Multilingual wiki (single wiki)

A wiki can be set up to include different language versions for wiki pages. Wiki administrators have to make some decisions regarding how they want to support each wiki user's language setting in addition to simply providing translated page content.

The following considerations refer to a single-wiki scenario in *BlueSpice free or pro* and don't apply if the language versions are maintained in separate wikis.

Contents

Concepts

To make a single wiki multilingual, you can:

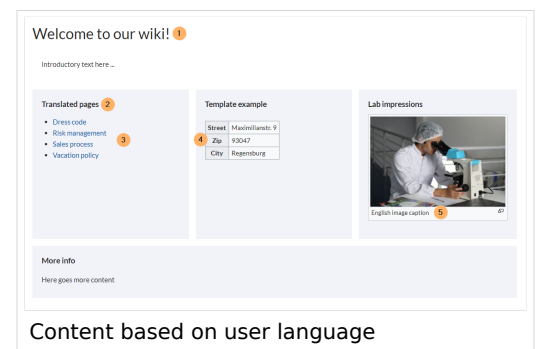- Create wiki pages with **automatically linked translation subpages**

In addition, depending on your wiki setup, it is possible to create:
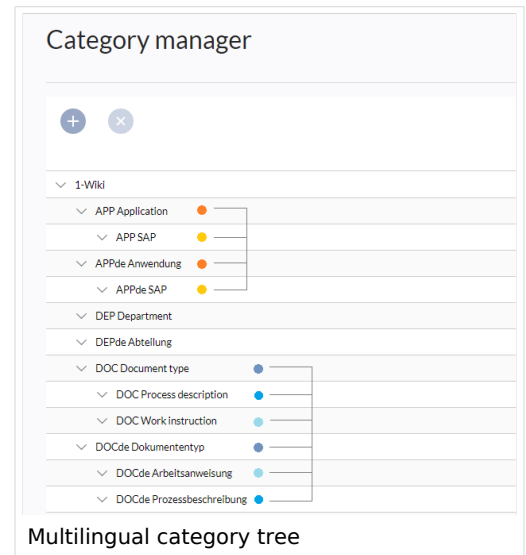
- multilingual **navigation links**
- **flexible page lists** for each language
- **portal pages** based on user language
- **template and form labels** based on user language
- a multilingual **category system**

Techniques

The explanations on this page use the following techniques, which we will look at in more detail:

1. **Automatic language switcher:**
   - The wiki automatically adds a language switcher to a page that has at least one language subpage.
2. **User language:** A simple mechanism to determine which language a user has defined in their personal settings. It provides options to:
   - show or hide content based on the user language
   - swap the entire page content shown to the user
   - customize the main navigation
   - display template and form labels to users based on their user preference.
3. **Categories:** A multilingual category system. It is useful for:
   - users that don't speak the "main" language (the content language) of the wiki and, therefore, want to tag pages in another supported language.

language switcher

Content based on user language

Multilingual category tree

A note about MediaWiki translation extensions

The following information gives you some suggestions for organising a wiki around its multilingual content based on the previously mentioned aspects. It does not consider using MediaWiki extensions like Translate or the MW Language Extension Bundle which can be installed in multlingual wikis to enable a section-by-section translation of wiki pages.

If most of your wiki content needs consistently accurate translations, these extensions can be valuable. Such a focus on translation, however, requires an understanding of how this mechanism technically works and might hinder the spontaneous contribution to the wiki by a broad user base. In addition, these extensions are not bundled with BlueSpice and require additional administrative effort during wiki updates or upgrades.

Here, we look at techniques that support multilingual content in an easy way.

Considerations before using the subpages approach

Many wikis use the approach of creating language subpages of a page that is written in the wiki's main language (=content language defined during wiki installation). This is a great mechanism if we assume that the majority of content is available in the wiki's main language first. This is often the case in English wikis. But what if there is no "main" language in a wiki - when the language(s) of many contributors do not match this content language of the wiki?

The solution in this case is often to maintain separate wikis for each language "community" and then connect the wikis via so called Interwiki links.

Interwiki links used within a single wiki

It is possible to think of each language as independent and to allow users to create pages in their preferred language. A German speaking user could create a page "Urlaubsregelung" in an English wiki. If another user creates a "matching" page "Vacation policy" in the wiki, these two pages can be connected manually by adding interlanguage links to the page. For this purpose, wiki administrators can create the needed interwiki links. The language switcher configuration can then be switched from suppporting language subpages to consider Interwiki links instead. A combination of both subpages and Interwiki links is not supported.
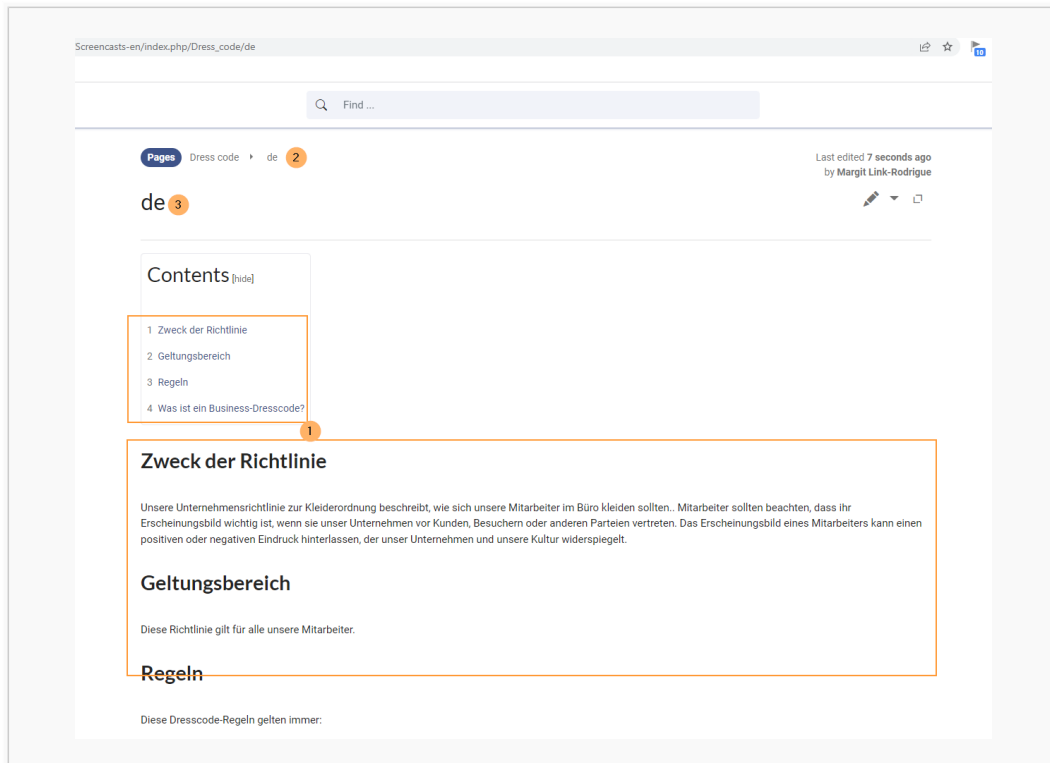
Creating wiki pages with translated subpages

Let's take a page called *Vacation policy* in a wiki with English as its content language and create a German language version.
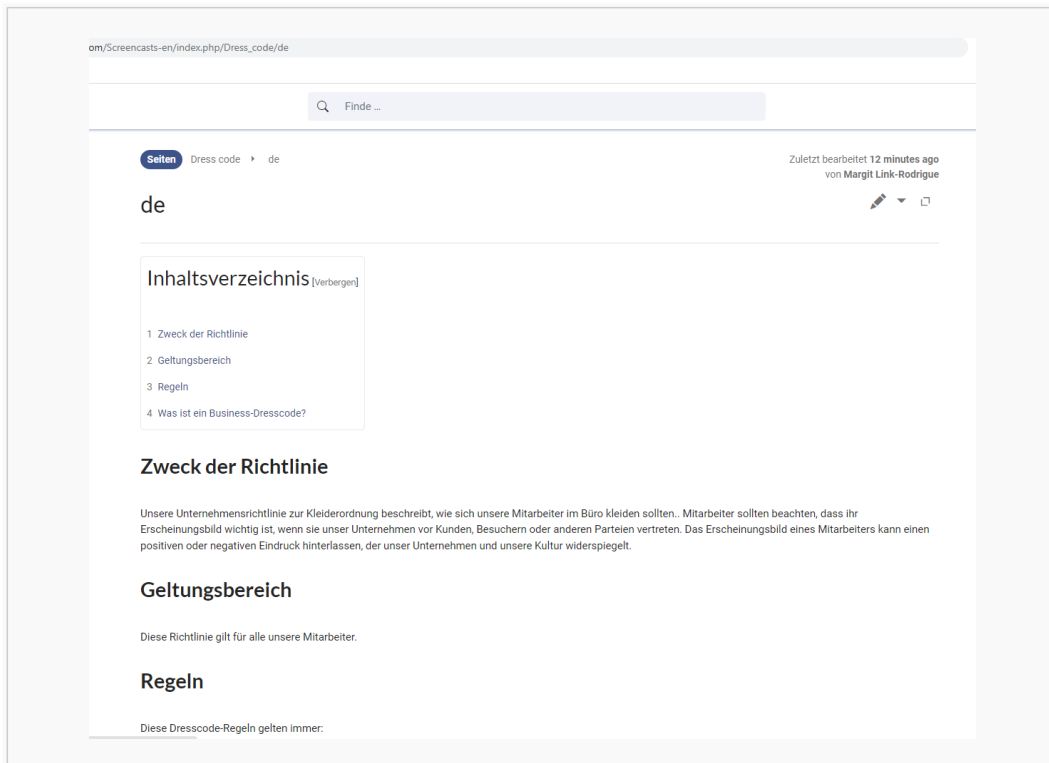
> **Note:** When working with translation subpages, it is important to know the content language of the wiki. It is the language in which the wiki has been installed. To find it out, you can add the magic word `{{CONTENTLANGUAGE}}` to the source view of any wiki page and save it. The page will then display the language code.

To create a language version for the main language (=content language), a subpage with the ISO-639-1 language code as subpage name is created. This means that for the page *Vacation policy* in an English wiki, you need to create the subpage *Vacation policy/de* for the German translation.

As you can see in the screenshots below, the page content is always shown in the context of the user language preferences of each user. This means that the user interface elements do not necessarily correspond to the translated language. But since the user decided to use this language setting, it does not negatively affect this user, but actually supports them when navigating the wiki.

German content version (1) with user language preference English (2).

German content version with user language preference German.

Connecting unrelated wiki pages instead of subpages

If you want to relate pages in the wiki as language equivalents, the language switcher needs to be configured on the server to consider interwiki links instead of subpages.

If you need a way to connect German and English pages on the same wiki, you need to set up the same interwiki link with a different prefix on the page *Special:InterWikiLinks*.

Setting up an interwiki link

The URL is always the same, the path of your wiki and the placeholder $1 for the pagename. The prefix reflects the language: *de*=German, *en*=English, *fr*=French, and so on.

You can then connect the page "Urlaubsregelung" to the English page "Vacation policy" by adding the following interwiki link to the wikisource of the page "Urlaubsregelung":

```
[[en:Vacation policy]]
```

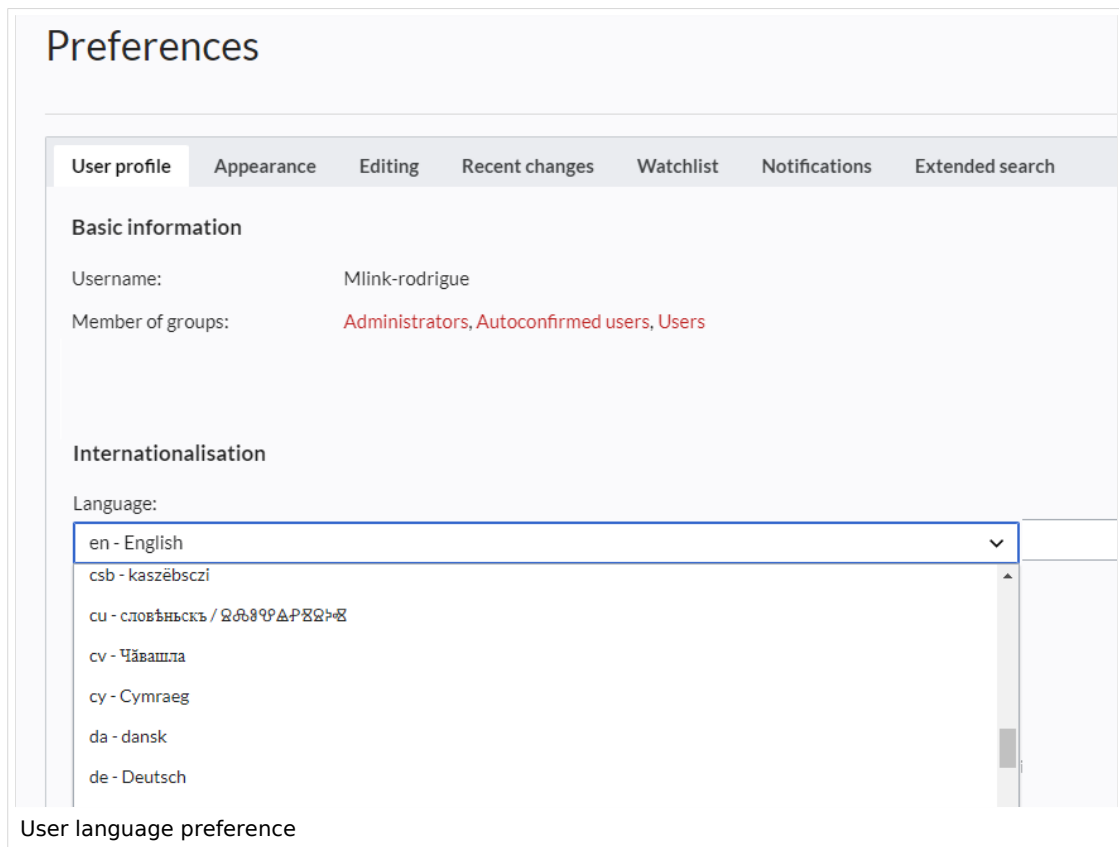## Creating portal pages based on user language settings

Portal pages are pages that serve as entry points to different topics or areas in the portal, often considering the needs of specific user groups. A typical example is the main page of the wiki, which should give users a high-level orientation in the wiki. A portal page is a good place to take the language needs of different users into account.

Here, we can take two different approaches:

1. Show the **same page to all users**, but include **some content** that targets specific user languages.
2. Show completely **separate content** to users based on their user language.

What is the user language?

Wiki users can change their own language setting in the user preferences:

User language preference

Depending on this setting, users then see the user interface elements in that language as we have seen in previous screenshots. This applies to interface elements that already have translated system messages.

To provide additional support for this setting, we can include checks on wiki pages that determine the user language and serve up additional or alternative information as needed.

Preparing the user language check

Checking for the user language requires that we create a system message for this purpose. This is easy to do, but needs to be done by users in the role *admin* or *structuremanager.*

To create a system message for the user language:

1. **Create** the page *MediaWiki:lang* with the content being the language code of the content language of your wiki. If you are not sure in what language your wiki was installed, you can add the magic word {{CONTENTLANGUAGE}} to the source of any wiki page and look at its output. If you have an English wiki, you need to add "en" as the content to this page. If your wiki is German, add "de" instead.

Page MediaWiki:Lang

2. **Create** subpages to this page for the user languages you want to support. For example in an English wiki:
   - MediaWiki:Lang/de
   - MediaWiki:Lang/fr

Now we have a mechanism to detect the user language. By checking for the value of the "magic word" {{int:lang}} , we can use it in many different scenarios.

Displaying additional content

To show content on a page only to users with a specific language setting — for example, German — we can add the following check:

```
{{#ifeq: {{int:lang}}|de|show this content|}}
```

Switching to different page content

To show entirely different content to users with different language settings, you can prepare different pages and then serve them up as needed. As an example we take the homepage of the wiki.

To display different homepage content based on user language in an English wiki, where the homepage is called *Main Page*:

1. Create a subpage for each language you want to support, e.g.
   1. Main Page/en
   2. Main Page/de
   3. Main Page/fr
2. On the page Main Page, add the following switch:

```
{{#switch: {{int:lang}}|de={{:Main Page/de}}|fr={{:Main Page/fr}}|{{:Main Page/en}}
}}
```

Users with language setting "de" will see the content of page *Main Page/de*, users with "fr" language setting the page *Main Page/fr*, all other users get the page Main *Page/en*.

Switching labels

To show different labels or short text snippets directly on a page, we can also use the switch function:

```
{{#switch:{{int:lang}}|de=Überschrift 2|en=Heading 2}}
```

To summarize, the following main page can be accomplished in two different ways:

1. By creating separate subpages for each user language and then transcluding them with a user language check on the actual Main Page.
2. By loading the actual Main Page and then building the language checks into the page itself. The following picture shows examples of:
   - (1), (2) Switching headings
   - (3) Displaying query results based on the tracked language from the language switcher template (e. g., [[Languages/tracklang::en]])
   - (4) Switching display labels in templates (this can also be applied to page forms)
   - (5) Switching image captions

Content based on user language settings

Switching template and form labels

As the previous screenshot shows, labels in templates (and forms) can also be shown based on user language. You can simply add the switch function directly in the template. The template *Template:Contactinfo* collects street, zip and city info:

```
{| class="wikitable"
! {{#switch:{{int:lang}}|en=Street|de=Straße}}
| {{{street|}}}
|-
! {{#switch:{{int:lang}}|en=Zip|de=PLZ}}
| {{{zip|}}}
|-
! {{#switch:{{int:lang}}|en=City|de=Stadt}}
| {{{city|}}}
|}
```

It displays the different labels as needed:

Street	Maximilianstr. 9
Zip	93047
City	Regensburg

Straße	Maximilianstr. 9
PLZ	93047
Stadt	Regensburg

English labels   German labels
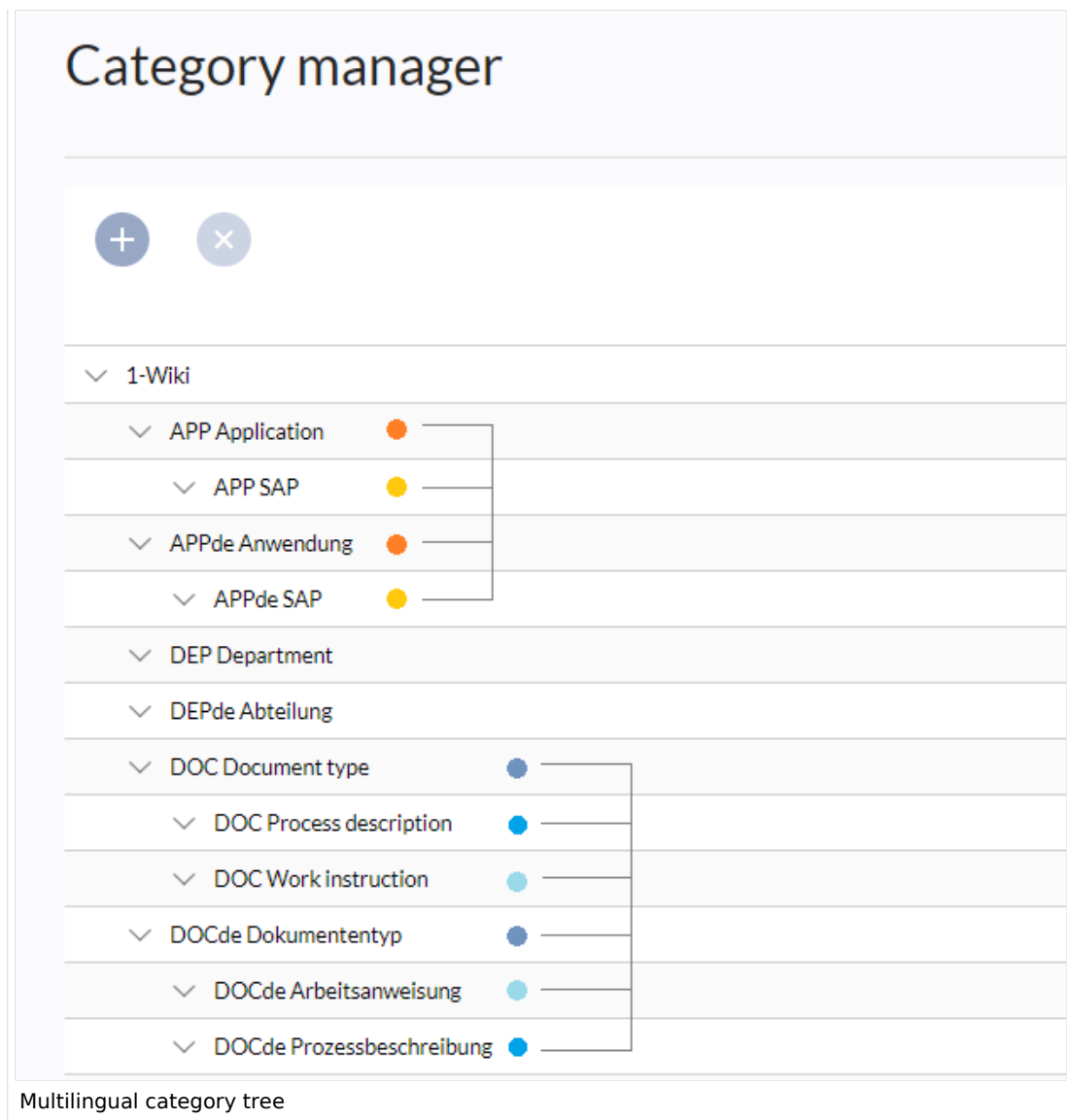
Multilingual navigation

Currently, there is no easy mechanism to create multilingual navigation menus, such as the main navigation or the custom menu. More support should be coming in one of the next minor releases of BlueSpice (4.2 or 4.3). In the meantime, you can create individual links based on a user's language settings.

Multilingual categories

An important aspect of organizing content is a well-designed category system for your wiki. By tagging your content, you support both the search functionality and the possibility of creating context-based page lists in all kinds of variations.
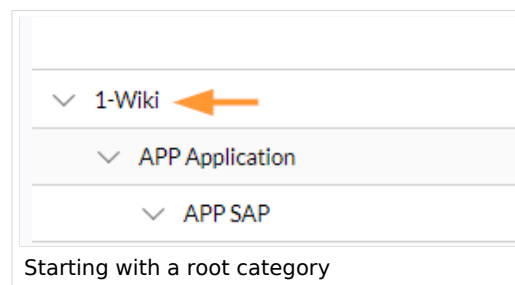
One way to accomplish this is to work with a prefix system that associates the different language categories.

For example, for the category Application, you would use (in an English wiki) the English category *APP Application* and the German category *APPde Anwendung*. They would have a subcategory APP SAP (English subcategory) and APPde SAP (German subcategory).

Multilingual category tree

Creating a root category first

It is also a good idea to start your category system with a single root category:

Starting with a root category

This allows you to later manage your category system without entangling it with other unrelated categories that might be created by users. Besides your category system, other types of categories might coexist:

- Hidden categories
- Tracking categories

- Informal tagging-categories created by users "on-the-go"

Querying pages by language categories

To find the pages in a particular language and in a particular category, we can now use our multilingual category system. For example, we can show all pages in the English category *DOC Process description:*

```
{{#ask: [[Category:DOC Process description]] [[Modification date::+]]
|format=Category
}}
```

> **Note:** To avoid deleted pages showing up in your results list, you can add the property [[Modification date::+]] to any of your queries.