



Semantic MediaWiki quick reference (1 of 2)

Covers MediaWiki extensions Semantic MediaWiki 1.6, Semantic Result Formats 1.6, Semantic Maps 1.0, Semantic Internal Objects 0.6, Semantic Compound Queries 0.3 and Semantic Forms 2.3

Properties

Special pages: Types, CreateProperty, Properties, UnusedProperties, Browse, SearchByProperty

Defining a triple:

```
[[property-name::value]]
{{#set:property-name=value}} (silent definition)
```

Defining a property:

```
[[Has type::type-name]]
[[Allows value::enum-value]] (for enumerations)
```

Property types: Page (default), String, Text, Code, Number, Date, Boolean, URL, Email, Telephone number, Temperature; *from Semantic Maps:* Geographical coordinate

Defining an internal object:

```
{{#set_internal:object-to-page-property|prop1=val1|prop2#list=val2|...}}
```

Example (on a page called "United States"):

```
{{#set_internal:Is president of|Has name=James Madison|Has start year=1801|Has end year=1809|Has vice president#list=George Clinton, Elbridge Gerry}}
```

Defining a recurring event:

```
{{#set_recurring_event:property=date property|start=start date|end=end date|period=number|unit={day|week|month|year}|week number=number|include=additional dates|exclude=excluded dates}}
```

Notable settings for LocalSettings.php

```
$smwgLinksInValues - set to true to allow wiki-links within property values
$smwgNamespacesWithSemanticLinks - all namespaces that can hold semantic values
$smwgShowFactbox - set to SMW_FACTBOX_NONEMPTY to display factbox at bottom of pages
$smwgInlineErrors - set to false to hide SMW error messages
```

Semantic templates

Special pages: CreateTemplate, Templates

Simple semantic template definition:

```
Field-label: [[property-name::{{{field-name}}}}]
[[Category:category-name]]
```

Alternate declaration of property in template:

```
{{#declare:property-name=field-name}}
```

Setting a field to represent multiple property values:

```
{{#arraymap:value|delimiter|var|formula|new_delimiter}}
{{#arraymaptemplate:value|template-name|delimiter|new_delimiter}}
```

Inline queries

Special pages: Ask

Syntax: `{{#ask:parameter1|parameter2|...}}, {{#show:page-name|parameter1|...}}`

Standard parameters:

<i>query conditions</i>	sort=	default=
?property-name=label	order={asc desc random}	intro=
+property-param=value	headers={show hide plain}	outro=
limit=	mainlabel=	searchlabel=
offset=	link={none subject all}	format=

Query conditions: `[[Category:category-name]]... [[Concept:concept-name]]... [[property-name::value]]... [[page-name]]`

Subquery: `[[prop1.prop2.prop3::value]]`

Inverse property: `[-property-name::page-name]`

"Or" conditions: `[[Category:category-name]] OR [[property-name::value]], [[property-name:value1||value2]], [[page-name1||page-name2||page-name3]]`

Comparators: `::` (equals), `::<` (less than), `::>` (greater), `::~` (like), `::!` (not), `::!~` (not like)

Property parameter types: limit, order, align, index

Format types: list, ol, ul, table, broadtable, category, rss, csv, json, embedded, template, count, debug; *from Semantic Result Formats:* timeline, eventline, calendar, graph, process, googlebar, googlepie, jqplotbar, jqplotpie, exhibit, outline, tagcloud, gallery, sum, average, min, max, bibtex, vcard, icalendar; *from Semantic Maps:* map, googlemaps, yahoo maps, openlayers

Notes on formats:

- list, ol, ul, category, template and calendar formats allow the 'template' parameter
- sum, average, min and max require 'limit' to be greater than number of queried pages
- for calendar, use #calendarstartdate and #calendarenddate to limit to current month

Sample query: To create a table of all cities in Europe and their populations, sorted by population:

```
{{#ask:mainlabel=City|[[Category:Cities]]|[[Has country.Has continent::Europe]]?Has population=Pop.|sort=Has population}}
```

Running a compound query:

```
{{#compound_query:query1|query2|...|joint parameters}}
Each query's parameters should be separated by semicolons, instead of pipes.
```

Sample compound query: to show a map with one icon for clothing stores and another for fast food restaurants:

```
{{#compound_query:[[Category:Clothing stores]]?Has coordinates;icon=Shirt.png|[[Category:Restaurants]]|[[Has cuisine::Fast food]]?Has coordinates;icon=Hamburger.png|format=googlemaps|height=400|width=600}}
```

Notable settings for LocalSettings.php

```
$smwgQDefaultLimit - default number of results displayed on a page; default is 50
$smwgQMaxInlineLimit - maximum results displayed on a page; default is 500
$smwgQMaxLimit - maximum results retrieved for a query; default is 10000
```

Concepts

Defining a concept: In the 'Concept' namespace:

```
{{#concept:query-conditions|description}}
```



Semantic MediaWiki quick reference (2 of 2)

Covers MediaWiki extensions Semantic MediaWiki 1.6, Semantic Forms 2.3, Semantic Drilldown 1.0, External Data 1.3 and others

Semantic Forms

Special pages: CreateForm, Forms, CreateCategory, CreateClass, RunQuery

Simple form definition:

```

{{{for template|template-name}}
field-label: {{{field|field-name}}}
{{end template}}
Free text:
{{{standard input|free text}}}
{{{standard input|save}}}

```

Form tags:

info field
for template standard input
end template

'info' parameters:

create title= partial form
edit title= page name=
query title= query form on top

'for template' parameters:

label= strict
multiple add button text=
embed in field=

'field' parameters:

input type= hidden
class= mandatory
list restricted
delimiter= default=
property= preload=
holds template
...for text inputs:
size=, uploadable, image preview,
default filename=
...for textarea inputs:
autogrow, rows=, cols=, editor=
...for 'category' and 'categories':
top category=, height=, width=
...for enums and autocomplete:
values=, values from {property|
category|concept|namespace}=
...for enums: show on select=
...for autocomplete: remote
autocomplete, values from url=

'standard input' types:

free text preview
minor edit changes
watch cancel
summary run query
save save and continue

'standard input' parameters:

label= class=

Input types

Field type	Allowed input types (default is bolded)
Page	text with autocomplete , combobox, text
String, URL, Number, etc.	text , text with autocomplete, combobox
Text, Code	textarea , text
Date	date , datetime, year
Enumeration	dropdown , radiobutton
Boolean	checkbox
List of Page	text with autocomplete , text, textarea (with autocomplete)
List of String, etc.	text , text with autocomplete, textarea (with autocomplete)
List of enums	checkboxes , listbox

More input types: *from Semantic Maps*: map, googlemaps, yahoomaps, openlayers; *from Semantic Forms Inputs*: datepicker, timepicker, datetimestepicker, menuselect, regexp; *from Semantic Image Input*: instantimage; *using CategoryTree*: category, categories

Form input:

```

{{{#forminput:form=|size=|default value=|button
text=|query string=|popup|autocomplete on
category=|autocomplete on namespace=
remote autocomplete}}}

```

Form input query string options:

(options are separated by '&')
template_name[field_name]=
namespace= super_page=

Setting the 'edit with form' tab:

(in a namespace page (*project-namespace*:
namespace-name) or category page)
[[Has default form::form-name]]
(in a regular or template page)
[[Page has default form::form-name]]

Pointing red links to forms:

(in a property or namespace page)
[[Has default form::form-name]]
(in a property page)
[[Has alternate form::form-name]]

Forms that set an automatic page name

In form definition:

```

{{{info|page name=...<template-name|field-
name> ... <unique number>}}}

```

Linking to the form:

```

{{{#formlink:form=|link text=|link type=|query
string=|target=|popup}}}

```

Making red links create pages automatically:

(in a property page)
[[Creates pages with form::form-name]]

A link to edit a page automatically:

```

{{{#autoedit:form=|target=|link text=|link type=|query
string=|reload}}}

```

Semantic Drilldown

Special pages: CreateFilter, Filters, BrowseData

Defining a filter:

[[Covers property::property-name]]
[[Has value::value]]
[[Use time period::Year|Month]]
[[Gets values from category::category-name]]
[[Has input type::(combo box|date range)]]
[[Requires filter::Filter:filter-name]]
[[Has label::label]]

In a category page:

[[Has filter::Filter:filter-name]]
[[Has display parameters::parameters]]
'parameters' should have same structure as #ask
query parameters, but separated by semicolons
__HIDEFROMDRILLDOWN__
__SHOWINDRILLDOWN__

Notable settings for LocalSettings.php

\$sdgFiltersSmallestFontSize, \$sdgFiltersLargestFont
Size - font sizes for "tag cloud" display of filter values

External Data

Getting data from a URL:

```

{{{#get_web_data:url=URL|format=format|data=data-
mappings|filters=filters}}}
Allowed formats: CSV, JSON, XML, GFF

```

Getting data from a database:

```

{{{#get_db_data:server=server_id|from=from-clause|
where=where-clause|data=data-mappings}}}

```

Getting data from an LDAP server:

```

{{{#get_ldap_data:domain=domain-id|filter=ldap-filter|
data=data-mappings}}}

```

"data", "filters" values should be separated by commas.

Displaying external data:

```

{{{#external_value:variable_name}}
{{{#for_external_table:expression}}}
The expression value in #for_external_table should
contain variables of the form {{{variable_name}}}.

```

Storing external data - examples:

Storing an external value with SMW:

```

[[Has capital::{{{#external_value:capital}}}]

```

Storing a table of external values (requires Semantic
Internal Objects extension):

```

{{{#store_external_table:Is leader of|Has
name={{{name}}}|Has start year={{{start year}}}|Has
end year={{{end year}}}}}

```

Notable settings for LocalSettings.php

\$sfgRenameEditTabs - set true to change 'edit with form' to 'edit', 'edit' to 'edit source'
\$sfgRenameMainEditTab - set to true to change 'edit' to 'edt source'
\$wgAmericanDates - set to true to show month first in date input
\$sfg24HourTime - set to true to display time input in 24-hour notation