ContentEditions Documentation

Contents

- ContentEditions documentation
- Glossary
- Editions explained
- Compare two editions
- Switch to another edition of a page
- How editions work
- Install and configure ContentEditions
- Create and publish your first edition
- Move an edition to another wiki
- Copy and publish when pages are not approved
- Fix pages that failed to copy
- ContentEditions reference for administrators

ContentEditions documentation

Audience: Editors, administrators and developers
Type: Reference
Prerequisites: none

The documentation has one section for each group of users. Editors use the wiki to write and change pages. Administrators set up the server, configure the extension and create and publish editions. Developers write code that extends or builds on the extension. Within each section, tutorials guide you step by step through an example. How-to guides show you how to reach one specific goal. Explanations tell you how things work and why. References list facts, such as settings and hooks, for you to look up. A glossary at the end defines the terms for everyone.

Editors

- Explanation: Editions explained: you learn what an edition is, why one page can exist in several editions, and what the banner and the `Latest:` address show. Administrators should read it before the other administrator pages.
- How-to: Switch to another edition of a page: you learn how to open the same page in another edition or in the working area.
- How-to: Compare two editions: you learn how to see what changed between two editions, or between an edition and the working area.

Administrators

- How-to: Install and configure ContentEditions: you learn how to install the extension and set it up for your wiki.
- Tutorial: Create and publish your first edition: you follow one example from the first copy of pages to a published edition.
- Explanation: How editions work: you learn how an edition keeps its own pages, and how links, templates and files behave in it.
- How-to: Copy and publish when pages are not approved: if ContentStabilization is installed, you learn how to copy and publish with unapproved pages.
- How-to: Fix pages that failed to copy: you learn how to retry or exclude pages that did not make it into an edition.
- How-to: Move an edition to another wiki: you learn how to export an edition and import it into another wiki with maintenance scripts, including the files you copy by hand.

- Reference: Reference for administrators: you look up user rights, settings, special pages, maintenance scripts, the edition log, Workflow activities and background jobs.

Developers

- Explanation: How ContentEditions is built: you learn the three ideas behind the code. Editions are namespaces added at runtime, links, templates and files are resolved when a page is shown, and copies are saved like normal edits.
- How-to: Add a source type to the copy dialog: you learn how to let the copy dialog select pages from your own extension.
- How-to: Run code when an edition changes: you learn how to run your own code when an edition is created, published, archived or deleted, or when another edition becomes the latest edition.
- Reference: Reference for developers: you look up hooks, the extension attribute, REST routes, services, database tables and the job type.

Everyone

- Reference: Glossary: you look up what the terms of this extension mean, such as edition, working area or latest edition.

Glossary

Audience: End users, also useful for administrators and developers
Type: Reference
Prerequisites: none

This page defines the terms used in the ContentEditions documentation. Terms are in alphabetical order.

Terms

Archived

The state of an edition that you no longer offer to readers as current. Its pages stay readable. You can publish an archived edition again. You cannot archive the latest edition.

Banner

The notice at the top of a page that names the page's edition and its state. Pages in the latest edition say so.

Compare editions

A page that lists the pages that differ between two editions, or between an edition and the working area. Each entry is marked as added, removed, changed or moved, with a link to the changes.

Edition

One maintained copy of your documentation, kept in its own namespace. Several editions exist side by side, for example "1.0" and "2.0". An edition is planned, published or archived.

Edition log

The record of everything that happens to editions, such as creating, publishing or deleting one. You find it in the wiki's log under the type for editions.

Edition overview

The page where you list all editions and manage them: create, copy pages, publish, archive, set as latest and delete. Its title is **Editions**.

Edition switch

A menu on a page that lets you open the same page in another edition or in the working area. Entries for editions without that page are shown as plain text.

Excluded page

A page that you decided not to copy into an edition, or that you removed from a failed copy. You can use **Exclude** for a page that cannot be copied.

Failed page

A page that could not be copied into an edition. You can **Retry** it or **Exclude** it. An edition with failed pages is incomplete.

Incomplete edition

An edition that still has failed pages. You cannot publish it until you retry or exclude every failed page.

Latest edition

The one published edition that readers get by default. At most one edition is the latest. The Latest: address always shows its pages.

Latest: address

A page address that starts with `Latest:`, for example `Latest:Installation`. It shows the page of the latest edition, so links to it keep working when another edition becomes the latest.

Namespace

A MediaWiki area of the wiki that has its own prefix in page names, such as `Help:`. Each edition has its own namespace, named with the prefix `Version_` and the edition name, for example `Version_1.0:`.

Page checklist

The list of pages with checkboxes in the copy dialog. You choose which pages go into the edition. Pages start ticked; categories start unticked. Ticking a parent page ticks its child pages.

Pinned file

The version of a file that was current when you created an edition. The edition keeps showing that version, even if someone uploads a newer one later.

Planned, Published, Archived

The three states of an edition. A **planned** edition is in preparation and not yet offered to readers. A **published** edition is offered to readers. An **archived** edition is retired but still readable.

Publish anyway

An option to publish an edition although some pages are not approved. It is available only if you may approve pages in ContentStabilization.

Start from

The choice in the create dialog that says where the content of a new edition comes from: a working area, or another edition.

Template copy

A copy of a template or module that belongs to one edition. The edition then uses the copy instead of the shared original.

Unapproved page

A page whose newest text has not been approved in ContentStabilization. When you copy such pages, you decide for each one what to copy.

Working area

The part of your wiki where you edit the live documentation every day, usually the main namespace. It is not an edition. You can start a new edition from it.

User right

A permission that a wiki administrator gives to a user group. ContentEditions has three: view editions (everyone), create and delete editions, and publish and archive editions and set the latest edition.

Workflow activity

A step you can add to a process in the Workflows extension. ContentEditions provides four: create an edition, publish an edition, archive an edition and set the latest edition.

Technical terms

These terms are for developers. They are in alphabetical order.

Approval gate

The code that asks ContentStabilization whether a page is approved. It does nothing without ContentStabilization.

Edition namespace

The pair of namespaces that an edition owns: one for its pages and one for their talk pages. ContentEditions adds them at runtime from the table `ce_edition`.

Page row

One row of the table `ce_page_source`. It records one page of an edition and its status: `pending`, `copied`, `failed` or `excluded`.

Reserved namespace range

The namespace IDs that start at `ContentEditionsFirstNamespace`. ContentEditions uses an ID only once, also after you delete an edition.

Source definition

The type and value that name the pages to copy into an edition.

Source type

A way to pick pages: `namespace`, `category`, `subpages` or `titles`. `book` and `hierarchy` are added when Bookshelf or PageHierarchies is installed.

Editions explained

Audience: Editors and administrators
Type: Explanation
Prerequisites: none

This page explains what an edition is and what you see when you read a page of an edition. To move between editions, see Switch edition.

What an edition is

Many teams document a product that exists in several releases at once. Version 1.0 is still in use while version 2.0 is being written. A single set of pages cannot describe both.

ContentEditions uses editions for this. An edition is a set of pages copied from the working area at one point in time. Each edition lives in its own namespace, for example `Version_1.0:`, so its pages never mix with the pages of another edition. Pages of one edition keep their own text, even if someone changes the same page in the working area or in another edition afterwards.

The working area is where you edit the live documentation every day. It is not an edition. When the content is ready, someone with the right to create editions copies it into a new edition.

An edition is always in one of three states: **Planned**, **Published** or **Archived**. Planned means in preparation. Published means offered to readers. Archived means retired but still readable.

The banner

When you read a page of an edition, a banner at the top tells you where you are. It reads, for example: "This page belongs to the edition "1.0" (Published, latest edition)." It names the edition and its state. If the edition is the latest edition, it says so. The banner does not appear when you edit the page or view its history.

The banner also holds the edition switch, labelled **Switch edition**. Use it to open the same page in another edition.

Links in the text of an edition page lead to the pages of the same edition, when that edition has a copy of the target. Other links keep their original target. Templates work in the same way: if the edition has its own copy of a template, the page uses that copy.

The `Latest:` address

Readers often want "the current documentation", not a specific edition. For this, the `Latest:` address exists. The address `Latest:Installation` shows the page `Installation` of the latest edition. After the page has loaded, your browser shows the address of the real page, for example `Version_1.0:Installation`. A note says "(Redirected from Latest:Installation)".

You can use `Latest:` addresses in links on your wiki or in documents outside it. When another edition becomes the latest edition, the same link shows the page of the new edition.

You cannot edit a page through `Latest:`. If you try, the wiki tells you to edit the page in the latest edition instead. If no edition is the latest, or the latest edition has no page with that name, the wiki says so.

Editions in categories and in search

When pages are copied into an edition, they keep their categories. If every copy appeared in the category list, each page would show up once per edition. For this reason, your browser hides the pages of editions on a category page. A notice above the list says how many pages of editions are hidden. Its button **Show** displays them, and the button then changes to **Hide**. The count at the top of the list ("The following N pages ...") still includes the hidden pages. Without JavaScript, the pages stay visible.

When BlueSpice ExtendedSearch is installed, search hides the pages of editions by default as well. It shows them only if you filter the results by the namespace of that edition. There is no notice in search.

Related pages

- Glossary
- Create your first edition

Compare two editions

Audience: Editors
Type: How-to
Prerequisites:

- You have the user right `contenteditions-view`. By default, all logged-in users have it.
- The wiki has at least one edition.

This page shows how to list the pages that differ between two editions, or between an edition and its working area.

Steps

1. Open a page of an edition. Select **... all actions** in the page menu, then **Compare with another edition**. The page **Compare editions** opens with the edition of your page already selected in **Older side**. You can also enter `Special:CompareEditions` in the address bar, after the wiki address.
2. In **Older side**, check the first side, or select another. The list holds **Working area** and the editions, most recently created first.
3. In **Newer side**, select the second side. **Working area** always means the working area of the edition on the other side.
4. Select **Compare**.
5. To see what changed in a page, select **Show changes** in its row. This opens the standard page comparison of the wiki.

You see a table with the columns **Difference**, **Older side**, **Newer side** and **Changes**. The rows are sorted by difference as in the list below, then by title. If the two sides are the same, you see the message **No differences.**

What the differences mean

- **Changed**: the page exists on both sides with different content.
- **Moved**: the page has a different name on each side.
- **Added**: the page exists only on the newer side.
- **Removed**: the page exists only on the older side.

Only **Changed** and **Moved** rows have a **Show changes** link.

Notes

- Both sides must belong to the same working area. Otherwise you see **These two cannot be compared.** You cannot compare the working area with itself either.
- If a bookmarked address names an edition that no longer exists, you see **One of the chosen editions does not exist.**
- **Older side** and **Newer side** only decide what counts as added or removed. The page does not check which edition is older.
- The page matches pages by their title without the namespace. If a rename left a redirect, the page matches the old and new name. Only a direct redirect counts, not a chain of redirects.
- A page that moved to another namespace shows as **Removed** and **Added**, not as **Moved**.
- Talk pages are compared too. Redirect pages are not listed.
- A page counts as changed when its text differs. Who saved the page and when does not matter.

Related

To open one page in another edition, see Switch to another edition of a page.

Switch to another edition of a page

Audience: Editors
Type: How-to
Prerequisites:

- You are viewing (not editing) a page of a working area that has editions, or a page of one of those editions.

This page shows how to open the page you are reading in another edition, or in the working area.

Steps

1. Find **Switch edition**:
   - On a page of an edition, it is inside the banner at the top.
   - On a page of the working area, it is in a bar above the page text. There is no banner.
2. Select **Switch edition**. A list opens. The working area comes first, then the editions, newest first. Each edition shows its state, for example "2.0 (Published, latest)". The entry for the page you are reading ends with ": this page".
3. Select the entry you want: **Working area** or an edition.

You see the page in the edition you chose. The banner reads, for example: This page belongs to the edition "2.0" (Published, latest edition). If you chose the working area, you see the bar with **Switch edition** instead of a banner.

If an entry cannot be selected

An entry that ends with ": no such page" is not a link. That edition has no page with this name. The page may exist there under another name. Search for it, or select another entry.

If a page has a different name in another edition, the entry shows that name: for example `1.0 (Published) (as "Setup")`. You then open the page with that name. This happens when the page was renamed and the rename left a redirect. The menu follows such redirects only in the namespace of the page you are reading.

On a talk page, the entries lead to the talk pages of the editions.

If you do not see the menu

The menu appears only when you view a page. It does not appear when you edit a page or open its history. Pages in other namespaces do not have it.

Related

To open the page of the latest edition without choosing an edition, use the `Latest:` address. To see all editions, select **Edition overview** at the end of the list. To see which pages differ between two editions, see Compare editions.

How editions work

Audience: Administrators
Type: Explanation
Prerequisites: Editions explained

This page explains how ContentEditions builds an edition and what it does and does not copy. Read it to plan an edition and to understand its limits. For the settings, see the reference.

Two namespaces per edition

Each edition gets two namespaces: one for its pages and the next higher ID for their talk pages. The wiki takes the IDs from a reserved range that starts at 6100. This is why the number of editions is limited. See the reference for the setting.

The namespaces exist only as long as the edition exists. The wiki registers them at runtime from its list of editions. When you delete an edition, the wiki deletes its pages, its talk pages and its template copies. The edition ID stays unique, even if you reuse the name later.

Copying creates independent pages

When you copy pages into an edition, the wiki saves each page as a new page in the edition namespace. The text is an exact copy of the newest revision, or of the revision you chose when pages were not approved. The edit summary names the source page and revision. The wiki saves each copy as a minor edit, so that notification extensions such as NotifyMe stay quiet during a bulk copy. The copy has no history from the source and no stored link back to it.

From then on, the copy is an ordinary page. Changes in the working area do not change it, and changes to the copy do not change the working area.

The wiki takes pages from one source: the working area or another edition. It copies only pages in the namespace of that source. Talk pages are not copied. The wiki never overwrites a page that already exists in the edition. If the name is taken, the copy fails. See Fix pages that failed to copy.

You can copy pages into a planned edition more than once. Pages that are already registered are not copied twice. An edition that you start from another edition copies the pages of that edition.

A small copy runs at once. A copy above the page limit in the reference runs as background jobs of ten pages each, so that no single request runs too long. The edition cannot be published until all jobs are done.

Links

The text of a copied page still names the pages of the working area. A link such as `[[Setup]]` points to `Setup`, not to `Version_1.0:Setup`. ContentEditions does not change the saved text. When the page is shown, the wiki checks each internal link:

- If the edition has its own copy of the target, the link leads there.
- If it has not, the link keeps leading to the original page. That content is shared by all editions.

This is why you can copy a part of the content: pages that you leave out stay reachable through their original address. An edition that you start from another edition follows the chain back to the working area.

Templates and modules

A template or module is a shared resource. By default, an edition uses the current version of each template, so a change to a template changes every edition that uses it.

You can copy selected templates and modules into an edition. After you copy pages, the dialog **Templates for "name"** opens by itself. It lists the templates and modules that the pages of the edition use and that have no copy yet. Nothing is ticked, because sharing is the default. Tick the ones that should get their own copy and select **Copy selected**. For a planned edition, the button **Templates** in the overview opens the dialog again. The list comes from the link tables, so a page that is still waiting for its copy job may not be counted yet: open the dialog again when the progress is complete. A copy of `Template:Logo` for the edition `Version_1.0` is the page `Template:Version_1.0/Logo`. A module copy is `Module:Version_1.0/Name`. Pages of the edition then use the copy instead of the original. Templates that you do not copy stay shared.

The wiki finds a copy by its name. There is no separate list. A page that someone creates by hand under that name also works as a copy, so limit who may create such pages. Deleting an edition deletes its copies.

Limits:

- The copy takes the newest revision of the template. It does not include the templates that it calls. Those stay shared unless you copy them as well.
- ContentEditions does not change the text of a copied module. A module that calls another module with `require` still calls the shared original.

- TemplateStyles pages stay shared.
- The copies appear in the list of templates and in prefix searches. The wiki does not hide them.

Files

Files are not copied. Instead, the wiki records the version of each file that a page uses, at the time you copy the page. It pins the file. The pin includes the files that a template adds. The edition keeps showing that version, even if someone uploads a newer one. The page for a file stays shared. Only the file content is pinned.

An edition that you start from another edition keeps the file versions of that edition.

Limits:

- A file that only a template copy uses is not pinned, if you copy the template after the pages.
- If someone deletes the pinned version, the edition shows the current version. The edition overview warns about it: "The edition "*name*" pins a file version that is missing from this wiki: *file*. The current version is shown instead." The same warning appears after an import when you have not yet copied the files.
- When you delete a file version that editions pin, the deletion page names those editions.

What stays shared

Part	In an edition
Page text, categories	Copied. Independent.
Pages you leave out	Shared. Links lead to the original, which keeps changing.
Templates and modules	Shared, unless you copy them.
TemplateStyles	Shared.
Files	Pinned version of the file. The description page is shared.
Talk pages	Not copied.

Related pages

- Fix pages that failed to copy
- Copy and publish when pages are not approved
- ContentEditions reference for administrators

Install and configure ContentEditions

Audience: Administrators
Type: How-to
Prerequisites:

- You run MediaWiki 1.43 or later.
- You can edit the wiki's settings file.
- You can run MediaWiki maintenance scripts on the server.

This page shows how to load ContentEditions and create its database tables. ContentEditions is tested with MySQL and MariaDB only. The table definitions for PostgreSQL and SQLite exist but are not tested, and are not supported. To create your first edition afterwards, follow Create your first edition. ContentEditions has no other required software.

Load the extension

1. Put the folder `ContentEditions` into the `extensions` folder of your wiki.
2. Add this line to your settings file, for example `LocalSettings.php`:

```
wfLoadExtension( 'ContentEditions' );
```

Create the database tables

Run the update script from the wiki's root folder (in the BlueSpice containers: `/app/bluespice/w`):

```
php maintenance/run.php update
```

You see that the script adds the tables `ce_series`, `ce_edition`, `ce_page_source` and `ce_file_pin`. If you run it again, it reports nothing to add.

Change the settings (optional)

Set a variable in your settings file only if you need a different value. The namespace prefix and the name of the `Latest` namespace are edited in the configuration manager instead (see the reference). Example for the settings file:

```
$wgContentEditionsPageLimit = 200;
```

The reference lists all variables and their defaults. Set them before you create the first edition.

Run jobs for large copies (optional)

If you copy more pages than `$wgContentEditionsPageLimit` allows (default: 100), the wiki copies them through its job queue. BlueSpice Galaxy runs jobs automatically through a cronjob. On a wiki that does not run jobs, for example because `$wgJobRunRate` is 0, run them yourself:

```
php maintenance/run.php runJobs
```

To see waiting jobs, run `php maintenance/run.php showJobs`. When the copy is finished, the job type `contenteditionsCopyPages` is no longer listed.

Verify the installation

1. Log in as an administrator.
2. Open **Special:SpecialPages**. You find **Editions** and **Compare editions** in the list.
3. Open **Editions**. You see the list of editions, or "There are no editions yet.", and the button **Create edition**.
4. Open **Special:ListGroupRights**. You see the three user rights of ContentEditions. Administrators hold the right to create and delete editions and the right to publish and archive editions and set the latest edition. Other groups only hold the right to view editions.

If **Editions** is missing, check that the `wfLoadExtension` line is in your settings file and that the update script ran without errors. If **Create edition** is missing, your account lacks the right to create editions. To give a group these rights, use the permission manager under **Administration > Permissions**; see the BlueSpice manual.

Create and publish your first edition

> **Audience:** Administrators, also useful for editors who may create and publish editions
> **Type:** Tutorial
> **Prerequisites:** ContentEditions is installed. You are an administrator, or you hold the user rights **Create and delete editions** and **Publish and archive editions and set the latest edition**. Your main namespace contains a few pages, one of them named `Installation`. If ContentStabilization is installed, all these pages are approved.

In this tutorial, you create an edition named `1.0` from your working area, copy its pages into it and publish it as the latest edition.

Create the edition

1. Open `Special:ContentEditions`. The page is titled **Editions** and is the edition overview. You see the button **Create edition**.
2. Select **Create edition**. The dialog **Create edition** opens.
3. In **Name of the edition**, enter `1.0`.
4. Under **Start from**, keep **A working area (namespace)**. In **Working area**, choose **(Main)**.
5. Select **Create edition**. You see the dialog **Copy pages into "1.0"**. The edition now exists. Its namespace is `Version_1.0`.

Copy the pages

6. In the tab **Pages**, you see the page checklist of your main namespace. All pages are ticked. Keep them ticked. The counter shows, for example, **12 of 12 selected**.
7. Select **Copy**. You see a message such as "12 pages were added, 0 left out."
8. Select **Close**. The dialog **Templates for "1.0"** opens. It lists the templates and modules that the copied pages use. Leave them unticked, so that the edition keeps using the shared originals, and select **Close**. In the overview, the edition `1.0` has the state **Planned**. Wait until its progress reads, for example, "12 copied, 0 pending, 0 failed, 0 excluded". The overview refreshes by itself while pages are pending.

Publish the edition

9. In the row `1.0`, select **Publish**. The dialog **Publish "1.0"** opens. The checkbox **Also make it the latest edition** is ticked. Keep it ticked.

10. Select **Publish**. In the overview, the state of `1.0` is now **Published**, and **(latest)** appears next to the name.

Check the result

Open the `Latest:` address of one of your pages: `Latest:Installation`. You see the content of that page from the edition `1.0`. Below the title, a note says "(Redirected from Latest:Installation)", and the address changes to `Version_1.0:Installation`. A banner at the top says: This page belongs to the edition "1.0" (Published, latest edition).

Next steps

- Open the same page in your working area with the edition switch: Open a page in another edition.
- Work with unapproved pages: Copy and publish unapproved pages.
- Read what an edition keeps and shares: How editions work.
- Move an edition to another wiki: Move an edition to another wiki.

Move an edition to another wiki

Audience: Administrators with shell access to both wikis
Type: How-to guide
Prerequisites:

- Both wikis run ContentEditions and use the same namespace prefix (default `Version_`). You can read it in the BlueSpice configuration manager.
- On the target wiki, no edition has the same name. Upper and lower case do not count: "Beta" and "beta" clash.
- If the edition was started from a working area namespace or from another edition, that namespace or edition exists on the target wiki as well.

This page shows how to move an edition from one wiki to another with two maintenance scripts. The edition keeps its pages, its state, its pinned files and the record of where each page was copied from. Run all commands from the MediaWiki installation folder. The scripts need no user right: they run with full rights as the user "Maintenance script". Do not use `Special:Export` and `Special:Import` for this: they move only the pages.

Export on the source wiki

1. Run the export. Replace `1.0` with the name of the edition and `/tmp/edition-1.0` with a new or empty folder:

   ```
   php maintenance/run.php ContentEditions:exportEdition --edition "1.0" --out /tmp/edition-1.0
   ```

   You see a line such as `Exported edition "1.0": 12 pages (incl. 0 template copies), 12 page sources, 2 file pins.` The folder now holds `pages.xml` (the pages with their whole history) and `edition.json` (the state, the copy records and the pinned files).

Copy the files

The export does not contain files. Move them before you import the edition, or the edition shows its pinned files as missing.

1. In `/tmp/edition-1.0/edition.json`, note the file names under `pins`, for example `Logo.png`.
2. On the source wiki, find where the files are stored:

   ```
   php maintenance/run.php dumpUploads --local
   ```

   The command lists all files of the wiki, not only those of the edition.

3. Copy the files from the list that you need into one folder on the target server, for example `/tmp/edition-1.0-files`.

4. On the target wiki, import them:

```
php maintenance/run.php importImages /tmp/edition-1.0-files
```

This moves only the current version of each file. A pin to an older version still shows as missing after the import.

Import on the target wiki

1. Copy the export folder to the target server.

2. Check it first. This writes nothing:

```
php maintenance/run.php ContentEditions:importEdition --in /tmp/edition-1.0 --check
```

   You see `OK: edition "1.0" can be imported.` The check covers the folder, the prefix, the name and the source. The real run can still stop at other rules for new editions, for example a taken namespace name.

3. Run the import without `--check`:

```
php maintenance/run.php ContentEditions:importEdition --in /tmp/edition-1.0
```

   You see `Created edition "1.0"`, the output of the page import, and `Restored 12 page sources and 2 file pins.`

4. Run `php maintenance/run.php runJobs` so that links and search catch up.

   To make the imported edition the latest edition in the same step, add `--make-latest` to the import command. The edition must be published. The label then moves like with **Set as latest**: it is logged, and integrations are notified.

Check the result

1. Open `Special:ContentEditions`, the edition overview. You see the edition "1.0" with the state it had before.
2. If the script said that the edition was the latest edition on the source wiki and is not the latest edition here, use **Set as latest** to change that, or import again with `--make-latest` (see below). The import never takes the label from another edition by itself. It sets the label itself only when the wiki has no latest edition yet.
3. If the overview warns about missing files, upload the missing file versions.

4. Open a page of the edition. Pages by users that do not exist on the target wiki show the user name with the prefix `imported>` in the page history.

Things that are not moved

- Page approvals (ContentStabilization), the creator and the creation date. The edition log shows one "create" entry by "Maintenance script".
- Copy records for pages from a namespace that the target wiki does not have. The copied pages arrive, but the overview counts fewer copy records. The script tells you how many it left out.
- Older file versions, see Copy the files.

The state is restored as it was; the checks and hooks for publishing and archiving do not run again. For this reason, integrations that listen to a change of the latest edition do not notice an edition that became latest through the import (with `--make-latest` they do).

If something goes wrong

- **"An edition named … already exists on this wiki"**, **"The export uses the namespace prefix …"**, **"The source … of this edition does not exist"** or **"The export folder cannot be used"**: nothing was imported. Fix the cause and run again.
- **The page import or the restore step fails:** the edition stays in the state **Planned**, possibly without all pages, copy records or pinned files. Delete it in `Special:ContentEditions` and start again.

Copy and publish when pages are not approved

Audience: Administrators and editors who hold the rights to create and publish editions
Type: How-to
Prerequisites:

- ContentStabilization is installed. Approval is on for the working area of the edition.
- You hold the user right to create editions (to copy pages) or to publish them.
- To approve pages while you copy, you hold the right **Approve pages** in the source namespace. To use **Publish anyway**, you hold it in the namespace of the edition.

This page shows what to do when some pages are not approved while you copy pages into an edition or publish it. A page is unapproved when its newest revision is not approved in ContentStabilization.

Without ContentStabilization, or without approval for the working area, none of this applies. Copying and publishing then ignore approval.

Copy pages that are not approved

1. In the row of the edition, select **Copy pages**. The dialog **Copy pages into "*edition*"** opens, as in Create your first edition.
2. Choose the pages and select **Copy**.

If some of the chosen pages are not approved, the dialog does not copy yet. It lists them and says: "Some pages are not approved. Decide what to copy for them. Links to excluded pages lead back to the source." The source is the working area, or the edition you copy from.

3. In **For all unapproved pages**, choose what happens to all of them:
    - **Copy the last approved revision**: you copy the newest approved revision. The copy is approved too.
    - **Approve now and copy**: you approve the newest revision in the source, under your name, then copy it. The approval stays even if the copy later fails.
    - **Copy unapproved**: you copy the newest revision as it is. The copy is not approved.

    - **Exclude**: you leave the page out of the edition. Links to it lead back to the source.
4. To treat single pages differently, change their entry in the column **Decision**. **Same as for all** keeps the choice of step 3.
5. Select **Continue**.

You see the message "N pages were added, M left out." For one page it reads "One page was added, 0 left out." Excluded pages count as left out. Pages that were approved at the source are approved in the copy.

If one decision cannot be carried out, nothing is copied, and the dialog shows an error:

- "The page "*page*" has no approved revision." Choose another decision for that page.
- "You may not approve the page "*page*"." You lack the right to approve in the source. Choose another decision.

Publish an edition that has unapproved pages

1. In the row of the edition, select **Publish**. The dialog **Publish "*edition*"** opens.
2. Select **Publish**.

If pages of the edition are not approved, the edition is not published yet. The dialog lists the pages and shows: "N pages are not approved. You may publish anyway if you are allowed to approve pages. This is logged." The button now reads **Publish anyway**.

3. To publish despite this, select **Publish anyway**. You need the right to approve pages in the edition. Otherwise you see an error.

If you want to approve first, close the dialog, approve the pages, and open **Publish** again.

You see the state of the edition change to **Published**. If **Also make it the latest edition** is ticked, the edition also becomes the latest edition. The checkbox is shown in this case too.

The edition log records that you published although N pages were not approved. N is the exact number.

Notes

- A copy of an approved page is approved by the system account of ContentStabilization.
- If approving a copy fails, the copy stays unapproved. It appears in the list when you publish.
- Your decisions apply to the pages you copy in this step. If you add more pages later, pages you copied earlier are not asked again.

- Large copies run as background jobs. "Added" means the page is registered for copying. Pages that do not exist or lie outside the source namespace are listed after **Not copied:**.
- The BlueSpice permission manager manual describes how you grant rights.

Related

For the terms, see Publish anyway in the glossary.

Fix pages that failed to copy

Audience: Administrators and editors
Type: How-to
Prerequisites:

- You hold the user right to create editions.
- An edition in the state **Planned** has at least one failed page.

This page shows how to retry or exclude pages that could not be copied into a planned edition. You cannot publish an incomplete edition until no page is failed or pending.

Find the failed pages

1. Open `Special:ContentEditions`.
2. In the table, look at the column **Pages (copied / pending / failed / excluded)**. The third number counts failed pages.

You see a list below the table with the heading **Pages that could not be copied to "*edition*"**. Each entry shows the page, the reason, and the buttons **Retry** and **Exclude**.

Common reasons:

- `The page "…" already exists in the edition and is not overwritten.` A page with this name is already in the edition.
- `The source page "…" does not exist any more.` Someone deleted or moved the page you chose.
- `You may not read the source page or create "…".` The user who started the copy lacks the right to read the page or to create it in the edition.
- `Saving the copy failed: …` MediaWiki refused to save the new page. The message gives the reason.

Retry a page

1. Remove the cause:
   - For an existing page, delete it in the edition. A rename can leave a redirect under the old name, which still blocks the copy.
   - For a deleted or moved source, restore it or move it back.
   - For a missing right, ask an administrator to grant it. See the BlueSpice permission manager manual.
2. Next to the page, select **Retry**.

The copy is saved as an edit by you. For a small copy, you see the page leave the list and the number under **copied** go up by one. If the cause is still there, the page stays in the list with the same reason. For a large copy, the page first counts as **pending**.

Exclude a page

If you do not want the page in the edition, select **Exclude** next to it. No confirmation is asked.

You see the page leave the list and the number under **excluded** go up by one.

An excluded page is not copied. Links to it in the edition lead back to the page in the working area or source edition.

Publish

When no page is failed or pending, select **Publish** in the row of the edition, as in Create your first edition. You see the state of the edition change to **Published**.

If failed or pending pages remain, the **Publish** dialog shows: `The edition "…" is incomplete: some pages are still being copied or failed to copy.` Retry or exclude the failed pages, wait for the pending ones, and select **Publish** again.

Notes

- The edition log records each retry with the number of pages, and each exclusion with the page name.
- Large copies run as background jobs. While pages are pending, the table reloads automatically, and you cannot publish. Wait until the number under **pending** is 0. If a job cannot run, for example because you lost the right, its pages show up as failed.
- You can only retry or exclude pages of a planned edition. Without the right to create editions, the buttons are not shown.

ContentEditions reference for administrators

Audience: Administrators
Type: Reference
Prerequisites: none

This page lists the user rights, settings, special pages, log actions and workflow activities of ContentEditions, with their defaults. For steps, see Install and configure ContentEditions. For the concepts, see Editions explained.

User rights

Right	Label in the permission manager	Allows
`contenteditions-view`	View the list of editions	Open `Special:ContentEditions` and `Special:CompareEditions`. See the entry **Compare with another edition** in the page actions of pages that belong to an edition.
`contenteditions-create`	Create and delete editions	Create and delete editions. Copy pages and templates into an edition. Retry or exclude failed pages.
`contenteditions-publish`	Publish and archive editions and set the latest edition	Publish and archive editions. Set the latest edition.
`contentstabilization-stabilize`	**Approve pages** (a right of ContentStabilization)	Choose **Approve now and copy**, when you hold it in the source namespace. Use **Publish anyway**, when you hold it in the namespace of the edition. See Copy and publish when pages are not approved.

The three `contenteditions-` rights are global. You cannot grant them for single namespaces or pages.

Defaults of the extension. No other group or role holds these rights by default.

Group or role	Rights
MediaWiki group `user`	`contenteditions-view`
MediaWiki group `sysop`	all three
BlueSpice role `reader`	`contenteditions-view`
BlueSpice role `editor`	`contenteditions-view`
BlueSpice role `admin`	all three

The BlueSpice permission manager manual describes how you grant rights.

Settings

Two settings can be edited per wiki in the BlueSpice configuration manager (**Content structuring**, **Content Editions**). A value saved there takes precedence over the variable in the settings file. The other variables go into your settings file.

Setting	Default	Meaning
Configuration manager: **Namespace prefix of editions** (`$wgContentEditionsNamespacePrefix` as default)	`Version_`	Prefix of the edition namespace names. The prefix is stored per edition when you create it. The field is read-only while planned or published editions exist, and the wiki refuses a saved change. Delete or archive those editions first. Archived editions do not count.
Configuration manager: **Name of the "Latest" namespace** (`$wgContentEditionsLatestNamespaceName` as default)	`Latest`	Name of the alias namespace (see Latest: address). It can always be changed. Links that use the old name stop working.
`$wgContentEditionsFirstNamespace`	`6100`	First namespace ID of the range reserved for editions. Each edition uses an even ID and the next odd ID, which is its talk namespace. IDs that other namespaces already use are skipped.
`$wgContentEditionsMaxEditions`	`100`	Number of editions the reserved range can hold. Subpage support

Setting	Default	Meaning
		is on for the whole range. The wiki cannot create more editions.
`$wgContentEditionsLatestNamespace`	`6098`	Namespace ID of the alias namespace. It must not lie in the edition range.
`$wgContentEditionsPageLimit`	`100`	Largest number of pages that one copy saves directly. A larger copy runs as background jobs of 10 pages each.

Special pages

Page	Title	Right	Purpose
`Special:ContentEditions`	Editions	`contenteditions-view`	Lists the editions with their state, the number of pages and whether they are incomplete. With the right to create editions, it shows the buttons **Create edition**, **Copy pages** and **Delete**, and for failed pages **Retry** and **Exclude**. With the right to publish, it shows **Publish**, **Archive** and **Set as latest**.
`Special:CompareEditions`	Compare editions	`contenteditions-view`	Compares two editions, or an edition and the working area, page by page. It works without JavaScript.

`Special:CompareEditions` accepts the parameters `from` and `to`. Each is an edition ID. The ID `0` means the working area. If one parameter is missing, the page shows only the selection form. If an ID matches no edition, the page shows the message "One of the chosen editions does not exist."

Maintenance scripts

Script	Purpose
`ContentEditions:exportEdition`	Writes an edition to a folder. Options `--edition`, `--out`.

Script	Purpose
`ContentEditions:importEdition`	Creates an edition from such a folder. Options `--in`, `--check`, `--make-latest` (make the imported edition the latest one; it must be published).

Run them as `php maintenance/run.php <script> ...`. See Move an edition to another wiki.

Edition log

ContentEditions writes every action to the log type `contenteditions`, labelled **Edition log**, in `Special:Log`. Every entry shows the edition name, the edition ID and the ID of the edition namespace (not of its talk namespace). The edition ID is never reused. The name can be.

Action	Entry reads	Written when
`create`	created the edition, from *source*	An edition is created.
`delete`	deleted the edition with *N* pages	An edition is deleted.
`copy`	copied pages into the edition: *N* added, *M* left out, approval decisions: *decisions*	Pages are copied.
`retry`	retried the copying of *N* failed pages	**Retry** is used.
`exclude`	left out the failed page *page*	**Exclude** is used.
`publish`	published the edition, previous state: *state*	An edition is published.
`override`	published the edition although *N* pages were not approved	An edition with unapproved pages is published with **Publish anyway**.
`archive`	archived the edition	An edition is archived.
`latest`	set the edition as the latest edition, previously: *edition*	The latest edition changes.

Workflow activities

If the Workflows extension is installed, the workflow editor offers four activities. Each runs as the user who started the workflow, so the rights of that user apply. A workflow without an initiating user fails.

Activity	Type	Properties	Right
Create edition	`create_edition`	`name` (**Edition name**); `sourceNamespace` (**Source namespace ID**, optional: empty starts the edition empty)	`contenteditions-create`
Publish edition	`publish_edition`	`edition` (**Edition name**); `makeLatest` (**Make it the latest edition**, default on); `override` (**Publish even if pages are not approved**, default off)	`contenteditions-publish`. With `override`, also **Approve pages** in the edition.
Archive edition	`archive_edition`	`edition` (**Edition name**)	`contenteditions-publish`
Set latest edition	`set_latest_edition`	`edition` (**Edition name**)	`contenteditions-publish`

A successful activity returns the outputs `editionId`, `edition` (the name) and `state`. The workflow editor adds them to each activity, so a later activity can read them by the ID of the earlier one, for example `{{{MakeEdition.edition}}}` as the **Edition name** of a publish activity. An activity that you wrote in BPMN by hand needs an empty property of the same name for each output it wants to pass on.

An activity that cannot run stops the workflow with one of these messages:

Message	When
The workflow has no initiating user. Edition actions need a user whose rights can be checked.	The workflow has no initiating user.
User "*name*" is not allowed to do this edition action.	The initiating user lacks the right.
There is no edition named "*name*".	`edition` matches no edition.
The message of the rule that was broken	A rule of the edition forbids the action. Example: the edition is incomplete.

Background jobs

ContentEditions defines one job type.

Job	Purpose
`contenteditionsCopyPages`	Copies a batch of pages into an edition when a copy is larger than `$wgContentEditionsPageLimit`.

On a test wiki, a copy of about 2000 pages took about 7 minutes (the job queue sets the pace). Deleting such an edition took about 100 seconds in one request, so it needs a web server timeout above that.