Documenting noteworthy local templates

(Adding features from a Russian Wikipedia page into others)

There are certain Wikipedia pages in certain languages which have some special features, not present in other languages. These features are not at all language specific and their absence in other Wikipedia pages is surprising. These are features which can be easily pulled into Wikipedia pages of other languages as well, making life easier for both users as well as editors.

One such feature can be found on the Russian Wikipedia page. When we go to edit a Wikipedia page on Russian Wikipedia, after the sufficient editing is done, just like any Wikipedia page, we get an option to add our 'Edit summary' or give a small description of the changes made. The extra feature that this page has is a set of option to choose from for this small summary. There are a set of words in a blue bar, which when clicked on gets placed into the text box kept for addition of description, making is easier for the editor to add in an appropriate description in just one click. This is similar to how we use tags in normal pages after adding some content. This feature is of great help for Russian editors (and all other editors of Wikipedia pages where this feature is present) and it can very easily be included into other Wikipedia pages (such as English) as well.

This is a very simply feature. The onClick function invokes a function call to the "insertTags" function. There are a set of pre-defined words which can be included into the text box. On clicking any of the words from the set, the 'insertTags' function is called and that word is automatically filled into the text box, making it simpler for the editor, as he doesn't need to separately write it into the text box. Also, having a set of pre-defined words to choose from makes it easier for the editor to exactly describe the changes made. The administrators and other editors can also know what changes have been made and where exactly to look for those changes, simply by looking at the description of the change.

The set of predefined words which are presently used in Russian Wikipedia are words like blank, ambiguity, clean etc. These are simple words which very appropriately describes the kind of changes that are made to the page.

How to pull this feature into any other Wikipedia page (where it doesn't exist):

To get this feature into a Wikipedia page, where it doesn't already exist, we can refer to the Russian Wikipedia page. We need to pull in two important functions- "onclick" function and the function which onClick invokes i.e. "insertTags" function.

The 'insertTags' is a simple function whose job is to simply pull the selected word into the text box. This function can be simply re-used from the Russian Wikipedia page.

When we pull in these functions and are re-using the script, we simply need to make changes in the set of words that we want to give as option to the user. The set of words which presently exist in Russian now needs to be changed to English, with appropriate translation. Also, additional words can be used.

Eg.

Russian Wikipedia page:

{{заготовка}}

This is the present original script of the Russian Wikipedia page. The underlined word is the word which when clicked first invokes the onClick function which in turn invoked the “insertTags” function. If this script was to be re-used in the English Wikipedia page, it would look something similar to this:

English Wikipedia page:
{{blank}}
The underlined word again is the word which when click will be shown in the text box for description of changes.

Why exactly is it hard to port it in the first place?

It is hard to port this feature into other Wikipedia pages because the JavaScript page is in the Russian Wikipedia, and there is not proper way to install it in another Wikipedia except for manually copying and pasting the page. In this model there is no proper source code management and no installation instructions.

How to make it work without needing to copy paste code in every language?

The easiest way is to convert it into a MediaWiki extension that can be ported easily into any language. Extensions let you customize how MediaWiki looks and works.
Each extension consists of three parts:

- setup,
- execution, and
- Internationalization

A minimal extension will consist of three files, one for each part:
MyExtension/MyExtension.php: Stores the setup instructions.
MyExtension/MyExtension.body.php: Stores the execution code for the extension. For complex extensions, requiring multiple PHP files, the implementation code may instead be placed in a subdirectory, MyExtension/includes. For an example, see the Semantic MediaWiki extension.
MyExtension/MyExtension.i18n.php: stores internationalization information for the extension.

setup

Your goal in writing the setup portion is to consolidate set up so that users installing your extension need do nothing more than include the setup file in their LocalSettings.php file.
To reach this simplicity, your setup file will need to accomplish a number of tasks (described in detail in the following sections):

- register any media handler, parser function, special page, custom XML tag, and variable used by your extension.
- define and/or validate any configuration variables you have defined for your extension.

- prepare the classes used by your extension for autoloading
- determine what parts of your setup should be done immediately and what needs to be deferred until the MediaWiki core has been initialized and configured
- define any additional hooks needed by your extension
- create or check any new database tables required by your extension.
- setup internationalization and localization for your extension

Execution

The technique for writing the implementation portion depends upon the part of MediaWiki system you wish to extend:

- Wiki markup: Extensions that extend wiki markup will typically contain code that defines and implements custom XML tags, parser functions and variables.
- Reporting and administration: Extensions that add reporting and administrative capabilities usually do so by adding special pages. For more information see Manual:Special pages.
- Article automation and integrity: Extensions that improve the integration between MediaWiki and its backing database or check articles for integrity features, will typically add functions to one of the many hooks that affect the process of creating, editing, renaming, and deleting articles. For more information about these hooks and how to attach your code to them, please see Manual:Hooks.
- Look and feel: Extensions that provide a new look and feel to mediaWiki are bundled into skins. For more information about how to write your own skins, see Manual:Skin and Manual:Skinning.

Security: Extensions that limit their use to certain users should integrate with MediaWiki's own permissions system. To learn more about that system, please see Manual:Preventing access. Some extensions also let MediaWiki make use of external authentication mechanisms.

Internationalization

(Note: While developing, you may want to disable both cache by setting *$wgMainCacheType = CACHE_NONE* and *$wgCacheDirectory = false*, otherwise your system message changes may not show up).

If you want your extension to be used on wikis that have a multi-lingual readership, you will need to add internationalization support to your extension. Fortunately this is relatively easy to do. See also Manual:$wgExtensionCredits for guidelines on the mandatory information.

- For any text string displayed to the user, define a message. MediaWiki supports parameterized messages and that feature should be used when a message is dependent on information generated at runtime. Assign each message a lowercase message id.
- In your setup and implementation code, replace each literal use of the message with a call to wfMsg($msgID, $param1, $param2, ...). Example : wfMsg('addition', '1', '2', '3')
- Store the message definition in your internationalization file (MyExtension.i18n.php) . This is normally done by setting up an array that maps language and message id to each string. Each message id should be lowercase and they may not contain spaces. A minimal file might look like Manual:Special pages#The Messages/Internationalization File.
- In your setup routine, load the internationalization file : *$wgExtensionMessagesFiles['MyExtension'] = dirname(__FILE__) . '/MyExtension.i18n.php';*

How to avoid hardcoding the pre-defined set of words for every Wikipedia page?

Hard coding the set of words for every single Wikipedia page separately by translating them every time is not a good option. It must be possible to translate them and to customize them for every project. The easiest way to do this is to have a page in the MediaWiki namespace that will have some defaults and that will be editable by the project's administrators to allow further customization.

A custom namespace can be created in the following ways:

Adding an appropriate line to LocalSettings.php,

e.g. *$wgExtraNamespaces[500] = "Foo";*