

USER MANUAL

Power-Search for **MediaWiki**



Version 2.1



DIQA Projektmanagement GmbH
An der Raumfabrik 33c
76227 Karlsruhe
<http://www.diqa-pm.com>

Document Information

Document	DIQA - User manual Power-Search for MediaWiki - October 2013en-v8.docx		
Type	User Manual		
created	2013-11-21		
released	2013-11-21		
Doc-Version	8		
Status	final		

History:

Version	Date	modified by	Description
1	2013-10-10	Dh	<ul style="list-style-type: none">Created initial version
2	2013-11-19	KK	<ul style="list-style-type: none">
5	2013-11-20	MER	<ul style="list-style-type: none">Cleanup, verification, wording ... a few issues left
6	2013-11-21	MER	<ul style="list-style-type: none">Finalization (SMW Excursus, Intro, Linux)
8	2013-11-21	DH	<ul style="list-style-type: none">Corrections to installation section
9	2019-08-29	KK	<ul style="list-style-type: none">Update to Enhanced retrieval 2.1 (MW. 1.31+, SMW .3.0+)

© Copyright 2019 DIQA Projektmanagement GmbH
An der Raumfabrik 33c
76227 Karlsruhe, Germany

Phone: +49 721 609 517 26
Email: info@diqa-pm.com

Table of Contents

1	Introduction.....	5
1.1	Product Overview	5
1.2	Structure of this document	5
2	Administration Guide	7
2.1	Prerequisites	7
2.2	Software Download	8
2.3	Installation	8
2.3.1	Power-Search	8
2.3.2	Enhanced Retrieval	11
2.4	Configuration	12
2.5	First Initialization of the Search Index	13
2.6	De-Installation	14
2.6.1	Power-Search	14
2.6.2	Enhanced Retrieval	14
3	User Guide	15
3.1	Accessing the user interface of Power-Search	15
3.2	Description of the user interface	17
3.3	Search Area	17
3.4	Facets Area	18
3.5	Results Area	20
3.6	Search syntax	21
3.6.1	Normal mode	21
3.6.2	Expert mode	22
3.7	Excursus: Structured Data in MediaWiki	24
4	Technical Support.....	27
4.1	Download new releases of Power-Search for MediaWiki	27
4.2	Support inquiries	27
4.3	Buying a license.....	27

1 Introduction

Power-search for MediaWiki is a powerful replacement for the standard MediaWiki search. It is faster, more accurate, more expressive, and much more interactive. Power-search for MediaWiki provides these three main features:

1. It makes the textual content of wiki pages accessible in a split of a second and allows for complex search expressions.
2. Power-search also handles documents that were uploaded to the wiki (e.g. PDF, Word, and PowerPoint).
3. Based on semantic annotation stored in wiki-pages users can interactively navigate the multi-dimensional search space and combine this faceted browsing experience with the full-text search.

Power-Search employs “faceted browsing”; a technique that lets users explore information based on filters, so called *facets*. It is especially useful for users who are not familiar with the type and structure of the stored information. Searching with this technique is effective and intuitive, as the search results can be refined interactively.

Power-Search determines the facets automatically, based on the categories, properties and namespaces of the Wiki. By selecting a facet, users narrow down the set of previously found search results to those that also fulfil this additional filter.

1.1 Product Overview

- Compatible with MediaWiki 1.31 – 1.33
- Full text search in both Wiki content, as well as uploaded PDF or Office documents
- Live indexing: Any modification of wiki pages instantly updates the search index
- Incomplete search terms and wild cards are supported
- Combines full-text search with structured data
- Refine the search by category and by namespaces, e.g. find only documents in the user namespace that are in the category “manager”
- Restrict search results to pages with specific properties or property values, e.g. find documents that contain the keyword “Sommer” and have the value “P201” for the property “forProject”
- Data types such as numeric or date data values are correctly interpreted

1.2 Structure of this document

This document contains all the information needed to install and use this product from **DIQA Projektmanagement GmbH**:

- Product overview,
- Administration guide, incl. pre-requisites, installation, and configuration

- User guide and
- Technical support.

2 Administration Guide

2.1 Prerequisites

The Power-Search software builds on top of MediaWiki and Semantic MediaWiki. Thus, some software components are required before Power-Search can be used:

1. MediaWiki \geq v1.31.x and PHP \geq v7.0
2. Semantic MediaWiki v3.0

Semantic MediaWiki¹ provides the notion of structured data to MediaWiki. In particular, it allows storing properties with values for wiki pages providing a powerful platform for building wiki-based applications.

You can download Semantic MediaWiki from:

<http://semantic-mediawiki.org/wiki/Help:Download>

3. Enhanced Retrieval extension v2.1.x

The latest version of Enhanced Retrieval is available on GitHub:

<https://github.com/kkthek/enhanced-retrieval>

as well as on packagist as a composer-package:

diqa/enhanced-retrieval

The installation of the Enhanced Retrieval extension is explained in the sections below.

All these components (and more) are conveniently bundled together in **DataWiki**.²

In order to run Power-Search a recent Java Runtime environment is required

- Java 6 or greater. When using Java 7, be sure to install at least Update 1!

Further, a suitable server infrastructure is needed, either

4. Windows 7 or Windows 2008 RC2 (deploying a built-in Jetty web server), or
5. Linux distribution with Tomcat v6 or Tomcat \geq v7.0.5

¹ <http://semantic-mediawiki.org/>

² <http://diqa-pm.com/en/DataWiki>

2.2 Software Download

Download the Power-Search for MediaWiki software from the DIQA software repository:

<http://downloads.diqa-pm.com/free/power-search/>

Note:

The evaluation version of Power-Search is fully functional for 30 days. If you want to continue to use Power-Search after this period has elapsed you must buy a valid license key from DIQA. Please contact: info@diqa-pm.com

2.3 Installation

This installation guide assumes that you have installed the following software:

- Java \geq v6
- MediaWiki \geq v1.31.x and PHP \geq v7
- Semantic MediaWiki \geq v3.0

2.3.1 Power-Search for Windows

For Windows

1. Download the latest version of the Power-Search (which is based on Apache SOLR) from the DIQA software repository and store it on the server.
2. Copy the folder contained in the zip file to a suitable location of the server, e.g. `c:\SOLR_4_4`
3. To start the server manually, open a command line shell in the `wiki-solr` subfolder and execute the `start_solr4wiki` batch script.
4. To install the server as a windows service, that automatically starts whenever the windows machine starts, open a command line shell in the `wiki-solr` subfolder and execute the `installAsService` batch script. In order to do this the shell must run in administrator mode.
5. To verify that SOLR is properly installed open a web browser and point it to <http://localhost:8983/solr/> or a similar URL referring to the SOLR server.

2.3.2 Power-Search for Linux/Tomcat

If you don't have a SOLR server running, please do the following steps. If you have already a server, continue with 2.3.2.2

2.3.2.1 New SOLR server

1. Create file solr.xml at

`/etc/tomcat6/Catalina/localhost` (or similar)

with the following content:

```
<Context docBase="/usr/share/tomcat6/webapps/solr.war"
  debug="0" crossContext="true">
  <Environment name="solr/home" type="java.lang.String"
    value="/usr/share/tomcat6/DataWiki"
    override="true"/>
</Context>
```

2. Copy the `wiki-solr/DataWiki` directory from the Power-Search distribution to

`usr/share/tomcat6` (or similar)

i.e. the following path should exist:

`/usr/share/tomcat6/DataWiki/collection1/conf`

NOTE: Make sure the DataWiki folder is writeable by Tomcat.

3. Copy the jar-files from

`wiki-solr/lib/ext`

into Tomcat's lib directory

`/usr/share/tomcat6/lib`

4. Copy folders `dist` and `contrib` into

`/usr/share/tomcat6/DataWiki`

i.e. the following paths should exist:

```
/usr/share/tomcat6/DataWiki/dist
/usr/share/tomcat6/DataWiki/contrib
```

5. Change the file paths to the libraries in step (4) in the file

`/usr/share/tomcat6/DataWiki/collection1/conf/solrconfig.xml`

from

`../../../../../contrib/extraction/lib`

to

```
../contrib/extraction/lib
```

and adapt the other paths accordingly. NOTE: There are 8 paths in the file that need to be changed.

6. Default port of Tomcat is 8080. Either change it to port 8983 OR change the default port used in EnhancedRetrieval to 8080.

```
EnhancedRetrieval/proxy/env.php
```

7. Restart Tomcat:

```
/etc/init.d/tomcat6 restart
```

8. Check if SOLR runs:

```
http://localhost:8080/solr
```

If something fails, check the logs in

```
/usr/share/tomcat6/logs
```

2.3.2.2 Existing SOLR server

1. Find the location where SOLR stores its cores. You can open the SOLR web-application and select a random core in "Core Admin"-panel. On the right-hand side, you'll find the location in the filesystem.

e.g. `/usr/share/apache-solr/server/solr`

2. Unzip the zip file "core-name.zip" in this directory and rename the folder name "core-name" to a new name.
3. Change rights of folder: `chown -R solr:solr "core-name"`
4. Change name in core-name/core.properties file accordingly (like directory name)
5. Go to SOLR web-application, reload the page, select the new core and reload the new core by clicking the button "reload"

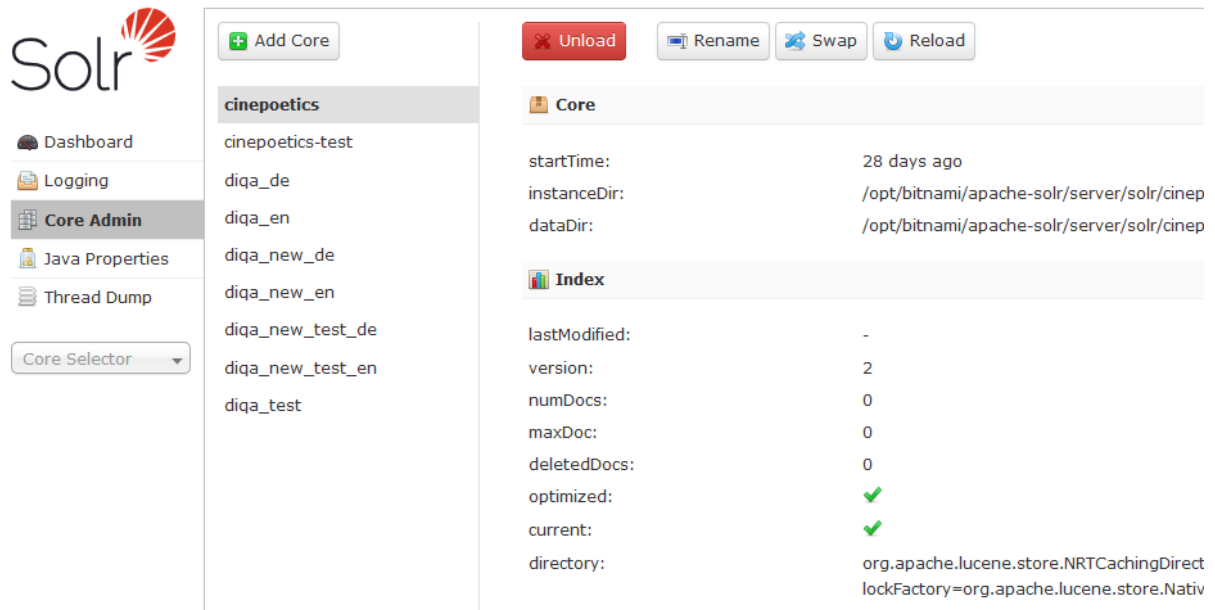


Figure 1: SOLR web-application

2.3.3 Enhanced Retrieval

1. Make sure you installed composer.
 - a. Windows: <https://getcomposer.org/download/>
 - b. Linux (Cent-OS): `yum install composer`
2. Install the latest release of EnhancedRetrieval-extension via composer. Go to a cmd-console, change to MW directory and type:

```
composer require diqa/enhanced-retrieval
```

3. Create a file in `$MW/extensions/EnhancedRetrieval/proxy/env.php` with the following content:

```
<?php

/**
 * SOLR connection data (REQUIRED!)
 */
global $SOLRhost, $SOLRport, $SOLRuser, $SOLRpass, $SOLRcore;
$SOLRhost = 'localhost';
$SOLRport = 8080;
$SOLRuser = '';
$SOLRpass = '';
$SOLRcore = '';

/**
 * Wiki connection data (REQUIRED!)
```

```
*/  
global $wgServer, $wgScriptPath, $wgDBname;  
$wgServer = "http://localhost";  
$wgScriptPath = "/mediawiki";  
$wgDBname = 'wikidb';
```

Note: we assume that your wiki database is named "wikidb" and that it's accessible under "http://localhost/mediawiki". Please correct this if necessary.

2.4 Configuration

Assuming that the wiki runs on the same machine as the SOLR server, the two components of Power-Search are aligned and work together without modifying any settings. The default port for the SOLR server is 8983. If some settings should be modified make sure that the settings are made on both sides, i.e. the SOLR server and the Enhanced Retrieval extension, otherwise they will not be able to successfully communicate with each other.

In order to configure the host or port for the SOLR server modify the configurations in `solr.xml` file in the `wiki-solr/DataWiki` folder. Further configuration settings are documented here:

<http://wiki.apache.org/solr/Solr.xml%204.4%20and%20beyond>

Similar settings can be changed on the client side, i.e. in the Enhanced Retrieval extension. Please adapt the settings in from this file, but don't configure it right there but rather in `LocalSettings.php`. Otherwise they get overwritten on the next update.

`EnhancedRetrieval/DefaultSettings.php`

Please note that the following settings have to be used only in `env.php`:

- `$SOLRhost`
- `$SOLRport`
- `$SOLRuser`
- `$SOLRpass`
- `$SOLRcore`
- `$wgServer`
- `$wgScriptPath`
- `$wgDBname`
- `$fsgNamespaceConstrain`
- `$fsgCustomConstraint`
- `$fsgUseStatistics`
- `$wgHTTPAuthForLocalProxies`
- `$fsgHTTPAuth`

For a detail description of the options, please check the files:

- EnhancedRetrieval/proxy/env.sample.php
- EnhancedRetrieval/INSTALL

Note:

An important configuration regarding SMW is that you *have* to disable deferred updates for now. Otherwise updating the SOLR index will not work. So please add to LocalSettings.php:

```
$smwgEnableDeferredUpdate = false;
```

2.5 First Initialization of the Search Index

Once the SOLR server and the Enhanced retrieval extension are successfully installed and configured the full text search index must be initially created. To do this an update script must be executed:

```
php EnhancedRetrieval/maintenance/updateSOLR.php -v
```

To verify if the installation was successful enter a search term in the wiki's search box. If a new search result page with some facets on the left hand side and some search results on the right hand side appear, the installation was successful.

In case the index is broken at some point in time or other issues arise with the Power-Search feature, it is a good idea to create the index completely from scratch. To do this, execute the following steps:

1. First stop the SOLR server (cf. next section on how to do this)
2. Then delete (or rename) the data-folder in the core. In our deployable of SOLR, it is located here:

```
{solrFolder}/wiki-solr/DataWiki/collection1/data
```

3. Restart the server.
4. Execute the update-script again and use the verbose (-v) and debug (-x) switches.

```
php EnhancedRetrieval/maintenance/updateSOLR.php -v -x
```

2.6 De-Installation

2.6.1 Power-Search

For Windows

Before uninstalling SOLR stop and uninstall the service (if it was installed). Go to *Services* in the *Administrative Tools* section of the *Control Panel*. Search for "SOLR service for DataWiki" and stop the service via the context menu. Once it is stopped open a command line shell in the `wiki-solr` folder and execute the `uninstallAsService` batch script. In order to do this the shell must run in administrator mode. If SOLR was not running as a service simply stop the SOLR process in the shell it is running.

Now the complete installation folder, e.g. `c:\SOLR_4_4` can be deleted. This will remove the program files as well as the full text index that was created for the Wiki.

For Linux/Tomcat

Before uninstalling SOLR please stop the application server, Tomcat:

```
/etc/init.d/tomcat6 stop
```

Remove or comment out the SOLR entry in `/etc/tomcat6/Catalina/localhost:`

```
<Context docBase="/usr/share/tomcat6/webapps/solr.war"
    debug="0" crossContext="true">
    <Environment name="solr/home" type="java.lang.String"
        value="/usr/share/tomcat6/DataWiki" override="true"/>
</Context>
```

Remove the folder:

```
/usr/share/tomcat6/DataWiki
```

Restart Tomcat (for other applications)

```
/etc/init.d/tomcat6 start
```

2.6.2 Enhanced Retrieval

In order to remove "Power-Search" from the Wiki installation remove, open a cmd-console, change to MW root folder and type:

```
composer remove diqa/enhanced-retrieval
```

The extension is now removed.

3 User Guide

Power-search for MediaWiki makes both, the full text of the wiki pages (and documents) as well as the semantic information contained in the wiki (cf. Section 3.7) accessible via a single, intuitive user interface.

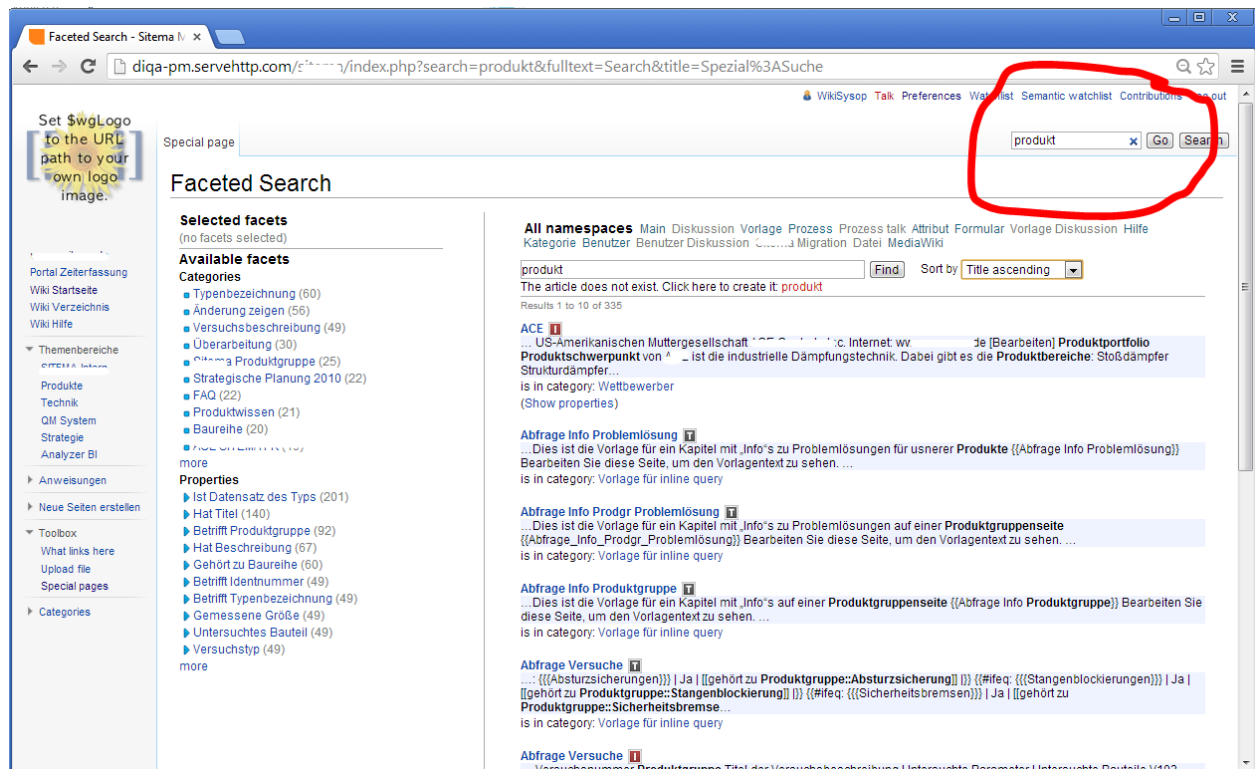
Power-Search employs “faceted browsing”; a technique that lets users explore information based on filters, so called *facets*. It is especially useful for users who are not familiar with the type and structure of the stored information. Searching with this technique is effective and intuitive, as the search results can be refined interactively.

Power-Search determines facets automatically, based on the categories, properties and namespaces of the Wiki. By selecting a facet, users narrow down the set of previously found search results to those that also exhibit this additional facet. The drill-down step can be repeated a couple of time, allowing exploring the wiki contents based on categories and property values.

3.1 Accessing the user interface of Power-Search

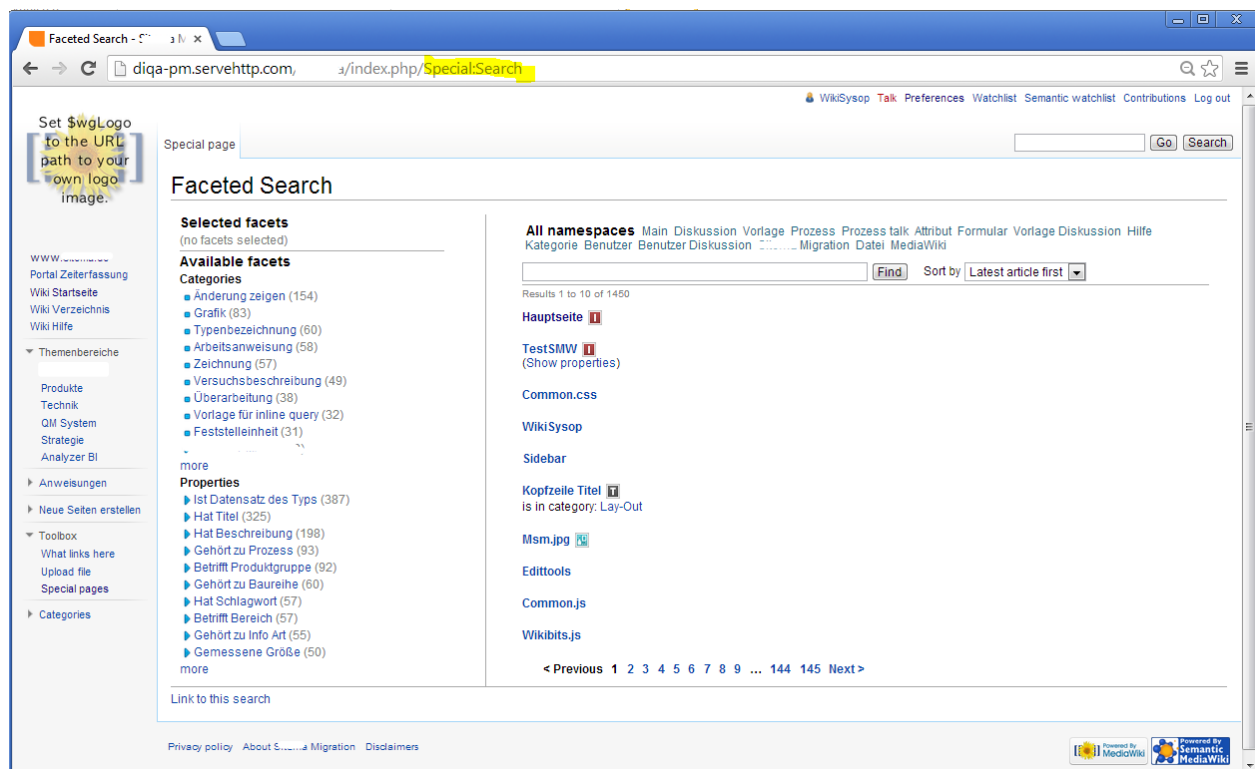
After the installation and configuration of Power-Search, users access the user interface in two ways:

- By pressing the **Search** (or **Go**) buttons in the Wiki's search bar --- with or without a search term:



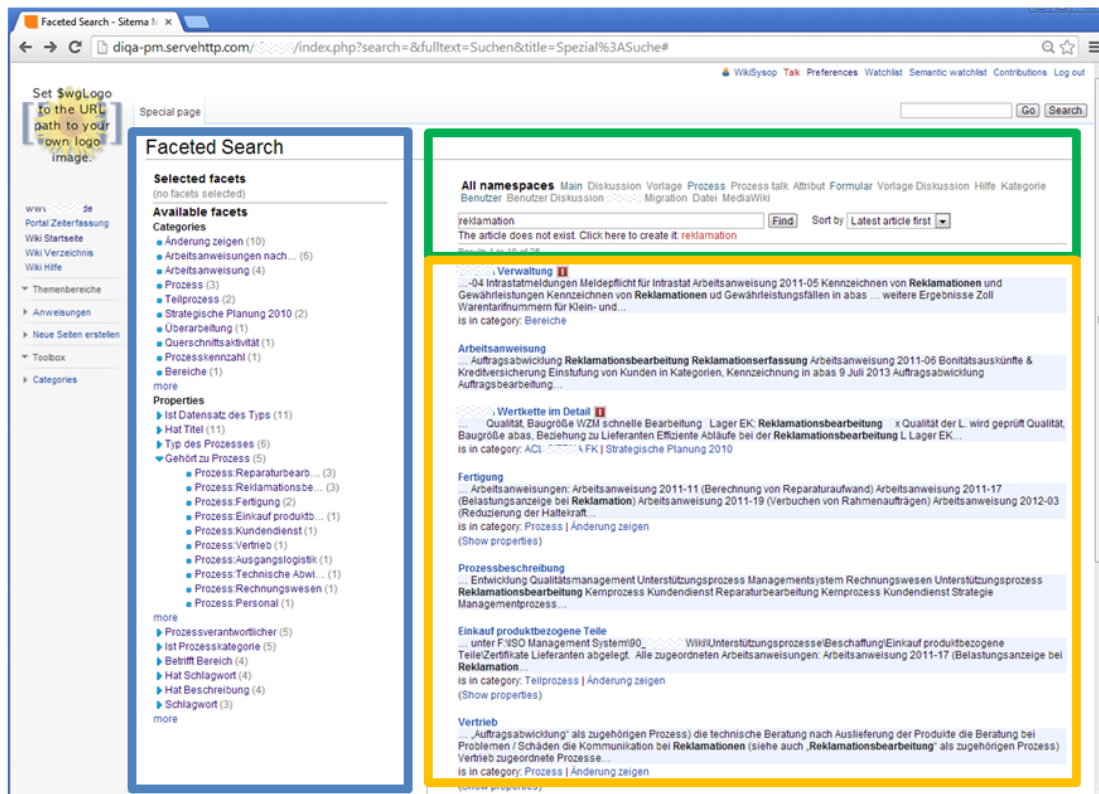
- Or by accessing the special page **Special:Search**.

Both actions will lead the user to the faceted search page:



3.2 Description of the user interface

The user interface of Power-Search consists of three main areas as depicted in the image below:



I - Search Area - Search Term and filters by Namespace

II - Facets Area - Apply Facets as filters

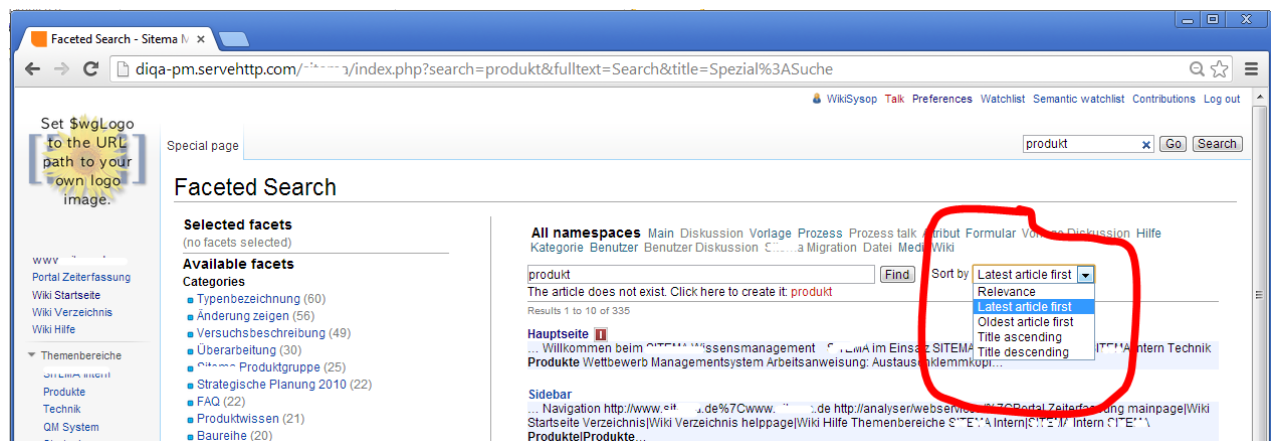
III - Results Area - Results found with selected filters

1. The *search area* contains the search box and filters for available namespace.
2. The *facets area* shows categories and properties (with values). These are the facets that can be applied to filter the results.
3. The *results area* shows the list of search results (after applying all selected facets).

3.3 Search Area

The search area has the following parts:

- **Search text field** - this is where you enter a search term. If a search term is present all wiki pages in the search result must contain this text. If the search text field is empty all the search results will contain all wiki pages (unless other filters are used).
- **Namespaces** - this section lets you limit the results to wiki pages of a single namespace. The selected namespace is in printed in *bold*. The default selection in this section is **All namespaces** as seen in the image below. If the search result contains at least one page from a given namespace, the namespace is shown in blue, otherwise it is greyed out.
- **Sort by drop down list** – Search results can be displayed in different orders.
 - *Relevance* - orders the results according to goodness of the match between search term and wiki page content
 - *Latest article first* - orders the results showing the newest articles at the top
 - *Oldest article first* - orders the result showing the articles oldest articles at the top
 - *Title ascending* - orders the results in ascending alphabetical order
 - *Title descending* - orders the result in descending alphabetical order

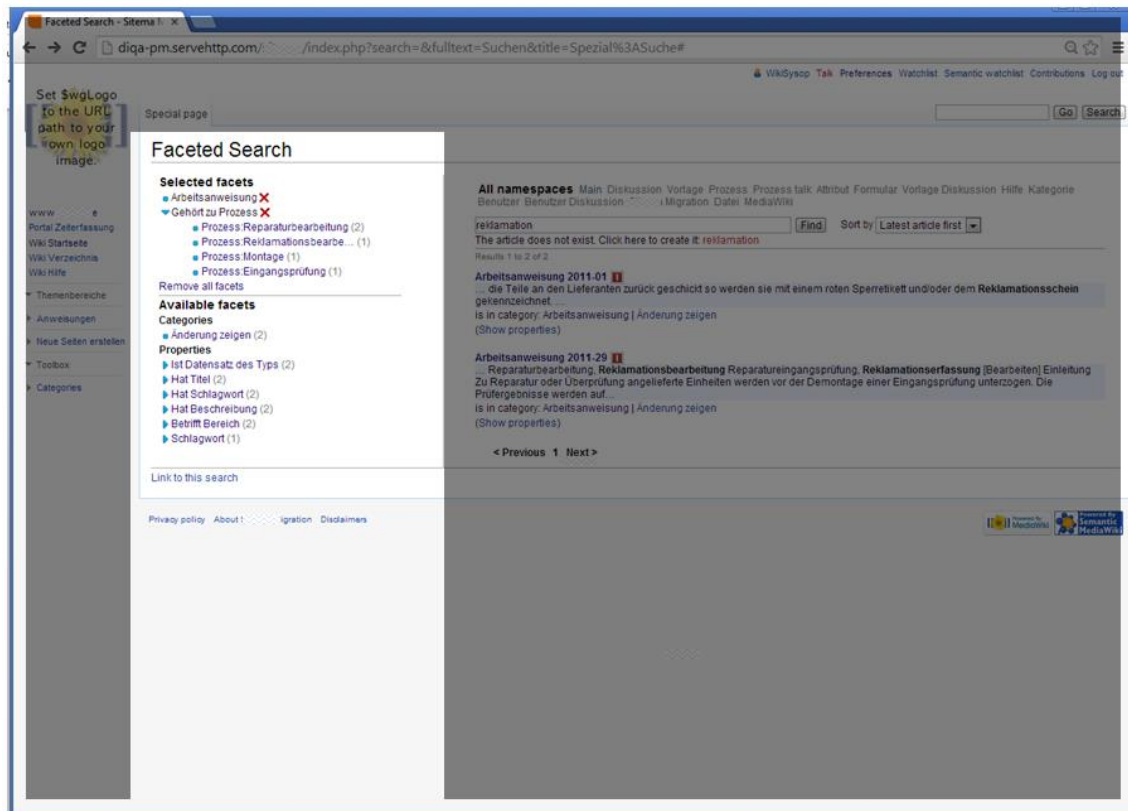


When users enter a character into the text field, the result area and the facets area will be automatically refreshed with every character entered.

After entering a complete term in the search box, users can continue to refine the search by selecting facets.

3.4 Facets Area

The facets area is generated automatically from the **Categories** and **Properties** that are present in the wiki pages that constitute the current search result. The categories, properties and property values can be used to further narrow down the search.



There are two subsections. Initially, the first subsection, titled “Selected facets” will be empty. The second subsection, titled “Available facets”, will list all categories, properties and property values that are present in the wiki pages that constitute the current search result. These categories, properties and property values represent the *facets* and can be used to further narrow down the search. Once a facet is clicked, it is immediately applied to the search result as an additional filter. This results in fewer search results in the right hand side and the selected facet will be moved to the “Selected facets” subsection.

The “Available facets” subsection is organized into categories and properties:

- The list of relevant **categories** is ordered according to the number of occurrences in the search results. The number of wiki pages (in the current search results) that are tagged with a category is given in parentheses after the category name. Initially the top ten categories are displayed. Clicking on **more** will add the next ten items in the category list (if applicable).
- Under “**properties**” a list of relevant properties is given. This list is also ordered by the number of occurrences of a particular property in the search results. The number of wiki pages (in the current search results) that have (any value for) the property is given in parentheses after the property name. Initially the top ten properties are displayed. Clicking on **more** will add the next ten items in the properties list (if applicable).

- Clicking on the triangle in front of a property will show a list of property values. These property values are retrieved exactly from the wiki pages of the current search result. Here also, initially ten different values are listed and the number of occurrences is given in parentheses.

Clicking on a **category** name, a **property** name, or a **property value** initiates a drill down, i.e. will add an additional filter condition to the search (and also to the list of “selected facets”).

Example:

In the screen shot depicted above, the user entered the **search term** *Patrick* and added the following Facets:

- **Category** *Science fiction films*
- **Property** *Actors*
- **Property value for Actor** *Patrick Stewart*

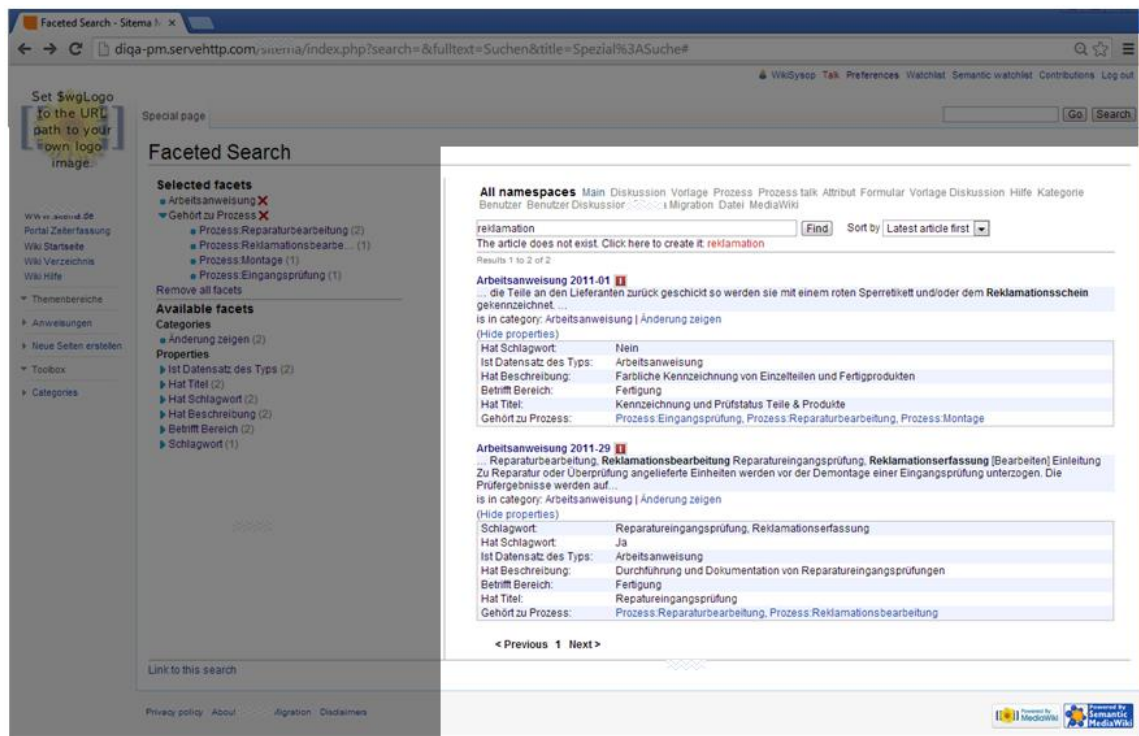
Bookmarking a Search:

At the bottom of the Facets area there is a link labelled “**Link to this search**”. The URL of this link specifies the current search, incl. the search term and all selected facets. This link can be used as a bookmark or can be put on wiki pages to provide a quick link to the search results. Note, that the search results are not stored, i.e. if the wiki is updated with new content, the link will always point to the most current search result for the given facets.

3.5 Results Area

The results area displays a list of search results matching the search term and all selected facets. Every entry in this list represents one wiki page and has the same structure, providing following information:

- **Title** - this is the title of the wiki page. A click on the title link will open the respective wiki page.
- **Icon** – an icon indicating whether the wiki page is an instance, property or category
- **Preview** – a snippet of text from the wiki page, including the search term
- **Categories** - a list of categories of the wiki page. If the page has many categories, only the first few are shown and a more button allows seeing all of them. A click on a category link will open the respective category page.
- **“Show properties” link** – a click on this link inserts a table in this search result. The table shows all properties and their values for this wiki page (cf. screen-shot).
- **Modification date** – the date and time of the last modification of this page is displayed.



3.6 Search syntax

Power-Search distinguishes two different modes of interpretation of the search terms from the search box: *Normal mode* and *expert mode*

3.6.1 Normal mode

In normal mode all entered search terms are interpreted literally, i.e. must occur as entered in the full text of the wiki page.

Exact match

Entering a term like 'Karlsruhe' (without the quotes) will find all pages containing this string.

Prefix match

Entering a term like 'Karl' will find all pages that contain 'Karl' or any term that *starts* with 'Karl', such as 'Karlsruhe'. If the search consists of multiple search terms, only the last one will also match as the prefix of a word in any wiki page; the other terms must match exactly.

Case-insensitive

Search is not case-sensitive, i.e. it does not matter if the search term is 'Karl', 'karl', 'KARL' or 'kaRI'. In any case pages containing 'Karl' or 'Karlsruhe' will be returned.

Phrases

Normal mode also supports phrases, i.e. search terms that contain more than one word, such as "Karlsruhe Durlach" (including the quotes). The words of a phrase must appear on a wiki page in the same order as in the search phrase, in order for the page to be part of the result set.

3.6.2 Expert mode

To switch into expert mode, the search string must be enclosed in **parentheses**, e.g.

```
(Karlsruhe OR Baden-Baden)
```

Expert-mode queries are also case-insensitive and support one-word search terms as well as multi-word phrases. In contrast to the normal mode, prefixes are not matched unless the query contains wildcards.

Wildcard Searches

Use wildcard searches to find wiki pages that contain various possible spellings of the search term. The wildcard is indicated by "*" and can replace any number of characters of a word.

Example: to search for "installation", "installations" or "installer", you can use the search:

```
(install*)
```

You can also use the wildcard searches in the middle of a word.

```
(install*tion)
```

...and even as the first character of a search

```
(*nstallation)
```

Fuzzy Searches

If the correct spelling of a word is not known a fuzzy search could help. Using the tilde "~" character at the end of a single word term will match "similar" words.

Example: to search for a word similar in spelling to "roam", use the fuzzy search:

```
(roam~)
```

This search will find terms such as "foam" and "rooms".

The Boolean Operator "OR"

The OR operator is the default conjunction operator in expert mode. This means that if there is no Boolean operator present, the OR operator is used implicitly. The OR operator links two terms and finds a matching document if at least one of the terms exist in a document. Instead of "OR" the characters "| |" can be used.

Example: to search for documents that contain either "Karlsruhe" or "Baden-Baden" (or both) use the query:

```
(Karlsruhe | Baden-Baden)
```

or:

```
(Karlsruhe OR Baden-Baden)
```

The Boolean Operator "AND"

The AND operator matches wiki pages where both terms exist in the content of a single document. The characters "&&" can be used instead of the word AND.

Example: To search for documents that contain "Karlsruhe" and "Baden-Baden" use the query:

```
(Karlsruhe && Baden-Baden)
```


The Boolean Operator “NOT”

The NOT operator excludes documents that contain the term after NOT. The "!" character can be used instead of the word NOT.

Example: to search for documents that contain "Stadt" but not "Karlsruhe" use the query:

```
(Stadt NOT Karlsruhe)
```

Note:

The NOT operator is a Boolean operator and must connect two terms, i.e. it cannot be used with just one term. For example, “NOT installation” is not a legal expert mode search.

Grouping

Expert mode supports complex queries by combining simpler subqueries. Use parentheses to group a subquery. Subqueries in parentheses can be combined with the Boolean operators, AND, OR and NOT to form more complex queries.

Example: to search for either "installation" or "package" and "software" use the query:

```
((installation OR package) AND software)
```

Note, the outmost pair of parenthesis indicates the expert mode.

3.7 Excursus: Structured Data in MediaWiki

In order to fully benefit from the faceted browsing capabilities provided by Power-Search for MediaWiki the wiki should contain structured data.

Every wiki contains pages with (textual) content, entered by users using wiki markup. The wiki might even contain some documents (PDF, word ...) that have been uploaded by users. In addition to this unstructured information Power-Search can benefit from structured data, if Semantic MediaWiki's (SMW) features are used within the wiki. The introductory article about SMW characterizes these features as follows³:

³ http://semantic-mediawiki.org/wiki/Help:Introduction_to_Semantic_MediaWiki

“While traditional wikis contain only text which computers can neither understand nor evaluate, SMW adds *semantic annotations* that allow a wiki to function as a collaborative database. ...

Semantic MediaWiki introduces some additional markup into the wiki-text which allows users to add "semantic annotations" to the wiki. While this at first appears to make things more complex, it can also greatly simplify the structure of the wiki, help users to find more information in less time, and improve the overall quality and consistency of the wiki."

The role of semantic annotations is crucial for the faceted search that Power-Search provided. These annotations represent the basic data from which Power-Search creates the multi-dimensional search space. Every single property and every single Category can be considered as one dimension in this data-cube.

The online-documentation of Semantic MediaWiki introduces the notion of semantic annotations as follows⁴:

“Annotations in Semantic MediaWiki can be viewed as an extension of the existing system of **categories** in MediaWiki. Categories are a means to classify articles according to certain criteria. For example, by adding `[[Category:Cities]]` to an article, the page is tagged as describing a city. MediaWiki can use this information to generate a list of all cities in a wiki, and thus help users to browse the information.

Semantic MediaWiki provides a further means of structuring the wiki. Wiki pages have links and text values in them, but only a human reader knows what the link or text represents. For example, «*is the capital of Germany with a population of 3,396,990*» means something very different from «*plays football for Germany and earns 3,396,990 dollars a year*». SMW allows you to annotate any link or text on the page to describe the meaning of the hyperlink or text. This turns links and text into **explicit properties** of an article. The property capital of is different from on national football team of, just as the property population is different from annual income.

This addition enables users to go beyond mere categorization of articles. Usage and possible problems with using these features are similar to the existing category system. Since categories and properties merely emphasize a particular part of an article's content, they are often called (semantic) annotations. Information that was provided in an article anyway, e.g. that Berlin is the capital of Germany, is now provided in a formal way accessible to software tools."

⁴ <http://semantic-mediawiki.org/wiki/Help:Editing>

DIQA presented an **introductory tutorial** for using MediaWiki and Semantic MediaWiki at the 2013 SMW conference in Berlin. The slide deck can be retrieved from the conference web site.⁵ Additional helpful links are also collected on the tutorial page.

Even if not a single semantic annotation is used in the wiki, Power-Search will improve the user experience when searching for the textual content from wiki pages (and uploaded documents).

⁵ http://semantic-mediawiki.org/wiki/SMWCon_Fall_2013/Introduction_to_MediaWiki_and_Semantic_MediaWiki

4 Technical Support

4.1 Download new releases of Power-Search for MediaWiki

You find new releases of the Power-Search for MediaWiki in the DIQA-download area:

<http://downloads.diqa-pm.com/free/power-search/>

4.2 Support inquiries

If you have issues with the software then please contact this email address: info@diqa-pm.com.

4.3 Buying a license

The evaluation version of the Power-Search is fully functional for 30 days. If you want to continue to use the Power-Search for MediaWiki after this period has elapsed you must buy a valid license key from DIQA at: info@diqa-pm.com