

Contents

1. Manual:Semantic MediaWiki/Naming conventions	2
2. Manual:The concept of namespaces	5

Naming conventions

{{Messagebox|

Contents

1	Consistent naming of pages	3
2	Classification of information	3
2.1	Categories	3
2.2	Properties	3
3	Subpages	4
4	Related info	5

Consistent naming of pages

Before you create a semantic template, you first need to define which data to collect. Usually the data can be seen as a data set.

Example: We want to collect customer data. All customers get their own wiki page. On each page we enter information about the company *location*, the primary *contact person* and the *date of first contact* with the company.

It makes sense to name the required pages consistently. In our example, we would create the pages *Template:Customer data*, *Form:Customer data* and *Category:Customer data*. You can even consider putting the pages in their own namespace [namespace Customer](#). [Manual:The concept of namespaces](#) For the properties, you have to consider that the property *location* could also be used outside the context of customer pages. There could be locations for partners or vendors, for example. Each context could get their own property (e.g. *Customer location*) or share the same property *location* and then use a different category (Customer data, Partner data, ...) on the wiki pages to distinguish them.

Classification of information

Categories

Generell unterscheiden wir bei der Klassifizierung von Seiten zwischen Kategorien und Attributen. Mit Kategorien wird die Seite an sich beschrieben. Am Beispiel Kunden kategorisieren wir jede Kundenseite mit dem Schlagwort *Kundendaten*. Die Kategorie sammelt also alle Seiten, auf der sich Kundendaten befinden.

When classifying pages, we generally differentiate between categories and properties. The page itself is described with categories. Using the example of customers, we categorize each customer page with the keyword *Customer data*. The category therefore collects all pages on which customer data is located.

Properties

In den Kundendaten werden nun bestimmte Eigenschaften gesammelt, die jeden Kunden genauer beschreiben. Hierzu werden Attribute erstellt. Im Normalfall stehen diese Attribute in einer direkten Beziehung zur Seite selbst. Daher kann es hilfreich sein, die semantische Beziehung über das Attribut auszudrücken:

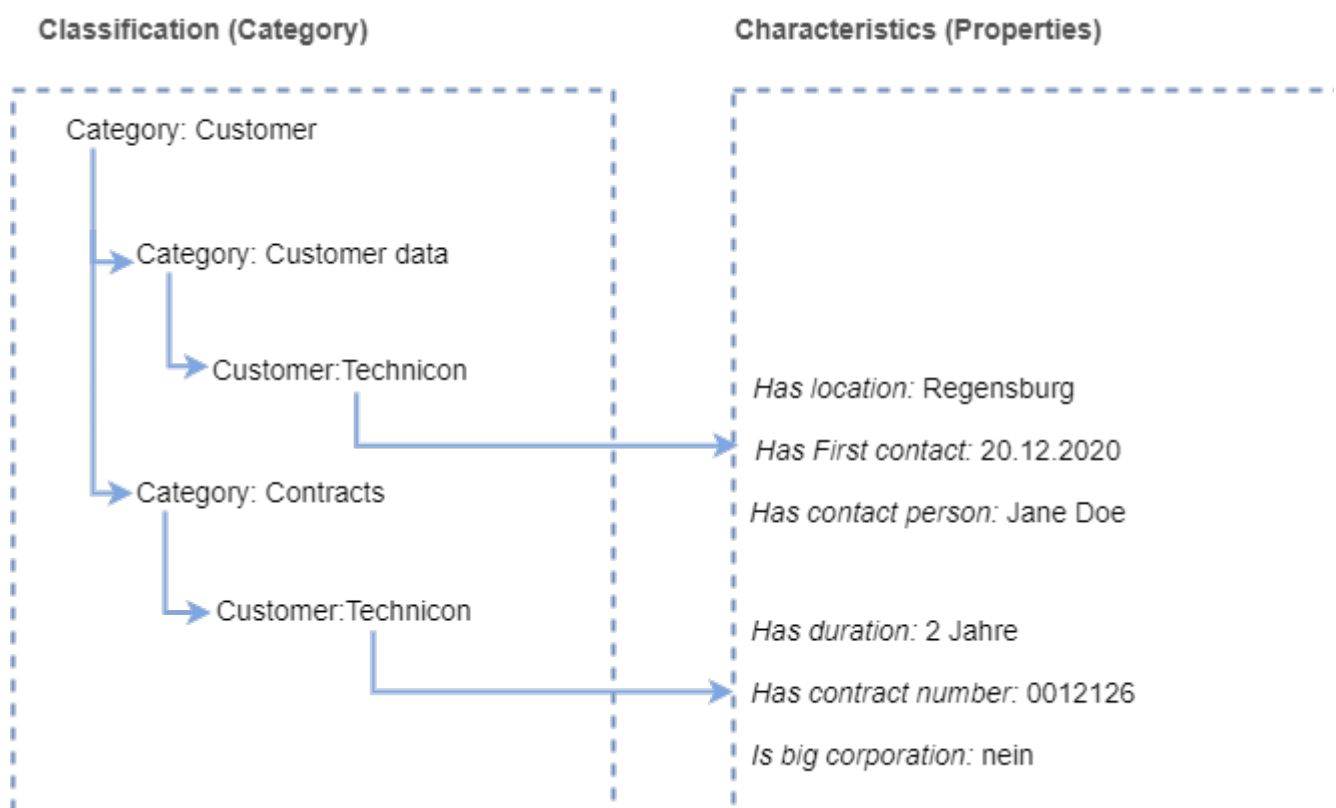
Certain properties that describe each customer more precisely are now collected as customer data. Normally, these properties are directly related to the page itself. It can therefore be helpful to express the semantic relationship directly in the property:

Customer Technicon has location Regensburg.
(page) (property) (value)

Therefore, we capture the relationship to the page in the property name: *Has location*.

Note: It is not required to express this relationship function of properties (as predicates). The property can also simply be named "location" if its intended use is clear.

There is, however a difference between "Has location" and "Is location of". For example, the customer Technicon has the location Regensburg. The city of Regensburg, on the other hand, is the location of the customer Technicon.



Subpages

We can also work with a subpage system and work with properties like Property:Customer/Has_location, Property:Customer/Has_First_contact, and so on. If the property "Location" is also to be used elsewhere, it is advisable to define Property:Has_location instead of Property:Customer/Has_location.

When properties can be clearly assigned to a use case or have several use cases, it makes sense to name them accordingly. For example, Property:Customer/Contract_number can be a sequential, whole number, but Property:Partner/Contract_number can contain entries such as "1.1.5" or "4.3.7".

Related info

- <https://www.semantic-mediawiki.org/wiki/Help:Classification>

Namespaces

Contents

1	What is a namespace?	6
1.1	Characteristics of a namespace	6
2	BlueSpice standard namespaces	6
3	Creating a new namespace	8
4	Namespace "MediaWiki"	8
5	Accessing pages in a namespace	8
6	Organizing content without namespaces	8
7	Viewing all pages of a namespace	9
8	Related info	9

What is a namespace?

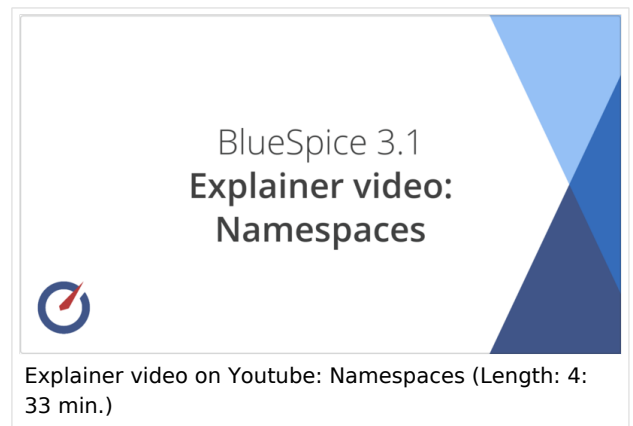
A **namespace** defines an area in a wiki. You can recognize a namespace by the namespace prefix in the page title:

```
<namespace>:Page title
```

We sometimes use shelves and closets as an example to make the principle of namespaces clear. There are shelves that everyone can access. Then there are glass cabinets with locks that everyone can look into, but not everyone can edit everything. There are also closed wooden cabinets with a lock, so only selected people can look into that space and work. And every closet can look different.

In a wiki, you will find predefined namespaces. In addition, wiki administrators can create custom namespaces as needed.

There are certain aspects of namespaces that are important to consider before creating a namespace. It is therefore important to understand the characteristics of namespaces and the reasons for using a custom namespace.



Characteristics of a namespace

- **Rights management:** Each namespace can contain its own user rights/roles assignments. However, we recommend flat user hierarchies for wikis.
- **Activation of features:** Different functions can be activated in each namespace (e.g., page approvals or read confirmations).
- **Search function:** Search can be limited to individual namespaces (namespace filter).
- **One namespace per page:** The same page name in a different namespace creates a new page.
- **Flat structure:** Namespaces cannot contain other namespaces.
- **File associations:** Uploaded files can be assigned to a namespace to apply namespace permissions to files.
- **Design:** Namespaces can be styled differently.

BlueSpice standard namespaces

BlueSpice uses, among others, the 15 standard and the two so-called pseudo namespaces from MediaWiki:

ID	Namespace	Function
	Media	Pseudo-namespace for uploaded images and files. Will be replaced by path of the files.
	special	Pseudo-namespace for special pages.

ID	Namespace	Function
0	(pages)	These are given without a prefix.
1	Talk	talk page. Is created in addition to each article (also in the following namespaces) to speak about its contents. Additional tab (+).
2	User	User page. Will be created for every logged in user.
3	User talk	Cf. talk
4	[Wiki-Name]	Information about the wiki or the current project. Here: BlueSpice.
5	[Wiki-Name] _Talk	Cf. talk
6	Image	Description of a picture and other files. Is automatically supplemented with version information and information about the occurrence of the file
7	File_Talk	Cf. talk
8	MediaWiki	system messages. A complete list can be found under Special: AllMessages. Can only be changed by registered users.
9	MediaWiki_Talk	Cf. talk
10	Template	Templates can be integrated into other pages.
11	Template_talk	Cf. talk
12	Help	Help pages describe how to perform actions in the wiki. No clear separation from the [wiki name] namespace.
13	Help_talk	Cf. talk
14	Category	Pages can be assigned to categories, which are then accessible in this namespace.
15	Category_talk	Cf. talk

Standard namespace: A page is assigned to the "Main" namespace, as long as no other namespace is specified.

Pseudo namespaces: The two pseudo namespaces (Media, Special) are used like normal namespaces, but you cannot edit them or add pages to them.

Most pages in the namespaces are created automatically:

- Special pages are generated by the system during installation.

- Media and image pages are created during file upload.
- User pages are created whenever a user acts on the wiki.
- Discussion pages are also created with the corresponding pages.

This leaves only the default namespaces (*Main*), *[wiki name]*, *Template*, *Help* and *Category* namespaces where pages can be freely created.

Creating a new namespace

You can create additional namespaces by using the Namespace manager under Global actions > Management > [Namespace manager](#). The menu item "Category manager" loads the page `Special:NamespaceManager`.

Namespace "MediaWiki"

Pages in the MediaWiki namespace can be created and edited by users with "editinterface" permission. By default, these are users whose group is assigned the *structuremanager*, *admin*, or *maintenanceadmin* role.

Important uses are:

- **System Messages:** Each message has a wiki page with the message key as the name of the page. A list of all announcement pages can be found on `Special:AllAnnouncements`.
- **Localizations via `{{int:messagename}}`:** An element of the user interface or within a wiki page can be translated into the user language via . Syntax: *Mediawiki:message name/ISO language code*.
- **MediaWiki:Common.js:** Contains JavaScript that is loaded for all pages and users.
- **MediaWiki:Common.css:** Stylesheet that is loaded after the stylesheet of the active skin in the wiki. It applies to all wiki pages.

Accessing pages in a namespace

To access or create a page in a specific namespace, its name is simply placed in front of the page title and separated by a colon:

```
[[Setup:Installation|]]
```

In the example, the pipe symbol (straight line) was also added. It has the effect that the namespace won't be displayed for the link label.

If you provide a namespace that is not known to the wiki, the page is created in the standard namespace Main. The portion of the reference specified as the namespace is then added to the title of the item instead and is not recognized as a namespace label.

Organizing content without namespaces

If you simply want to group pages in your wiki without the need for controlling access rights, you can use the syntax of the namespaces to **simulate a namespace in the namespace Main**. Simply prefix the pages that belong to a group with a name and a colon. If no namespace with this name exists, the prefix simply becomes part of the page name.

That way, it is possible to use pages with the same name in the main namespace and thus let different groups work on content "sets". Since these own "namespace" prefixes are part of the page name, you can now display all pages in a simulated namespace by searching for the prefix in the search field.

Since you can also organize pages with categories and subpages, you should start by planning how you would like to organize the content in your wiki.

After installing your wiki, you can visualize this with the BlueSpice draw.io Plugin directly in your wiki and discuss it with your team.

Viewing all pages of a namespace

You can view all pages in a namespace:

1. **Go** to the Special page "All Pages"
2. **Leave** the input field empty
3. **Select** a namespace from the Namespace drop-down menu
4. **Click** *Go*

Related info

- [Namespace CSS manager](#)
- [Namespace manager](#)