

Contents

1. UserManual:Extension/FlaggedRevs .....	2
2. Manual:Extension/BlueSpiceReview .....	11
3. Manual:Quality management .....	19
4. Manual:Semantic MediaWiki .....	27
5. Reference:BlueSpiceFlaggedRevsConnector .....	33
6. Reference:FlaggedRevs .....	36
7. Using the Version History .....	37

## 7.6 UserManual:Extension/FlaggedRevs

The extension **FlaggedRevs** is a MediaWiki extension that activates an approval mechanism. New versions of a page are initially marked as drafts and remain so until a user with *review* rights approves the page.

### Contents

1 What are page drafts? .....	3
2 Activating the acceptance feature .....	3
2.1 Existing pages .....	3
2.2 New pages .....	4
3 Accepting a page .....	4
4 Setting read permissions for drafts .....	6
5 Changes to embedded templates and files .....	8
5.1 Approving a resource .....	9
6 Settings for resources on approved pages .....	9
7 Effects on Semantic MediaWiki properties .....	10
8 Quality management overview .....	11
9 Related info .....	11

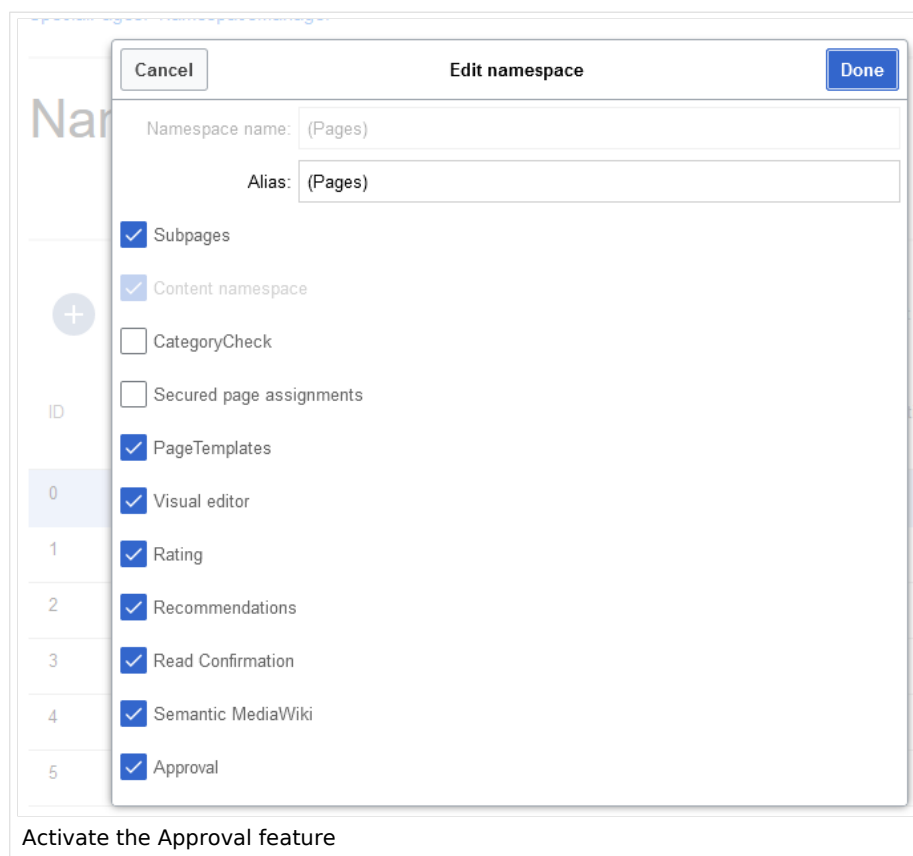
## What are page drafts?

In BlueSpice, you can create draft versions of a page. After a page has been edited and saved in draft mode readers of the page will, by default, see the most recently accepted version of the page. This allows editors to work on a draft version until all changes "fit". The changes can then be approved by users with reviewer rights.

By default, all users can see the latest draft version by clicking on the status information of a page or by switching to the [version history](#). It is possible to limit view permissions of drafts to specific user groups.

## Activating the acceptance feature

Activating the acceptance feature is done on the namespace level. A user with admin permissions activates the feature in *Global actions > Management > Namespace manager* for individual namespaces. The dialog window opens after clicking the edit icon (wrench) in the *Actions* column. Select *Approval* and click *Done*.



## Existing pages

After activating *Approvals* for a namespace, all pages in the namespace are set into *first draft* mode.

The screenshot shows a MediaWiki page titled "Working with drafts". At the top, a purple banner notice (1) states "There are no accepted versions of this page." Below the banner, the page status bar (2) shows "Test: Entwurfsseite", "Page" and "Discussion" links, "No categories assigned", and "Last edited 1 second ago by WikiSysop". The page content is "In progress." and "This page is first draft".

Draft banner notice (1) and page status information in the status bar (2).

Each page in the namespace now displays a banner at the top of the page with the acceptance status (1) and a quick link to publish the page (2).

## New pages

When a new page is created and first saved, it will automatically be visible to all users. Subsequent page edits will be collected in the draft version of the page. If a page is edited and saved multiple times while in draft mode, each edit gets logged in the version history:

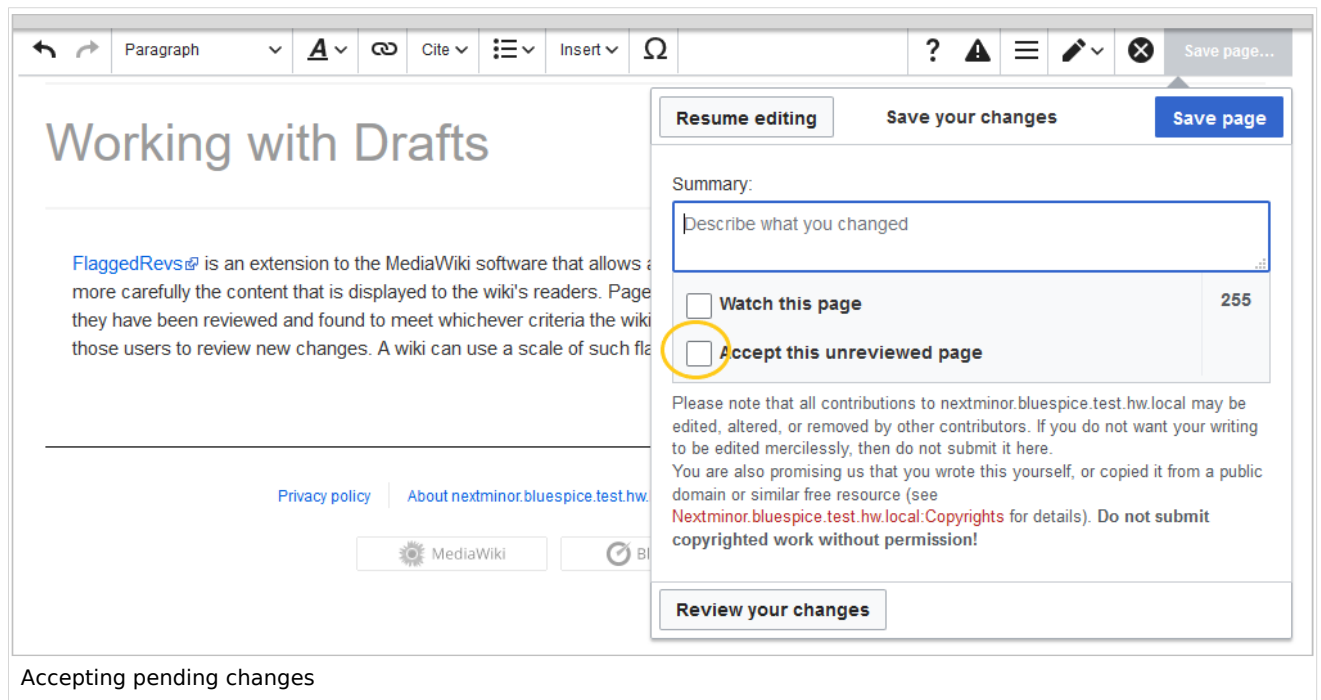
The screenshot shows the draft revisions list for the page "Working with drafts". It lists four revisions, all marked as "pending review". The first two revisions are highlighted in orange and have a dotted border. The first revision is from 14:04 on 4 November 2019 by WikiSysop, with a size of 1,332 bytes and a change of +486. The second revision is from 14:03 on 4 November 2019 by WikiSysop, with a size of 846 bytes and a change of +213. The third revision is from 13:36 on 4 November 2019 by WikiSysop, with a size of 633 bytes and a change of +38. The fourth revision is from 12:45 on 4 November 2019 by WikiSysop, with a size of 595 bytes and a change of +595. A "Compare selected revisions" button is visible at the bottom.

Draft revisions

The revisions in draft status have a dotted border and are highlighted in orange. In addition, they are marked as *pending review*. These revisions can be compared before the pending changes of the page are accepted.

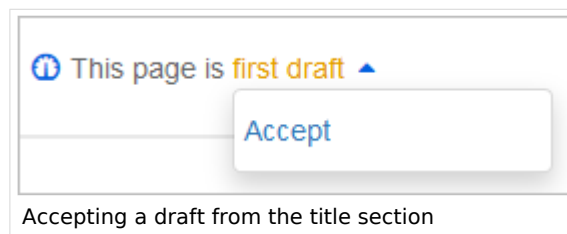
## Accepting a page

Users with *review* rights can accept a draft page any time they save a page (*Accept this unreviewed page*):

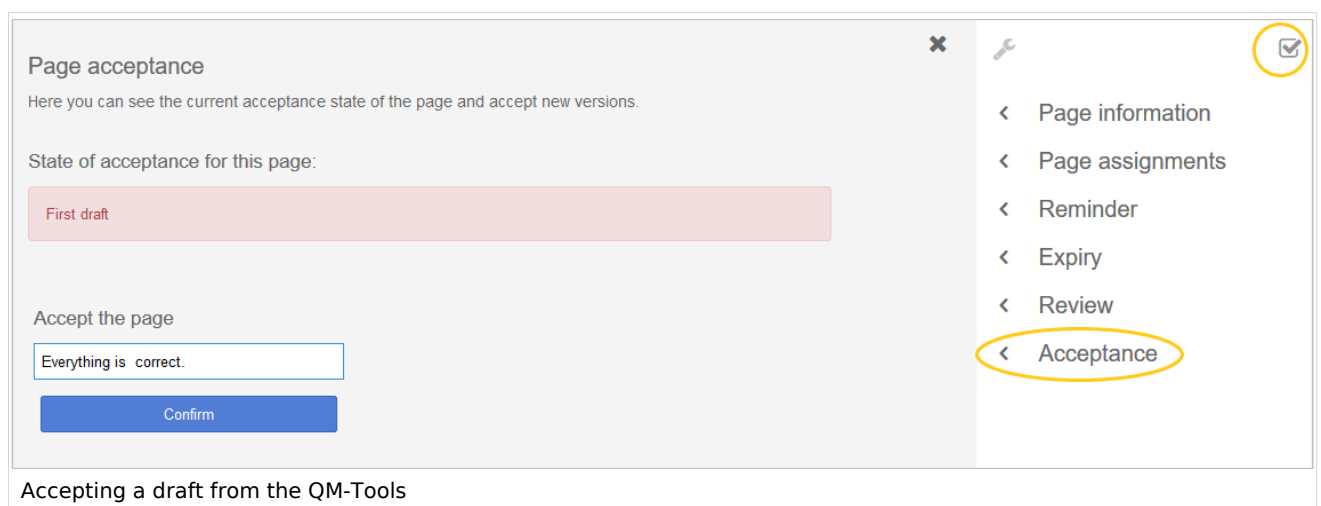


If the page is not accepted at this point, it can be accepted independently from the editing process.

A user with *review* rights can accept the page quickly with the review feature in the title section of the page. The draft message will display an arrow which includes the *accept* function:

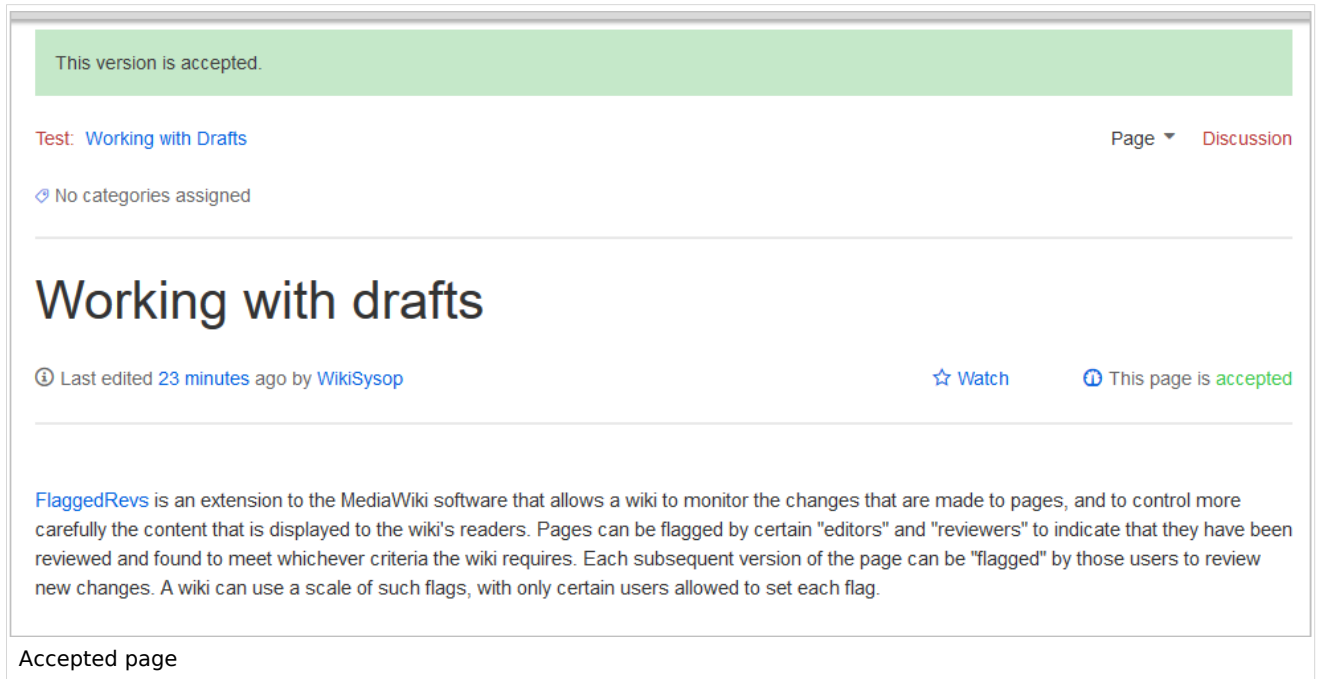


Alternatively, the reviewer can go to the Acceptance page in the Quality Management toolbar instead:



In both cases, the reviewer can leave a comment. This comment will then be shown in the version history of the page.

The page now shows the page version as accepted.



The screenshot shows a MediaWiki page interface. At the top, a green banner states "This version is accepted." Below this, the page title "Working with drafts" is displayed. To the right of the title, there are links for "Page" and "Discussion". Below the title, it says "No categories assigned". Further down, it indicates "Last edited 23 minutes ago by WikiSysop". To the right of this, there are links for "Watch" and "This page is accepted". The main content area contains a paragraph about the "FlaggedRevs" extension. At the bottom of the page, it says "Accepted page".

This version is accepted.

Test: [Working with Drafts](#) Page ▾ Discussion

[No categories assigned](#)

## Working with drafts

📄 Last edited [23 minutes](#) ago by [WikiSysop](#) ☆ Watch 📄 This page is [accepted](#)

[FlaggedRevs](#) is an extension to the MediaWiki software that allows a wiki to monitor the changes that are made to pages, and to control more carefully the content that is displayed to the wiki's readers. Pages can be flagged by certain "editors" and "reviewers" to indicate that they have been reviewed and found to meet whichever criteria the wiki requires. Each subsequent version of the page can be "flagged" by those users to review new changes. A wiki can use a scale of such flags, with only certain users allowed to set each flag.

Accepted page

## Setting read permissions for drafts

By default, all users can see the drafts of a page, even readers.

To hide drafts from readers, the group *user* has to be removed from the configuration. An administrator deletes the group user from "Show unstable pages for the following groups" in *Global actions > Management > Config manager*.

## Config Manager

Feature ▼ |  |

<input type="checkbox"/> Export	BlueSpiceChecklist
<input type="checkbox"/> Skinning	<input checked="" type="checkbox"/> Mark checklist changes as minor edit
<input type="checkbox"/> System	
<input type="checkbox"/> Searching and Navigation	BlueSpiceExpiry
<input type="checkbox"/> Data analysis	<input checked="" type="checkbox"/> Show watermark for expired pages
<input type="checkbox"/> Personalization	
<input type="checkbox"/> Book	BlueSpiceFlaggedRevsConnector
<input type="checkbox"/> Content structuring	<input checked="" type="checkbox"/> Automatically set the page to stable, when the workflow was finished successfully
<input checked="" type="checkbox"/> Quality assurance	<input checked="" type="checkbox"/> Automatically delete the workflow, when a page was marked as stable
<input type="checkbox"/> Administration	Show unstable pages for following groups
<input type="checkbox"/> Editor	<input type="text" value="user X"/>

**Warning:** If the group *user* is removed from the *BlueSpiceFlaggedRevs* configuration, users in this group will not be able to see draft versions when they are added as reviewers in a review process.

When the group *user* is removed, only *sysop* and *reviewer* users can access drafts. If you want other groups, for example *editor*, to see draft versions, you need to add them to the configuration and save it:

BlueSpiceFlaggedRevsConnector

☒ Automatically set the page to stable, when the workflow was finished successfully

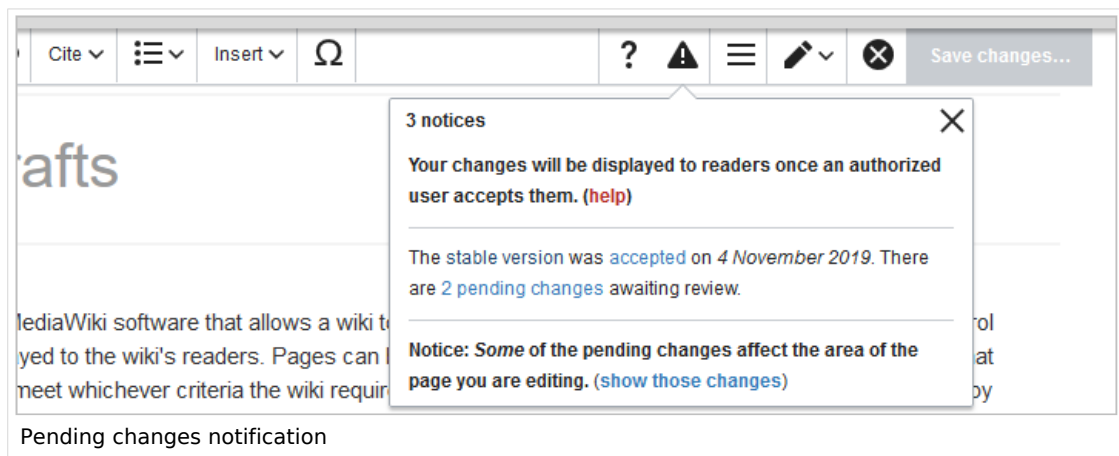
☒ Automatically delete the workflow, when a page was marked as stable

Show unstable pages for following groups

The two checkboxes *Automatically set the page to stable...* (checkbox is available until version 3.1.) and *Automatically delete the workflow...* determine what happens at the end of a workflow, if there is an active [Review](#) for a page. A page can be set to accepted status by a workflow and the workflow can be deleted after the page is marked as stable.

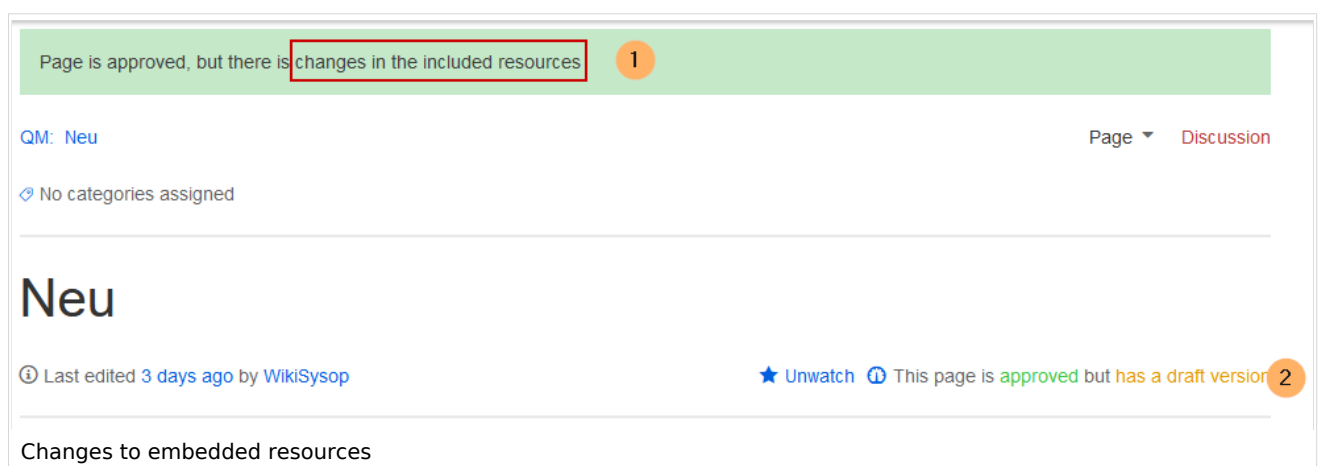
**Note:** If no user groups are specified in the configuration field, not even reviewers and sysops users will see a visual status indication of a page. They will have access to drafts through the revision history of a page instead.

Anyone with edit permissions will receive a notification that there are pending changes for the page when they access the page in edit mode:



## Changes to embedded templates and files

If a page includes templates or files such as images or PDFs and there is a change to these files after the page has been approved, this will also be reflected in the status banner (1) and the QM status information in the title area (2).





If the embedded resources are located in a namespace without its own approval function, the version of the resource at the time of page approval is displayed in the approved version of the page. The draft version accordingly shows the most current version of the embedded resource. To approve the resource change, the page has to be approved again. This creates a new page version.

## Approving a resource

Before approving the changed resource(s), you should check which resources are affected.

1. **Open** the quality management tools (1). Here you can see under Approval that there are changes to resources (2).
2. **Click** on a changed resource (3) to display the changes between the last two versions of the resource.
3. **Click** "*Review now*" (4) under Unapproved changes. This will take you to the draft version of the page. From there you can approve the page in the title area. **Note:** If there are several changed resources, they can only be approved together.

Page approval

Here you can see the current approval state of the page and approve new versions.

State of approval for this page:

Approved

Pending changes

There are pending changes waiting for approval. View now. 4

Changes to included resources since the last page approval

- File:QM:pexels-martin-damboldt-814499.jpg 3

Approving a ressource

## Settings for resources on approved pages

It is possible via server configuration to show embedded resources (i.e., files, templates or transcluded content) that were changed after the page has been released, in a certain version. The following options exist:

Setting	Description
	Shows the resource version that was valid when the page was approved.

Setting	Description
FR_INCLUDES_FREEZE	<div><b>Warning:</b> The setting FR_INCLUDES_FREEZE for handling the state of included files of an approved wiki page has <a href="#">no long term support by</a> and should not be introduced in a new wiki for that reason.</div>
FR_INCLUDES_STABLE	Always shows the latest approved (stable) version of a resource, even if it was approved after the page was approved (this works only if the resource is located in a namespace with active approval feature).
FR_INCLUDES_CURRENT	Always shows the latest version of the included resource.

**Important!** Results of a **semantic query**, unlike attribute values **cannot** be "frozen", even if FR\_INCLUDES\_FREEZE has been set. The query results do not persist in the source text of a page, but are generated dynamically at the time of the query. As a consequence, they are not available for versioning.

## Effects on Semantic MediaWiki properties

Wenn in einem Namensraum gleichzeitig mit der Seitenfreigabe auch [Semantic MediaWiki](#) aktiviert ist, sind je nach Freigabestatus der Seite folgende Attribute für SMW-Abfragen verfügbar:

Document state	Available properties
First draft	Current values
Approved	Values of the approved (stable) version
Draft	Values of the approved (stable) version (default setting)
Implicit draft <sup>[1]</sup>	Values of the approved (stable) version

The availability of the semantic properties is determined by the following configuration setting:

```
$bmsgFlaggedRevsConnectorStabilizeSMWPropertyValues = true;
```

The value for this setting can be set to *false* in the *LocalSettings.php* of the server file system. If set to *false*, the property values of the draft version are used for semantic queries instead of the values for the stable version.

## Quality management overview

The Special page *Special:Quality\_management\_overview* allows users with *admin* or *reviewer* rights to view and filter a list of wiki pages and their review status.

### Quality management overview

Export table ▾

Page title	State ↑	Page categories
<a href="#">Manual:What is a wiki</a>	Approved	
<a href="#">Manual:Create pages</a>	Approved	
<a href="#">Manual:Subpage</a>	Approved	
<a href="#">Setup:System requirements</a>	Approved	
<a href="#">Setup:Release Notes</a>	Approved	
<a href="#">Setup:Download</a>	Approved	
<a href="#">Setup:Release History</a>	Approved	
<a href="#">Setup:Portal</a>	Approved	
<a href="#">Manual:Getting started - Login and configuration</a>	Approved	<a href="#">Manual</a>
<a href="#">Manual:Using the Version History</a>	Approved	

Sort Ascending  
Sort Descending  
Columns  
Filters  
Setup

☒ First draft  
☒ Draft  
☒ Approved  
☐ Not enabled  
☒ Implicit draft

Special page Quality management overview

## Related info

- [Quality management tools](#)
- [Reference:FlaggedRevs](#)
- [Reference:BlueSpiceFlaggedRevsConnector](#)
- [Manual:Semantic MediaWiki](#)

- [↑](#) An approved page has an implicit draft, if there have been changes to embedded resources (e.g., a template or an image) since the last page approval.

## Create a review

A [quality version](#) of this page, [approved](#) on *24 November 2021*, was based off this revision.

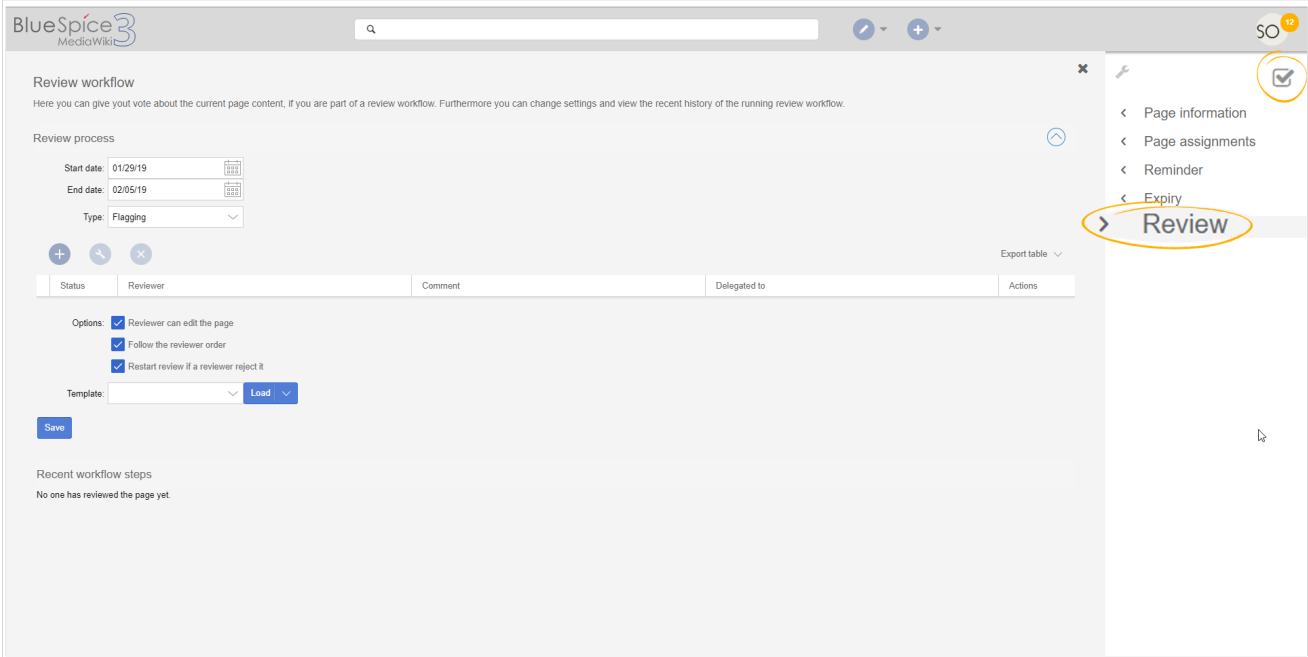
**BlueSpiceReview** is a workflow tool for MediaWiki. It allows assigning users to a page who then should complete their review within a certain timeframe.

## Contents

1 The review page .....	13
2 Starting a review .....	13
2.1 Managing templates .....	14
2.1.1 In BlueSpice 3.2 .....	14
2.1.2 In BlueSpice 3.1 .....	16
2.2 Applying a template .....	17
2.3 Deleting a template .....	17
3 Casting votes .....	17
3.1 Rejecting a task .....	18
4 Review overview .....	18
5 Configuration .....	19
6 Related info .....	19

## The review page

All relevant info can be accessed from the *Review* flyout in the *Quality management* tools. This screen shows the review settings. If a review is running, users see a form to cast their votes. They can also expand the collapsed workflow settings to see the details for this review.



The review workflow

Here you can give your vote about the current page content, if you are part of a review workflow. Furthermore you can change settings and view the recent history of the running review workflow.

Review process

Start date: 01/29/19  
End date: 02/05/19  
Type: Flagging

Options: ☒ Reviewer can edit the page  
☒ Follow the reviewer order  
☒ Restart review if a reviewer rejects it

Template:  Load

Save

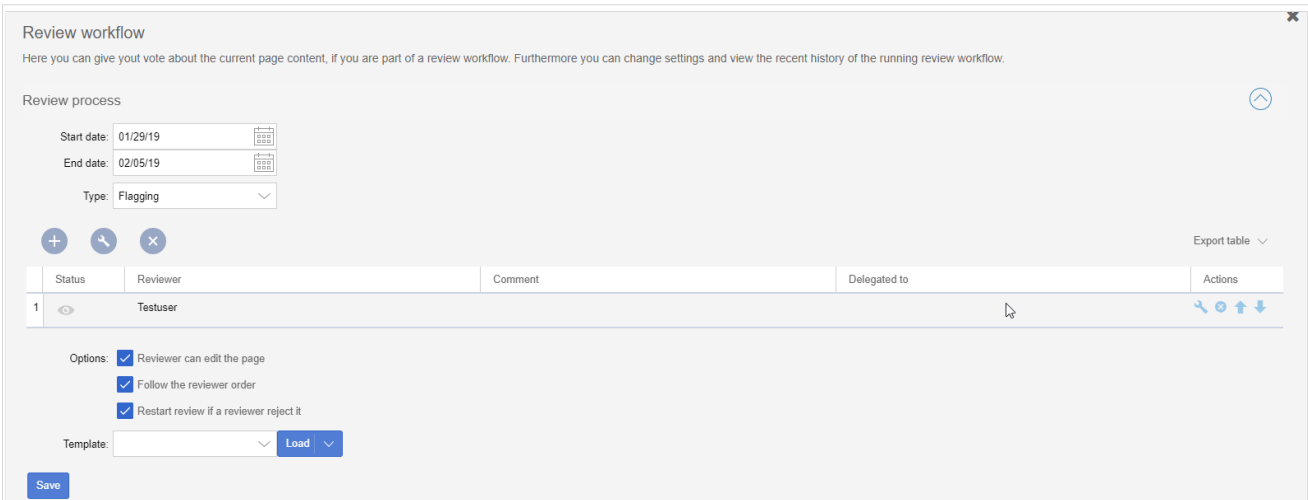
Recent workflow steps

No one has reviewed the page yet

The review workflow

## Starting a review

All users with sufficient permissions can start (or edit) a review on a page.



Review workflow

Here you can give your vote about the current page content, if you are part of a review workflow. Furthermore you can change settings and view the recent history of the running review workflow.

Review process

Start date: 01/29/19  
End date: 02/05/19  
Type: Flagging

Options: ☒ Reviewer can edit the page  
☒ Follow the reviewer order  
☒ Restart review if a reviewer rejects it

Template:  Load

Save

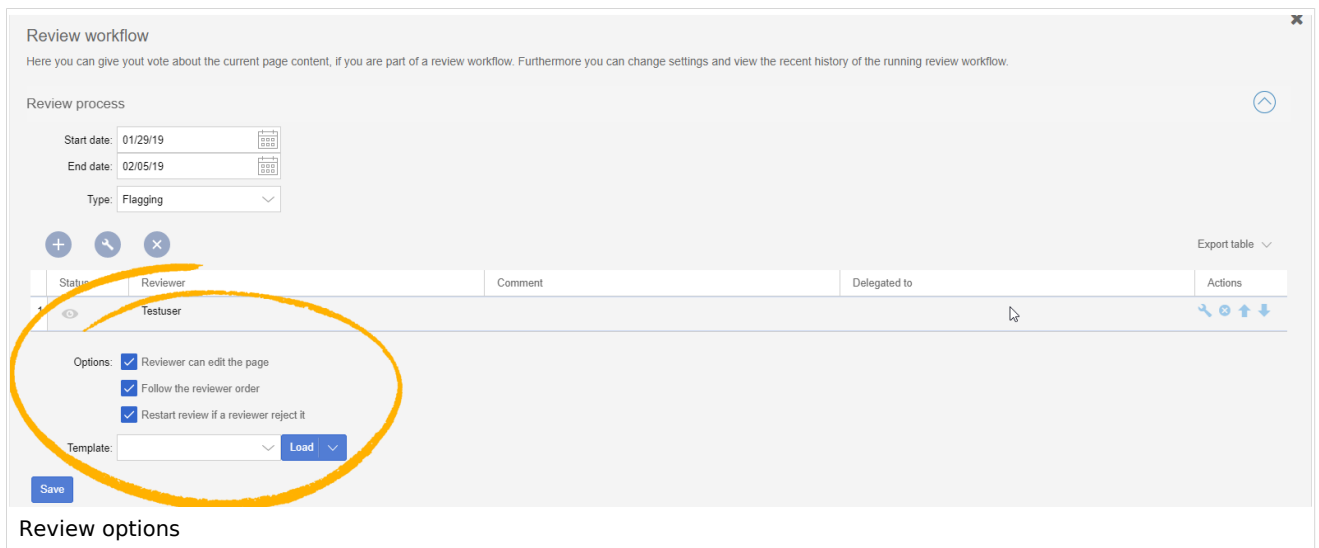
Review settings

Possible settings:

- **Start date:** The date when the review officially begins. Please note that you cannot schedule a review in advance. The page is locked for editing as soon as the workflow is saved.
- **End date:** The date on which the review is listed as expired. It has no effect on the review itself and is simply used for managing workflow lists.
- **Type:** The review type determines what will happen to the page once the review has been successfully completed. The default type is *Approval* so that it can be used in conjunction with the Approval feature if this feature is activated in a namespace. BlueSpice 3.1 only ships with this one type. BlueSpice 3.2 has a *Default* and an *Approval* option which is explained in the steps to set up a workflow template.
- **Reviewers grid:** A list of all reviewers assigned to review a particular page. Reviewers can be added by clicking on the "Plus" button above the grid and by typing in the name of the user in the dialog that pops up. Existing reviewers can be removed by selecting a reviewer and clicking on the "x" button.

When you set up a new workflow, the following options are available:

- **Reviewer can edit the page:** Everybody that is assigned as reviewer can edit the page while the review is active. Other users cannot edit the page during this time. If this box is not checked, then reviewers will also not be allowed to make changes to the page.
- **Follow the reviewer order:** Reviewers need to cast their votes in sequence. Only when one is finished casting a vote, the next reviewer can start. The order of the reviewers can be adjusted by clicking on up and down arrows that appear when hovering over the reviewer in the grid.
- **Restart review if a reviewer rejects it:** If any of the reviewers rejects the page, the review is restarted and all of the reviewers must cast their votes again.



Review workflow

Here you can give your vote about the current page content, if you are part of a review workflow. Furthermore you can change settings and view the recent history of the running review workflow.

Review process

Start date: 01/29/19

End date: 02/05/19

Type: Flagging

Options:

- ☒ Reviewer can edit the page
- ☒ Follow the reviewer order
- ☒ Restart review if a reviewer reject it

Template:  Load

Save

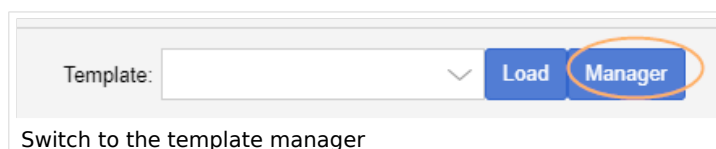
Review options

To minimize the effort when starting a review, templates are introduced.

## Managing templates

### In BlueSpice 3.2

Click Manager next to the template select menu if you are already setting up a review.



Template:  Load Manager

Switch to the template manager

Alternatively, you can create a workflow template by going to *Global actions > Management > Review template manager* in the navigation panel of the wiki.

## Review template manager

**Public review templates** 1 | **Private review templates** 2

+ 3 ⚙️ × Export table ▼

Type	Name <span>↑</span>	Namespaces	Categories	Actions
default	QA Check ISO9001			<span>4</span> <span>⚙️</span> <span>×</span> <span>Edit</span>
default	QA Research and Development	DEV	RE	
flagging	QA Technology			

⏪ ⏩ | Page 1 of 1 | ⏴ ⏵ 🔄 Displaying 1 - 3 of 3

Managing review templates in BlueSpice 3.2

Create a new template:

1. Select if you want to create a template that is available to all users (1) or that is available only to you (2).
2. Click the "+"-button (3). This opens the template dialog. Create your template.

Cancel Edit review template Done

Name:

Type: Approval ▼

Namespaces: QM × ▼

Categories: Audit × ▼ 👤

Options: ☒ Reviewer can edit the page  
☒ Follow the reviewer order  
☒ Restart review if a reviewer reject it

+ ▼ ⚙️ × Export table ▼

	Reviewer	Comment	Actions
1	<span>👤</span> Florian Bäckmann		
2	<span>👤</span> Renate Memminger		
3	<span>👤</span> Reviewers (reviewer)		

Creating a review template

3. Enter a name.
4. Select the template type:
  1. **Default:** The review is independent of settings in the Config manager. For reviews in **namespaces with approvals**, the page is not automatically approved at the end of the review.
  2. **Approval:** The review applies the [setting from the Config manager](#) for automatically deleting the workflow after the page has been approved. Pages in namespaces that do not create draft versions are not affected by this setting.
5. If the template should only appear on pages in certain namespaces, specify the appropriate namespaces.
6. If the template should only appear on pages that are assigned to certain categories, specify the categories.
7. Select the review options described above.
8. Use the "+"-button to add the reviewers. If you enter a comment here, it will be displayed in the version history. **Note:** A group vote can also be set up for an individual workflow step using the plus button. For this purpose, a wiki group is selected as the reviewer.

Cancel Edit step Done

Group: Reviewers (reviewer) ▾

Comment: Check for ISO9001 criteria

Required approvals: 2 + − User(s) ▾

Required rejections: + − User(s) ▾

Group vote

1. *Group:* The group that will vote on the selected step.
  2. *Comment:* Comment that is displayed in the workflow and in the version history for this step.
  3. *Required approvals:* If this number is reached first, the process step will be completed positively.
  4. *Required rejections:* If this number is reached first, the process step will be completed negatively.
9. Click *Done*.

The template can now be selected accordingly in a review.

**Note:** The review type can be changed after loading the template and then behaves according to the newly selected type. This means that the "Approval" type from a template can be set to be a "Standard" workflow on a page. In that case, the draft version of the page will not be approved after the workflow has finished.

## In BlueSpice 3.1

Once the list of reviewers is entered and the other options have been set, enter the name for the template in the input field and click on the arrow next to the "Load" button. Several options are available:

- *Save for myself:* The template will be shown only to the user who saves it
- *Save for everybody:* the template will be shown to everybody

Pick the desired option and the template will be saved.



Review workflow

Here you can give your vote about the current page content, if you are part of a review workflow. Furthermore you can change settings and view the recent history of the running review workflow.

Review process

Start date: 01/29/19

End date: 02/05/19

Type: Flagging

Status	Reviewer	Comment	Delegated to	Actions
1	Testuser	Testuser: My comment		

Options:

- ☒ Reviewer can edit the page
- ☒ Follow the reviewer order
- ☒ Restart review if a reviewer reject it

Template: Test Load

Save Delete

Save for myself

Save for everybody

Delete

Creating a new template in BlueSpice 3.1

## Applying a template

Once there are templates saved, users can choose from the list of templates. After clicking the "Load" button, the reviewers that have been saved in this template are added to the reviewer list.

In BlueSpice 3.1, only the list of reviewers is stored in the template.

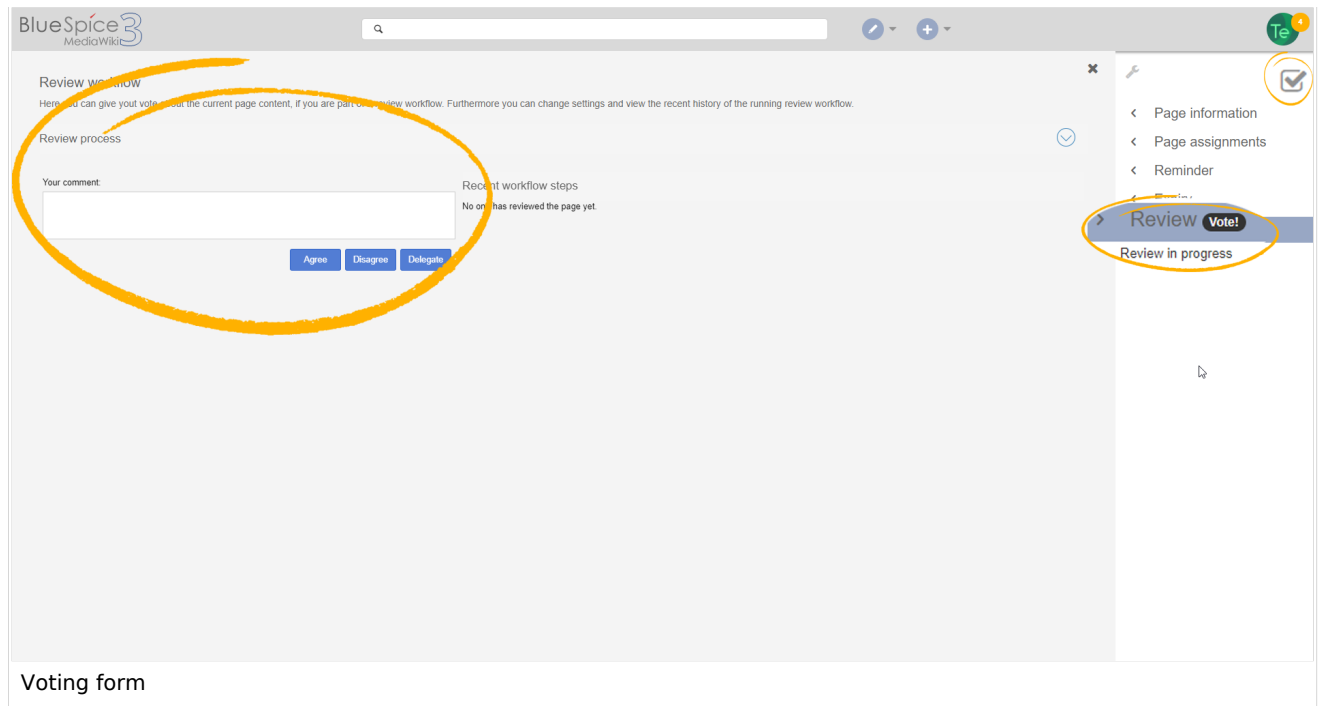
## Deleting a template

To delete a template, choose the template to delete, click on the arrow in the "Load" button and select "Delete".

A review can only be started if at least one review has been added. Once all settings are set, click save to start the review. At this point, all assigned reviewers receive a notification (depending on their notification settings) that there is a review pending, and that they need to review it. During the review process, editing of the page is disabled unless the setting to allow editing has been activated.

## Casting votes

When assigned reviewers visit that page, they are prompted to cast their vote.



The voting form consists of:

- Comment box - here reviewers can explain the reason for their vote
- Agree button - by clicking this button, users cast a "yes" vote and agree to the proposed version of the page.
- Disagree button - by clicking this button, users cast a "no" vote and reject the proposed version.
- Delegate button - users can delegate their assignment to another user. This button opens a dialog for delegating the vote.

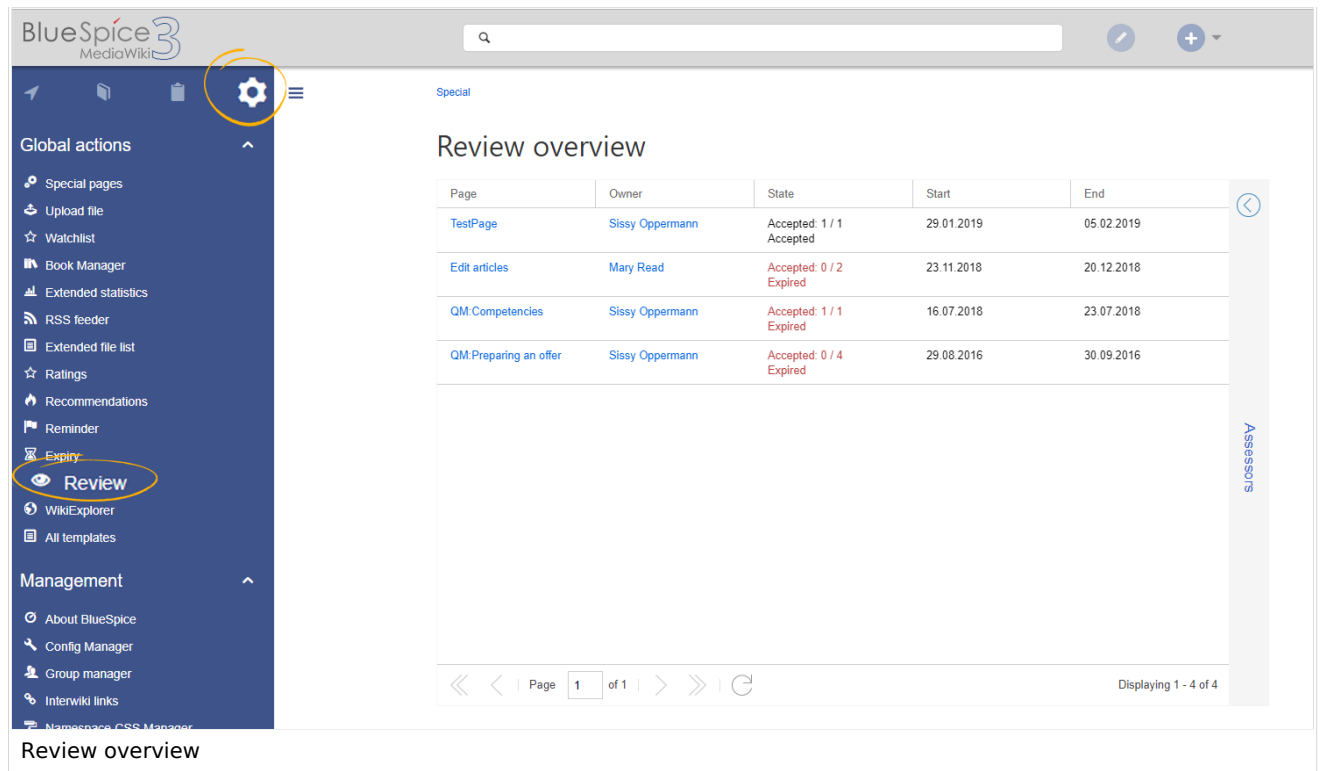
All votes along with comments are displayed under "Recent workflow steps" in the same flyout as the vote form and in the page history.

## Rejecting a task

If a reviewer votes with "Disagree" the workflow returns to the beginning. The next reviewer in line is now the originator of the workflow. The order of the initial reviewers after the originator is kept the same.

## Review overview

The page `Special:Review` provides an overview of all ongoing reviews. It can be accessed from the main navigation under "Global actions", item "Review". Clicking on an individual review opens a list of all reviewers assigned to review this page along with their votes (indicated by the eye icon before the username). A grey eye icon means the reviewer has not yet reviewed the page, green means a "yes" vote, while red means a "no" vote.



The screenshot shows the BlueSpice3 MediaWiki interface. The left sidebar has a 'Global actions' menu with 'Review' highlighted. The main content area displays the 'Review overview' page, which includes a table of review items.

Page	Owner	State	Start	End
TestPage	Sissy Oppermann	Accepted: 1 / 1 Accepted	29.01.2019	05.02.2019
Edit articles	Mary Read	Accepted: 0 / 2 Expired	23.11.2018	20.12.2018
QM: Competencies	Sissy Oppermann	Accepted: 1 / 1 Expired	16.07.2018	23.07.2018
QM: Preparing an offer	Sissy Oppermann	Accepted: 0 / 4 Expired	29.08.2016	30.09.2016

## Configuration

In the [Config manager](#), you can change the following settings:

- **Only the owner of a review can change it:** If this is unchecked, any user with edit rights can update a workflow.

## Related info

- [Reference:BlueSpiceReview](#)

## Manual:Quality management

A [quality version](#) of this page, [approved](#) on *27 April 2021*, was based off this revision.

### Contents

1 The importance of content quality .....	21
2 Page information .....	21

3	Page assignments .....	22
3.1	Characteristics .....	22
4	Reminder .....	22
4.1	Characteristics .....	22
5	Expiry .....	22
5.1	Characteristics .....	22
6	Review .....	23
6.1	Characteristics of the review feature .....	23
7	Acceptance .....	23
7.1	Characteristics of the acceptance feature .....	23
8	Permissions overview .....	24
9	Related info .....	27

## The importance of content quality

---

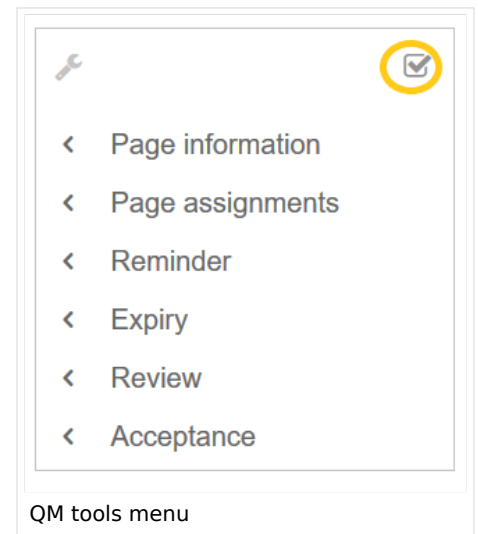
For many companies, the quality of the information they produce and share is of great importance. At the same time, they want to capture all the knowledge of their employees without creating any unnecessary barriers.

In BlueSpice, many users can contribute their knowledge. Ideally, anyone can contribute information right away and specialists can then review and approve the content. This ensures that the quality of the content meets the standards of the company. You should always weigh the benefits and drawbacks of individual quality management features so that you don't create unnecessary roadblocks for users who are eager to contribute.

To access these features, you open the QM tools (with the checkbox icon) in the Page tools panel.

For quality assurance purposes, each content page has the following built-in tools:

- Page information
- Page assignments
- Reminder (BlueSpice pro)
- Expiry (BlueSpice pro)
- Review (BlueSpice pro)
- Acceptance (BlueSpice pro - if FlaggedRevs is activated for the namespace)



## Page information

---

The page information gives you some very interesting statistics about the page. Based on this information, you can decide what quality management measures are necessary to improve the page.

The information that is listed on this page is:

- time of last edit and last editor
- number of views
- templates used on this page
- similar pages
- categories
- readers of this page
- authors of this page

## Page assignments

---

Page assignments are a quick way of giving one or more users informal responsibility and ownership of a page. They are very similar to a [watchlist](#). A user can add a page to their watchlist themselves, but assignments are made by other users.

Page assignments do not change the behavior of the page or grant special rights on a page. For example, the page does not get locked for editing when it is assigned to another user. It is rather a simple mechanism to have users "take a look" at a page. If the feature [Read confirmation](#) is activated for a namespace, assigned users have to confirm with a checkbox on the page that they have read the page.

### Characteristics

- **Notifications** are sent to assigned users when their page assignment is changed. (This can be deactivated in the user preferences.)
- **Transparency:** Assigned users are listed in the QM tools panel of the page
- **Assignment management:** ( [Special:ManagePageAssignments](#) )
- **Quick access** from the user profile menu
- **Read confirmation:** page shows a read confirmation checkbox for assigned users (optional)

[More about page assignments](#)

## Reminder

---

A user can create a reminder for a page by selecting a reminder date and by adding an optional comment. Reminders can also be created for other users by administrators and are a quick and informal way of setting due dates for page-related tasks.

### Characteristics

- The user with the reminder receives a notification. (This can be deactivated in the user preferences.)
- Users with reminders are listed on the reminders for the page in the QM tools.
- The page is listed on the Reminders special page under "Global actions" ( [Special:Reminder](#) ).
- The page is listed on the user's Reminders page, which is easy to access from the user profile menu.

## Expiry

---

If a page contains content with a known lifecycle, the page can be set to expire on a particular date. On the expiration date, a watermark "Outdated" is added to the page if the corresponding option is set in the [ConfigManager](#). In addition, a reminder can be set so that the user will receive a notification when the page expires. The expiration can be deleted from the page at any time.

### Characteristics

- **Watermark** on page background and in PDF (can be configured)
- **Management list:** *Special:Expiry*
- **Reminder** possible
- **Duration:** Expiry can be removed at any time

## Review

---

When a more formal review process is necessary, users can create a review workflow. The review workflow has an end date and is the only quality management tool that can lock the page for editing during the review phase. Each user casts a vote whether the page is acceptable or not.

The page is listed on the special page "Review" under "Global actions" ( `Special:Review` ) and on the reviewers' personal Review page.

### Characteristics of the review feature

- **One active review process** per page
- **Types:** Sequential or parallel review
- **Acceptance or rejection** by participants
- **Page is locked** for editing (Exception: depending on the configuration, reviewers can get edit rights)
- **Automatic restart** possible
- **Revisions-based:** Entries in the version history
- **Acceptance:** can be combined with the Acceptance feature

## Acceptance

---

If the pages in a namespace should have a published and a draft version, the setting [FlaggedRevs](#) needs to be activated for that namespace. This activates the *Acceptance* feature of the quality management tools.

All changes to a page are collected in a draft version until any user with review rights accepts the page. The pages in this namespace are in draft mode until a user with the right to publish content accepts the page.

There is a global option in the ConfigManager to define which user groups are allowed to read unpublished content. By default, all logged-in users can view drafts.

### Characteristics of the acceptance feature

- **Revisions-based:** First draft, draft, accepted
- **External files:** "Freezes" embedded media and templates
- **Automatic acceptance:** Often the result of a review process
- **Permissions:** Read permissions can (cautiously) be limited

Permissions overview

This permissions table shows which [roles](#) have the necessary right to perform different quality assurance actions. The name of the right that grants a certain permission is shown in parentheses:

QA action	
user (read)	
editor	
reviewer	
admin	
intenance- admin	
bot	
Can create and manage <b>page assignments</b> for any user ( <i>pageassignments</i> )	
x	
x	
x	
Can <b>be assigned</b> to a page ( <i>pageassigneable</i> )	
x	
Can create and	



manage  
**reminders**  
for any  
user  
*(remindere<sup>editall</sup>)*

x
x

Can  
create  
and  
manage  
their  
own  
**reminders**  
*(read)*

x

Can set  
and  
manage  
page  
**expiration**  
dates  
*(expire<sup>article</sup>)*

x
x
x

Can  
create  
and  
manage  
**review**  
**workflows**  
*(workflow<sup>edit</sup>)*

x
x
x

x
Can be assigned to a <b>review workflow</b> <i>(read)</i>
x
Can view the <b>review workflow</b> on the page <i>(workflowview)</i>
x
x
x
x
x
Can view the complete <b>list of reviews</b> <i>(workflowlist)</i>
x
x
x
x
Can <b>approve</b> a page <i>(review)</i>

x
x
x
Own changes are <b>automatically approved</b> changes <i>(autoreview)</i>
x
x
Rollbacks are <b>automatically approved</b> <i>(autoreviewrestore)</i>
x
x

## Related info

- [Working with drafts \(Flagged revisions\)](#)
- [The review workflow](#)

## 1 Manual:Semantic MediaWiki

A [quality version](#) of this page, [approved](#) on *22 December 2021*, was based off this revision.

Contents

1 Scope and terminology .....	29
2 Define properties .....	32
3 Create a template .....	32
4 Create a form .....	32
5 Assign a category .....	33
6 Example .....	33

## Scope and terminology

---

Semantic MediaWiki (SMW) is a MediaWiki extension that enables you to define and query data on a wiki page.

The following steps are usually taken to work with data-based content in your wiki :







1. **Define properties:** Properties determine which values can be queried in the wiki.
2. **Create a template:** This creates the output format of the data set on the wiki pages.
3. **Create a form:** Using this form, Wiki users can comfortably enter their data.
4. **Assign a category:** Each page that contains a semantic query is usually tagged with a category that was created for this purpose. This, in addition to the attributes, provides an important means of querying and filtering the pages. It also creates a form editing mode for the wiki pages that contain this category.

## Properties



### List of properties

---

1. Has First contact +  of type [Date](#) (0 uses) 
2. Has contact person +  of type [Text](#) (0 uses) 
3. Has location +  of type [Text](#) (0 uses) 

### Semantic properties

## Template



<b>Location</b>	Regensburg
<b>Contact</b>	Jane Doe
<b>First contact</b>	2020/12/21

Hallo Welt! was founded to make the technology and concept

Since 2007 Hallo Welt! builds collaborative platforms of knowl  
online documentation and quality management.


### Semantic template

## Form



**Location:**

**Contact person:**

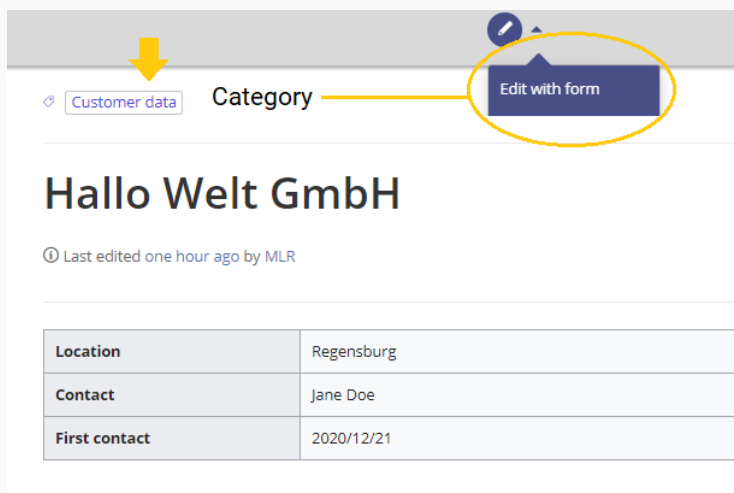
**First contact:**  

### Free text:

Hallo Welt! was founded to make the technology and concept c

Since 2007 Hallo Welt! builds collaborative platforms of knowle  
management, online documentation and quality management.

## Data entry form



The screenshot shows a data entry interface. At the top, there is a grey header bar with a yellow arrow pointing to a 'Customer data' tab and a blue 'Edit with form' button circled in yellow. Below the header, the title 'Hallo Welt GmbH' is displayed, followed by a note 'Last edited one hour ago by MLR'. A table below contains the following data:

Location	Regensburg
Contact	Jane Doe
First contact	2020/12/21

## Category to organize data

## Define properties

To make information available as data on a wiki page, we create so-called **properties**. For example, if we want to semantically record customer data, we create a wiki page for each customer. We collect customer data such as location on every customer page. To do this, we create a property that records the location of the customer.

```
[[Has location::Regensburg]]
```

Afterwards I can list all customers at a certain location or all customers with their associated locations in the wiki.

## Create a template

Since we often want to display several such properties and their values on one page at the same time, we format a special area that clearly shows the collected page attributes. This requires a (data) template, which can then be used on any wiki page.

Location	Regensburg
Contact	Jane Doe
First contact	2020/12/21

Data template

## Create a form

To support our users with data entry tasks, we create an input form. The input form is shown when users switch to the edit mode of a wiki page that has the semantic template included.

Location:	<input type="text" value="Regensburg"/>
Contact person:	<input type="text" value="Jane Doe"/>
First contact:	<input type="text" value="21/12/2020"/> 

Data entry form



## Assign a category

Wiki pages that use semantic templates are tagged with a category. This category enables the assignment of the input form to the wiki page. The category is also used to query the data collected via the template and display it anywhere in the wiki.

## Example

On the following pages, we will create an example that collects and displays customer data it in the wiki.

[Example: Collect customer data](#)

## Reference:BlueSpiceFlaggedRevsConnector

### Extension\: BlueSpiceFlaggedRevsConnector

Overview			
Description:	Adds support for FlaggedRevs to a range of other extensions.		
State:	stable	Dependency:	BlueSpice
Developer:	HalloWelt	License:	GPL-3.0-onlyProperty "BSExtensionInfoLicense" (as page type) with input value "</br>GPL-3.0-only" contains invalid characters or is incomplete and therefore can cause unexpected results during a query or annotation process.
Type:	BlueSpice	Category:	Infrastructure
Edition:	BlueSpice pro, BlueSpice Farm, BlueSpice Cloud		
<a href="#">View help page</a>			

## Technical Information

---

This information applies to BlueSpice 3 . Technical details for BlueSpice cloud can differ in some cases.

### Requirements

MediaWiki: 1.31  
BlueSpiceFoundation: 3.2

### Integrates into

- BlueSpiceExtendedStatistics
- BlueSpicePrivacy
- BlueSpiceReview
- BlueSpiceSMWConnector

### Special pages

- QMOverview

### Permissions

Name	Description	Role
autoreview	Have one's own edits automatically marked as "checked"	bot, maintenanceadmin
autoreviewrestore	Auto-review on rollback	bot, maintenanceadmin
movestable	Move pages with stable versions	admin, editor, maintenanceadmin, reviewer, structuremanager
review	Mark revisions as being "checked"	admin, maintenanceadmin, reviewer
stablesettings	Configure how the stable version is selected and displayed	admin, maintenanceadmin, reader
unreviewedpages	View the <a href="#">list of unreviewed pages</a>	admin, editor, maintenanceadmin, reviewer
validate	Mark revisions as being "quality"	admin, maintenanceadmin, reviewer

## Configuration

Name	Value
\$bsgFlaggedRevsConnectorBookshelfShowNoFR	true
\$bsgFlaggedRevsConnectorBookshelfShowNoStable	true
\$bsgFlaggedRevsConnectorBookshelfShowStable	true
\$bsgFlaggedRevsConnectorDraftGroups	array ( 0 => 'user', )
\$bsgFlaggedRevsConnectorFlaggedPagesStore	'BlueSpice\\FlaggedRevsConnector\\Data\\Flagg
\$bsgFlaggedRevsConnectorStabilizeSMWPropertyValues	true
\$bsgFlaggedRevsConnectorIndexStableOnly	false
\$bsgFlaggedRevsConnectorPageApprovedReminderDelay	168
\$bsgFlaggedRevsConnectorUEModulePDFShowFRTag	true
\$bsgFlaggedRevsConnectorautoDeleteWorkflow	true

## API Modules

- [bs-flaggedpages-store](#)

## API Modules

- [bs-flaggedpages-store](#)

## Hooks

- [ArticleContentOnDiff](#)
- [ArticleViewHeader](#)
- [BSExtendedSearchRepoFileGetRepoFile](#)
- [BSExtendedSearchWikipageFetchRevision](#)
- [BeforePageDisplay](#)
- [BeforeParserFetchTemplateAndtitle](#)
- [BlueSpiceEchoConnectorUserLocatorValidUsers](#)
- [EditPageGetCheckboxesDefinition](#)
- [FlaggedRevsRevisionReviewFormAfterDoSubmit](#)
- [ImagePageFileHistoryLine](#)
- [ImgAuthBeforeStream](#)
- [LoadExtensionSchemaUpdates](#)
- [MessageCache::get](#)
- [OutputPageBeforeHTML](#)

- [PageContentSaveComplete](#)
- [PageHistoryLineEnding](#)
- [SMW::RevisionGuard::ChangeRevision](#)
- [SMW::RevisionGuard::ChangeRevisionID](#)
- [SMW::RevisionGuard::IsApprovedRevision](#)
- [SkinTemplateOutputPageBeforeExec](#)
- [SpecialPage\\_initList](#)
- [checkPagelsReviewable](#)

## Reference:FlaggedRevs

## Extension\: FlaggedRevs

Overview			
Description:	Allows for article content management by editors and reviewers.		
State:	stable	Dependency:	MediaWiki
Developer:	Aaron Schulz, Joerg Baach	License:	<a href="#">GPL v2+</a>
Type:	<a href="#">MediaWiki</a>	Category:	Quality Assurance
Edition:	BlueSpice pro, BlueSpice Farm, BlueSpice Cloud		
For more info, visit <a href="#">Mediawiki</a> .			

## Features

**FlaggedRevs** is a MediaWiki extension that activates an approval mechanism.

New versions of an article are initially marked as drafts and remain so until a user with revision rights checks and approves the page.

The MediaWiki extension is customized in BlueSpice. It uses the [FlaggedRevsConnector](#) to work alongside the work flow tool [ReviewExtended](#), realising the complete functionality of an editorial process with a final approval step.

**Warning:** The setting `FR_INCLUDES_FREEZE` for handling the state of included files of an approved wikipage has [no long term support by](#) and should not be introduced in a new wiki for that reason.

## Using the Version History

Redirect to:

- [Manual:Using the Version History](#)