Manual:Using custom namespaces

From MediaWiki.org

In addition to the built-in namespaces, it is possible to add custom namespaces to a MediaWiki installation, to further separate content and allow more logical organization.

Custom namespaces are simple to manage using the `$wgExtraNamespaces` configuration directive. It is also possible to define alias names for custom (and also predefined) namespaces, using the `$wgNamespaceAliases` configuration directive.

Contents

Creating a custom namespace

You register additional namespaces by adding them to the `$wgExtraNamespaces` global variable. All namespaces require a unique numerical index in this array. As an example of simple custom namespace creation, adding the following lines to `LocalSettings.php` defines a "Foo" namespace 200 and its associated "Foo talk" namespace:

```
// Define constants for my additional namespaces.
define("NS_FOO", 200); // This MUST be even.
define("NS_FOO_TALK", 201); // This MUST be the following odd integer.

// Add namespaces.
$wgExtraNamespaces[NS_FOO] = "Foo";
$wgExtraNamespaces[NS_FOO_TALK] = "Foo_talk"; // Note underscores in the namespace name.
```

Pick an unused number

As a convention, the namespaces numbered 100-199 are reserved for site-specific namespaces, although there are some extensions that don't follow this convention. Extension writers use higher numbers, up to 32767. When choosing your index, you should avoid any number already in extension default namespaces, since you might want to install that extension later.

Even then odd

Note the namespace array index is 200 in the above example.

- an **even** namespace index denotes a subject namespace.
- the **odd** index immediately following that number denotes its associated discussion ("talk") namespace

Create the talk namespace too

You typically create a discussion "Talk" namespace along with each custom namespace. With this example, if you move a page into the "Foo" namespace, will be prompted to move its associated talk page, if any, and if you choose to do so, MediaWiki will place the talk page in "Foo talk".

No spaces

Use underscores instead of spaces when registering namespace names. "My Namespace" is invalid here; use "My_Namespace" instead.

Name the numbers you pick

The example defines constants (http://php.net/manual/en/language.constants.php) for the namespace IDs, so that you can refer to these namespaces later on in the configuration, for example in `$wgNamespaceProtection`, `$wgNamespacesWithSubpages`, or `$wgExtraGenderNamespaces`.

You could go on to configure additional settings for your new namespace.

```
$wgNamespaceProtection[NS_FOO] = array( 'editfoo' ); // permission "editfoo" required to edit the foo namespace
$wgNamespacesWithSubpages[NS_FOO] = true;            // subpages enabled for the foo namespace
$wgGroupPermissions['sysop']['editfoo'] = true;      // permission "editfoo" granted to users in the "sysop" group
```

Do it early

Manipulation of `$wgExtraNamespaces` must be completed during MediaWiki initialization; for instance it cannot be manipulated in a post-initialization hook like `$wgExtensionFunctions`.

Watch out for collisions with URL protocols

MediaWiki's linking code knows about a number of URL protocols, defined in the `$wgUrlProtocols` variable. If your namespace name is identical to one of these protocols, you're going to have trouble creating [[wikilinks]] to pages in your custom namespace. This most commonly arises when someone tries to create a "News" namespace, because `news:` is a URL protocol for NNTP newsgroups.
To avoid this issue, you can deregister the relevant URL protocol by adding the following code to LocalSettings.php (replacing `news` by the lowercased name of the protocol you wish to remove):

```
$wgUrlProtocols = array_diff( $wgUrlProtocols, array( 'news:' ) );
```

In extensions

Extensions often add their own namespaces, such as the Flow extension's "Topic" namespace. An extension can unconditionally add to `$wgExtraNamespaces` as described above, or if its namespace registration is conditional (for example EventLogging only defines its "Schema" namespace on the wiki where it stores schemas), then it can add a handler function for the CanonicalNamespaces hook that decides what to do.

The timing of registering extensions is subtle. Functions that extensions register with `$wgExtensionFunctions` are executed too late to register additional namespace (bug T47031). So extensions should bind the `CanonicalNamespaces` hook at file scope (in *`MyExtension`*`.php`) and check there whether the wiki should activate the extra namespace or not. Extensions can configure namespace permissions and content handlers unconditionally at file scope since they do not require the namespace to actually be created.

The new `extension.json` registration system has a namespace key for an extension to list its namespaces that should always exist.

MediaWiki version:	**≥ 1.25** Gerrit change 166705

Content namespaces

When building the site statistics page (see Special:Statistics), MediaWiki uses values stored in the database to calculate certain totals. One particular total is the "number of articles" or "number of content pages" figure.

For a page to be considered an article, or proper content, it must:

- Be in the main namespace, or a defined **content namespace**
- Not be a redirect page
- Contain at least one internal link

When creating custom namespaces to hold additional content, it is a good idea to indicate this in the configuration. This is done via the $wgContentNamespaces configuration directive.

To extend the example above, one might add the following to **`LocalSettings.php`**:

```
$wgContentNamespaces[] = 500;
```

or

```
$wgContentNamespaces[] = NS_FOO;
```

MediaWiki will now consider pages in the "Foo" namespace to be articles, if they meet the remaining criteria, and will include them when updating the site statistics counters.

Running maintenance scripts

- When adjusting the value of `$wgContentNamespaces`, it is a good idea to run either the `maintenance/updateArticleCount.php` or `maintenance/initStats.php` script to update the internal statistics cache (see: Manual:Maintenance scripts).

Why you would want a custom namespace

There are several reasons on why you would want this:

- A custom namespace can be used to hold content that should not be shown on the search results page, for example pages that are used only for transclusion.
- Certain namespace require additional privilege(s), i.e. for reading or editing
- You want certain namespace not to be subjected to certain limitation or default settings ($wgNoFollowNsExceptions for example)
- A uniform prefix for specific content(s), which is searchable for that namespace only
- If you're an MW developer, sometimes you need to have a custom namespace for your extension(s)

Dealing with existing pages

When storing page records, MediaWiki uses a namespace's numerical index, along with the remaining title text. Thus, when a page is created in a namespace that doesn't exist, e.g. "Bar:Some page", it is treated as being in the main namespace.

This can cause problems if adding a custom namespace definition for "Bar" at a later date, as MediaWiki will look for a page indexed via the proper namespace, but won't be able to find it, thus making the content inaccessible.

To correct this problem, there are three main approaches.

Move conflicting pages

If the number of pages affected is small (e.g. "Bar" held five pages created before the namespace was defined in the site configuration), then the following approach might be suitable:

1. Comment out the namespace definition in the configuration file
2. Access each affected page, and move it out of the pseudo-namespace, e.g. move "Bar:Some page" to "Bar2:Some page"
3. Un-comment the namespace definition
4. Move the affected pages back into the new namespace

Use a maintenance script

Within the **maintenance** directory, there is a maintenance script which performs the above operation more effectively for a large number of pages; **NamespaceDupes.php**. It is simple to use, but as with all MediaWiki maintenance scripts, consult the available usage information first (use `--help`) as an option.

Use a database query

To move all pages "Bar:Some page" into namespace 500, make the following database query:

```
UPDATE page SET
page_title = REPLACE(page_title, 'Bar:', ''),
page_namespace = 500
WHERE page_title LIKE 'Bar:%' AND page_namespace=0
```

To handle discussion pages:

```
UPDATE page SET
page_title = REPLACE(page_title, 'Bar:', ''),
page_namespace = 501
WHERE page_title LIKE 'Bar:%' AND page_namespace=1
```

Removing custom namespaces

The problem addressed above also occurs when a custom namespace definition is removed; MediaWiki is no longer aware of the numerical index for the namespace, and attempts to search the main namespace for the desired pages, leading to inaccessible content. This is a rare occurrence, since most sites will not need namespaces *removed*, but it is a problem. (See mailing list discussion (http://old.nabble.com/Removing-a-custom-namespace--tp30762329p30762329.html)).

Avoid namespace conflicts

In order for you to avoid namespace conflicts [e. g. your namespace has the same number as a namespace defined by an extension, the extension namespace list shows you which numbers to avoid to prevent conflicts.

Defining $wgNamespacesToBeSearchedDefault, $wgNamespacesWithSubpages, $wgContentNamespaces or $wgNamespaceAliases for an ID not associated to any existing namespace in $wgExtraNamespaces doesn't break the wiki; MediaWiki gracefully ignores such configurations.

Styling namespaces

For example, to set the background color of pages in a particular namespace (and its associated talk namespace) you can add following code to your common.css:

```
.ns-500 #content, .ns-501 #content { background-color: #f3f3ff; }
.ns-500 div.thumb, .ns-501 div.thumb { border-color: #f3f3ff; }
```

where `500` is the namespace's index and `#f3f3ff` is the color you want as its background color.

See also

- Manual:Configuration settings#Namespaces
- `$wgNamespacesToBeSearchedDefault`
- Namespace manager as originally proposed for MW1.6-wikidata and its successors. Currently in use by the OmegaWiki project.
- Extension:SkinPerNamespace to use a different skin in a namespace
- Extension:SpecialNamespaces, a modified version of the Extension:Interwiki which changes it to provide a namespace manager as a special page.
- Extension:NamespacePermissions and Extension:Lockdown to control access to namespaces
- Extension namespace registration

- Maintenance scripts

Language:	**English** • Deutsch • 日本語 • 中文

Retrieved from "http://www.mediawiki.org/w/index.php?title=Manual:Using_custom_namespaces&oldid=1538719"

Categories: MediaWiki configuration | Namespace

- This page was last modified on 17 April 2015, at 12:19.
-