissions for image and file rights protection.		
State:	stable	Dependency:	BlueSpice
Developer:	HalloWelt	License:	GPL v2+
Type:	BlueSpice	Category:	Document Management
Edition:	BlueSpice pro, BlueSpice Farm, BlueSpice Cloud		
View help page			

Features

NSFileRepo enables uploading files to namespaces. The extension restricts access to upload and read files and images to a given set of user groups associated with the protected namespaces.

Usage / features

Using this extension (within the security limitations noted above), you can protect not only pages and areas of your wiki, but also any uploaded images or files within those namespaces. Use the same syntax as a normal file reference link, adding the namespace between the file specified ("File", "Image" or "Media") and the file name.

Technical information

Configuration

[Image Authorization](#) must be configured properly.

Rights (permissions)

This extension does not require any special rights. If users will be allowed to access files in a particular namespace depends on the "read" permission for that namespace.

Requirements

NSFileRepo requires [BlueSpiceFoundation](#).

More info

For detailed description and usage, visit our [help page](#).

Technical details on how this extension works can be found on [MediaWiki](#).

Demo

Try out associating files such as images with namespaces in our [BlueSpiceDemo](#).