

# HelpOnConfiguration

- [UserPreferences](#)
- [SurgeProtection](#)
- [SecurityPolicy](#)
- [FileAttachments](#)
- [EmailSupport](#)
- [SupplementationPage](#)

# UserPreferences

Please edit system and help pages  
ONLY in the master wiki!

For more information, please see  
MoinMoin:MoinDev/Translation.

##master-page:Unknown-Page ##master-date:Unknown-Date #acl -All:write Default #format wiki  
#language en = User Preferences = You can predefine, disable or remove several options on the  
user preferences page using the following parameters in your wiki config:

```
||<tablewidth="100%"rowbgcolor="#ffffcc"25%>""Variable"" ||""Default"" ||""Description"" || ||
user_checkbox_defaults || dict || Sets the default settings of the user preferences checkboxes.
Example: user_checkbox_defaults = {'edit_on_doubleclick': 0} || || user_checkbox_disable || [] || a
list of checkbox names to be disabled in the user preferences. A disabled checkbox is displayed
greyedout and uses the default value set in user_checkbox_defaults. || || user_checkbox_fields ||
[...] || list of checkbox items. || || user_checkbox_remove || [] || a list of checkbox names to be
removed from the user preferences. A removed checkbox uses the default value set in
user_checkbox_defaults.|| || user_form_defaults|| [] || a list of form defaults for user preferences.||
|| user_form_disable || [] || a list of form field names to be disabled in the user preferences. A
disabled field is displayed greyed out and uses the default value set in user_form_defaults.|| ||
user_form_fields || [...] || list of user preferences form items. || || user_form_remove || [] || a list
of form field names to be removed from user preferences. A removed form field uses the default
value set in user_form_defaults.||
```

The possible keys / clear text explanations are found in `MoinMoin/config/multiconfig.py`:

```
||<tablewidth="100%"rowbgcolor="#ffffcc"25%>""Variable"" ||""Description"" || || mailto_author ||
Publish my email (not my wiki homepage) in author info || || edit_on_doubleclick || Open editor on
double click || || remember_last_visit || After login, jump to last visited page || || show_comments ||
Show comment sections || || show_nonexist_qm || Show question mark for non-existing pagelinks ||
|| show_page_trail || Show page trail || || show_toolbar || Show icon toolbar || || show_topbottom ||
Show top/bottom links in headings || || show_fancy_diff ||Show fancy diffs || || wikiname_add_spaces
```

```
|| Add spaces to displayed wiki names || || remember_me || Remember login information || ||  
want_trivial || Subscribe to trivial changes || || disabled || Disable this account forever ||
```

So you could do, for example:

```
user_checkbox_remove = ['edit_on_doubleclick', 'show_nonexist_qm', 'show_toolbar',  
'show_topbottom', 'show_fancy_diff', 'wikiname_add_spaces', 'remember_me', 'disabled',]
```

```
user_form_remove = ['password', 'password2', 'css_url', 'logout', 'create', 'account_sendmail',  
'aliasname',]
```

# SurgeProtection

Please edit system and help pages  
ONLY in the master wiki!

For more information, please see  
[MoinMoin:MoinDev/Translation](#).

##master-page: ##master-date: #acl -All:write Default #format wiki #language en = Surge Protection = <<TableOfContents>> Especially on an internet wiki sometimes the cpu and disk load gets quite high because the wiki has to process many requests. A high load makes the wiki slow or unusable for its legitimate users.

This is caused by:

- some bad guys (spammers, vandals, ...) trying to put spam or trash on your pages
- some less-bad guys (people just "mirroring" the wiki, without thinking about the load and the lots of useless requests and high traffic they cause)
- some bots, indexing the wiki content for some search engine

== What can we do? == MoinMoin tracks requests by user name (if logged in) or IP address (if not logged in).

If the request count goes beyond some configured limit in some configured time interval, it will start to warn you (you will see a surge protection warning message instead of the expected wiki output). If you see this warning, just read it and do what it tells.

You should stop doing those requests for a minute in that case (just read the stuff you already got, drink some water, ...). You will have no problems if you continue a while later (just slow down a bit, so you won't go beyond the limit again).

If you don't pause and ignore the warnings and the requests don't stop hammering the wiki server, the wiki will think you are doing something nasty and will lock you out for some configurable amount of time.

== Configuration == Configuration is done as usual: in your wikiconfig. These are the built-in default values:

```
{{#!python surge_action_limits = { # allow max. requests per  
secs # Format: # action: (count, dt) 'all': (30, 30), # all requests (except cache/AttachFile action)  
count for this limit 'default': (30, 60), # default limit for actions without a specific limit 'show': (30,  
60), 'recall': (10, 120), 'raw': (20, 40), # some people use this for css 'diff': (30, 60), 'fullsearch':  
(10, 120), 'edit': (30, 300), # can be lowered after making preview different from edit 'rss_rc': (1,  
60),
```

```
    # The following actions are often used for images - to avoid pages with lots of images  
    # (like photo galleries) triggering surge protection, we assign rather high limits:  
    'AttachFile': (300, 30),  
    'cache': (600, 30), # cache action is very cheap/efficient  
  
    # The following actions are often abused by spambots, trying to create and login into  
accounts:  
    'newaccount': (4, 4000),  
    'login': (4, 400),  
  
}  
surge_lockout_time = 3600 # secs you get locked out when you ignore warnings
```

```
}}}
```

== Hints == === Big proxy or firewall === If you happen to be behind a big proxy or firewall (china? big company network?) you might get trouble with surge protection because all requests come from a single IP address, looking like a really nasty user or bot.

But you can easily work around that by just logging in, so it really counts "your" requests (it uses your name in that case to count requests, not your proxy/firewall IP).

You will also have no problem, if your proxy uses the 'x-forwarded-for' header with your real IP.

=== Disabling surge protection === (!) It is not advised to do that. If it triggers too fast, adjust the limits to your needs. Even in an intranet wiki, it is no good if a single user can make the wiki unusable for everybody by firing up some leech tool at it.

If you want to do it nevertheless:

```
{{{ surge_action_limits = None # disable surge protection }}}
```

# SecurityPolicy

Please edit system and help pages  
ONLY in the master wiki!

For more information, please see  
MoinMoin:MoinDev/Translation.

```
##master-page:Unknown-Page ##master-date:Unknown-Date #acl -All:write Default #format wiki  
#language en = Security Policy Configuration = <<TableOfContents>>
```

`SecurityPolicy` is a config option that allows wiki admins to dynamically enable or disable certain key actions in a MoinMoin wiki, most notably editing and deleting content. See also `HelpOnAccessControlLists` for an easier way.

== Mechanics ==

Security restrictions in a MoinMoin wiki work by the interface defined in the `MoinMoin.security` module. The `Permissions` class implements the basic interface for user permissions and system policy. If you want to define your own policy, inherit from that base class, so that when new permissions are defined, you get the defaults.

Then either assign your new class to `Config.SecurityPolicy` in `wikiconfig.py` (and I mean the class, not an instance of it), or define your class as a member of `class Config` in `wikiconfig.py` and name it `SecurityPolicy`.

== Anti-Spam protection ==

If you enable the "antispam" utility your wiki will fetch the page `BadContent` from `MoinMaster:BadContent` and keep it automatically in sync. Do not edit it, because your edits will be overwritten.

```
{{{ # add this within Config class, be careful with indentation: from MoinMoin.security.antispam  
import SecurityPolicy }}}}
```

## == SSL-only Write ==

If you want for security reasons that users do not edit the wiki without using SSL you should add:

```
{{{ # add this within Config class, be careful with indentation: from MoinMoin.security import
Permissions class SecurityPolicy(Permissions): def write(self, pagename): # make wiki only
writeable via SSL return (self.request.is_secure and Permissions.write(self, pagename) }}}}
```

## == Samples ==

For samples on using this mechanism, see [MoinMoin:SecurityPolicy](#).

## == See also ==

- [HelpOnAutoAdmin](#) when you want to give some users or user groups admin rights on some pages.

# FileAttachments

Please edit system and help pages  
ONLY in the master wiki!

For more information, please see  
MoinMoin:MoinDev/Translation.

```
##master-page:Unknown-Page ##master-date:Unknown-Date #acl -All:write Default #format wiki  
#language en = File Attachments Configuration =
```

The `[[HelpOnActions/AttachFile|AttachFile action]]` enables a page to have multiple attached files. Since file uploads could be abused for DoS (Denial of Service) attacks, `AttachFile` is an action that may be disabled by the wiki administrator (by adding `actions_excluded = ['AttachFile']` to `wikiconfig`).

This is all you usually need to do for configuration.

== How attachments are handled ==

Attachments are stored "privately" in the `data/pages/` directory (with paths like `data/pages/<pagename>/attachments/<filename>`) and can only be retrieved by using the `AttachFile` action (via URLs like

```
http://example.org/mywiki/<SomePage>?action=AttachFile&do=get&target=filename.ext).
```

`direct serving method` (deprecated since long) was finally removed in moin 1.7.0. If you have used it until now (`attachments = {...}`), please copy your attachments into your data directory (and make sure they arrive at the right location there).

# EmailSupport

```
mail_import_secret = u"do_not_choose_a_too_short_secret_or_it_wont_work"
```

## The target URL

```
mail_import_url = u"http://localhost:81/?action=xmlrpc2" }}}
```

After you have configured the settings explained above, you should be able sending yourself mails. The actual usage is explained below.

From :: We check the From: header if it is from a valid wiki user. Of course this is no real security, but maybe better than nothing.

:: When we create a mail overview table, we use the email address part of From: to try to lookup a wiki user account with that email address. In case we find some, we will link to his homepage from the From: column of the table. If we don't find a matching user account, we just list the realname part in the From: column.

To / Cc :: To: is (as usual) made of 2 parts, like `TEMPLATE <wiki@example.org>`. We use the "realname" part to make up the target page name template (can contain special chars, blanks etc.) and the email address part to get it to the right destination. The simplest form of a target page name template is just a pagename like WikiSandBox.

:: See the example for the 2 different modes of operation.

:: When we create a mail overview table, we process To: and Cc: for the To: column in the same way we do it for the From: column (see above). We do not include the address of the wiki server (and if we only have the wiki server email, we just display "-").

Subject :: The page name template can also be extracted (and also removed) from the first occurrence of mail\_import\_pagename\_regex within the subject.

:: Examples: {{{ Subj: [[TargetPageName]] will work if you put it either at the beginning or the end of the subject. Subject: bla bla bla [[PageName/\$date]] }}}}

:: The text of mails with such subject will be added to the pages {{{ TargetPageName or PageName/2006-07-20T14:48:27 }}}}

:: The full subject will also be used to build the change comment, prefixed with "`Mail:`"; variables are not replaced there.

Mail body :: The new page content is made from old page content (if any) plus the content of the email appended at the end. Page content will never get overwritten. Mails that are sent as `alternative` are shown with their HTML body on the page which is processed by a HTML parser.

:: E-Mail attachments will get put into page attachments under the page where the content will be put. Attachments will never get overwritten. The attachments will be linked from the content page (e.g. from a list of attachments at the end).

Generated content :: If requested by a leading `+`-sign (e.g. `+ PageName/`), Moin adds a table item to the parent page listing details about the e-mail.

Pagename generation :: The placeholders `$from` (containing the name of the sender), `$date` (the date) and `$subject` (subject line) are supported.

:: The pagename is generated like explained below:

1. Check if one of the mail target addresses (to, cc, ...) is in `mail_import_wiki_addrs`
2. Process TO/SUBJECT in order as given by `mail_import_pagename_search` until we have found a pagename or have to give up:

- Try to get the pagename template from the target addrs, or
- Try to extract the pagename from the subject (explained above).

1. If we didn't find a pagename, use `mail_import_subpage_template` as last resort.
2. Apply `mail_import_pagename_envelope` to (optionally) extend the pagename.
3. If the pagename ends with a slash, append the configuration setting `email_subpage_template` (see above) to the pagename.
4. Expand the pagename template to get the actual pagename.

# SupplementationPage

Please edit system and help pages  
ONLY in the master wiki!

For more information, please see  
[MoinMoin:MoinDev/Translation](#).

```
##master-page:Unknown-Page ##master-date:Unknown-Date #acl -All:write Default #format wiki
#language en = Supplementation Page = Supplementation pages are just additional pages that
can be associated with your normal wiki pages by a link in the theme.
```

One usage of this feature is to have one "discussion page" in addition to each of your wiki pages. Then each wiki page will have a link to the "discussion page" by default.

== Configuration by the wiki admin ==

A wiki administrator can choose to enable this feature for all pages by putting this into wikiconfig: {{{ supplementation\_page = True # default is False }}}}

The name of the supplementation page and the template used for these pages can be set like this: {{{ supplementation\_page\_name = u'Discussion' # this is the default supplementation\_page\_template = u'DiscussionTemplate' # this is the default }}}}

== Control by the wiki user == A wiki user can control if he wants to have the supplementation feature on a per-page basis by using a processing instruction at the top of the page content: {{{ #pragma supplementation-page on }}}}

Usage with "'off'" disables the feature on that page.

== Hints for using supplementation pages ==

- Clean pages vs. good context:
- You can use this to keep your wiki pages clean from discussions or comments (they will be put on the supplementation page).

- Separating discussions / comments also has some negative impact as you will have to provide context. If you don't do this, it might be unclear to what you are referring. If you could just comment "near" the context you are commenting, it would be less clean, but somewhat clearer/easier.
- There is only one "fixed" supplementation\_page\_name (default: 'Discussion'):
- If you just have a single language in your wiki, this is no big issue, you just set that page name to whatever you want. Either keep the default english name 'Discussion' or choose it wisely.
- If you have multiple languages in your wiki, it is recommended to use a pagename that everybody can understand and be happy with (e.g. the english name 'Discussion', just keep the default).

There is also another, quite different "commenting feature" described on [HelpOnComments](#).