BSCW 5.0 Administration Manual

BSCW Version 5.0.9

September 2014

Contents

1 How to read this Manual

Before installing your BSCW server you should read at least:

- the introduction to section 2 *Installation of the BSCW Server* (in particular section 2.3 *Upgrading* if you are upgrading an BSCW instance,
- either section 3.2 *Installation procedure for Unix* or section 4 *Installation procedure for Windows 7, 8/Server 2012, 2008, 2003*, depending on the operating system you are using.

This should be sufficient to install the BSCW server and carry out the initial configuration of the server. If you have problems with the installation and initial configuration process, you should read

- the respective sub-section of section 3.2 or 4, depending on your operating system or Web server,
- the frequently asked questions (FAQ) section 10.

In general, this should give you enough information to get your BSCW server up and running.

The BSCW server is initially equipped with a license which allows usage and testing of the BSCW server for a trial period of 90 (ninety) days. After 90 days, the BSCW server is no longer fully usable (except for a few fundamental operations such as the upgrade license operation). Therefore, if you decide to use the BSCW server for a longer period, you need to acquire a license. The acquisition of licenses is described in section 9 *BSCW License*. If you have problems when upgrading your BSCW license, you should also have a look at the respective entries in the frequently asked questions in section 10 or contact OrbiTeam (support@orbiteam.de).

The BSCW server has a considerable number of configuration options. If you have gained some experience with usage of the BSCW system you should read section 5 to find out what configuration options are available and whether they could be used to satisfy the requirements of your users better than the default settings as specified in the code you downloaded. Additionally you may want to enable some extra packages, see section 6 for a detailed description of the additional BSCW functionality provided in package extensions.

In general, the administrative overhead for running a BSCW server is very low. In fact, you may install and configure your BSCW server such that you practically never need to bother with administration. Most likely, however, sooner or later you may have questions such as "How many users are registered at my server?", "How can I rename or delete a user?", or " How can I restrict the creation of workspaces?" Answers to such administrative questions can be found in section 7 *Administration of BSCW Servers* and in section 10.3 of the frequently asked questions section.

2 Installation of the BSCW server

As a prerequisite for installing a BSCW server you need either a server host running a Unix system - the BSCW server can be installed on a large variety of Unix systems, including Linux, Solaris, HP-UX, AIX, MacOS X and BSD - or a server host running Microsoft Windows 7, 8/Server 2012, 2008, 2003.

2.1 General Requirements

The hardware requirements depend largely on the number of users that are expected to use the system. In general, the hardware requirements are not particularly high. For example, a dual core >2,5 GHz Intel Core/Xeon or AMD Athlon/Phenom/Opteron with 4 GB RAM and >100 GB disk space should provide an environment with satisfactory performance for about 200 users.

The BSCW server is an extension of a Web Server with the respective BSCW functionality. The extension is implemented through the CGI interface, which is supported by almost all Web servers. The BSCW software is written in Python (see the Python web site at http://www.python.org/). Therefore, besides the BSCW software, the installation of the BSCW server requires

- a Web Server
- a Python interpreter
- (optional) extensions for Python (lxml, PyLucene, Python-LDAP)

The BSCW server can be installed on a CGI 1.1 compliant Web server, e.g. the Apache HTTP server or the Microsoft's Internet Information Server (IIS) (we recommend the Apache HTTP server version 2.2 or 2.4). Python interpreters are freely available from the Python web site (http://www.python.org/). We currently support version 2.7 or 2.6 of the Python interpreter.

After installation the BSCW server needs to be configured. Only very few configuration efforts are required as a minimum since a few variables (e.g. the email address of the system administrator of the BSCW server) need always be set individually. The server offers a large number of configuration options but we recommend that initially a BSCW system administrator uses the default settings, except for those options which need to be configured as a minimum.

The installation process is different between Unix systems and Windows 7, 8/Server 2012, 2008, 2003. Therefore, the installation process for Unix and Windows is described separately in section 3.2 and section 4, respectively. You need only read one of the two sections, depending on your platform.

The configuration process is to a large extent identical for Unix and Windows. Whenever a difference is necessary, this is described at the respective places in this manual.

Note: Please also **consult the frequently asked questions section** in this manual - or the on-line version at http://www.bscw.de/english/faq.html - for common and platform-specific installation questions; if you have a problem not addressed there, send an email to support@orbiteam.de.

2.2 Security considerations

New installed BSCW instances do have the following possibly security relevant features enabled per default:

1. *Enabled user self-registration*
   A newly installed BSCW instance allows every registered user to create a new user account by registering a new email address. This is probably not in all situations the desired behavior. If you do not want to allow the self-registration of new user accounts by registered users you

have to disable this feature by setting in the instance configuration file (`<bscw-runtime-path>/conf/config.py`) the directive `MAY_REGISTER` to a non empty list. See the directive description in the instance configuration file for details.

2. *Enabled web services*
   BSCW offers a range of services via different web service protocols: XML-RPC, JSON, SOAP. Basically most of the actions available on the user interface (like "add folder") are accessible via a web service API. Of course access to API is restricted via access control as in the regular user interface (i.e. authentication and BSCW internal roles and rights are respected). The availability of the web service API on different user levels can be configured by editing the `ACCEPT_WEBSERVICES` directive in the instance configuration file (`<bscw-runtime-path>/conf/config.py`). See in section 5.2.7 the description of the `ACCEPT_WEBSERVICES` directive on page 80 for details.
3. *Enabled "public space"*
   By default BSCW allows users in the "manager" role to publish the contents of a folder in a "public space", which can be accessed by everyone over the World-Wide Web without being a registered user of the server. Recently misuse of this feature was reported (users published inadequate contents). To disable the public space for all users see section 7.5.
4. *Environment with credential information (Unix)*
   Depending on the authentication method the user credentials are passed via an environment variable (Basic/Cookie authentication) in plain text to the `bscw.cgi` process. Even if the credential information is removed immediately from the environment this might impose a security problem on systems running other applications with the user-id of the Apache web server. In this case such an application may disclose user names and passwords form the environment of a running `bscw.cgi` process.

2.3 Upgrading to BSCW 5.0.9

If you are installing BSCW for the first time please refer to the installation sections (3.2 Unix, 4.2 Windows). For upgrading, you essentially proceed the same way as shown in the installation section.

Notes:

- If you are using the Apache HTTP server you must **restart** the web server after each upgrade.
- It is possible to upgrade your Python to a supported version (Python 2.6 or Python 2.7) before a BSCW upgrade.
- To list all installed BSCW instances on the installation host run `bsadmin manage_servers -l`:

```
$ ./bin/bsadmin manage_servers -l
/opt/bscw/srv/bscw.domain.org:  BSCW 5.0.8

> bin\bsadmin manage_servers -l
c:\bscw\srv\bscw.domain.org:    BSCW 5.0.8
```

Substitute `<bscw-runtime-path>` by your actual BSCW instance installation path. However, please take note of one or more of the following points which might apply to your situation:

Please make a backup of your current BSCW data before you upgrade your BSCW Server.

DO NOT UPGRADE

If your current license is invalid (e.g. license expired, wrong host). Upgrading of BSCW with an invalid license will fail. Please obtain and install a new valid license first. Contact license@orbiteam.de for details.

DO NOT UPGRADE

If your license does not include free upgrades. (If you have a *time-unlimited* license, i.e. a license which does not expire, your license does NOT include free upgrades.) Upgrading of BSCW will invalidate your existing license key and will result in an inoperable BSCW system. Contact license@orbiteam.de for details.

Consider the following advices when upgrading and refer to the "Upgrading on Unix" resp. "Upgrading on Windows" section below:

When **upgrading from BSCW 5.0.7 or lower**

A security vulnerability has been discovered, which may disclose the name of objects stored in BSCW. Thanks to RedTeam Pentesting (https://www.redteam-pentesting.de/) for identifying this problem, see CVE-2014-2301 for details. Please upgrade all BSCW instances to at least version 5.0.8.

The *MoinMoin* Wiki integration (`moin`) has been adopted to support Apache HTTP server version 2.4. The default location of the *MoinMoin* instances was moved from `<bscw-runtime-path>/var/data/moin` to `<bscw-runtime-path>/moin`. To preserve all existing *MoinMoin* Wikis the contents from `<bscw-runtime-path>/var/data/moin` must be moved to `<bscw-runtime-path>/moin`. Since the file layout has been changed `bin/bsadmin conf_moin -r` must be run to update the configuration. The newly generated configuration file `conf/apache{2,24}/moin.conf` must be included into the Apache HTTP server site-configuration (in place of the old `var/data/moin/apache.conf` file).

When **upgrading from BSCW 5.0.6 or lower**

Support for the Apache HTTP server version 2.4 has been added.

When **upgrading from BSCW 5.0.4 or lower**

The BSCW *ldap* package has been updated to map the BSCW 5.0 user meta data to LDAP attributes. If you use the BSCW *ldap* package, please adopt your configuration file located in `<$HOME>/srv/<bscw-runtime-dir>/conf/ldap/config_ldap.py` according to the changes of the new default configuration file template `<$HOME>/lib/bscw-5.0.?-3????-py2?/conf/ldap/config_ldap.py`. (Note: the `update_bscw` directive has been converted from a tuple to a dictionary).

When **upgrading from BSCW 5.0.3 or lower**

Please note BSCW 5.0 triggers a bug in the RHEL 6 or derivatives (CentOS/Scientific Linux) Python 2.6 interpreter, which causes crashes. This problem has been fixed by Red Hat beginning with version 6.4. So it is advisable first to upgrade to version 6.4 (or higher). Another workaround is to compile and install a Python 2.7 environment in `/usr/local` (see 10.2.4 on page 164).

When **upgrading from BSCW 4.5.9 or lower**

When upgrading to BSCW 5.0 the file system layout is automatically transformed to the new BSCW 5 layout (see installation section for details). **Note**: the upgrade procedure checks all existing packages and disables outdated or non-working packages. The resulting enabled packages are sorted alphabetically into the `PACKAGES` list in the instance configuration file (`<bscw-runtime-path>/conf/config.py`). If you defined an own package (e.g. to adopt the default role configuration) be aware the package might become disabled after an upgrade. If in doubt please ask `support@orbiteam.de` for advice how to upgrade your customizations. Due to the new BSCW 5 layout:

- Please update your Apache HTTP server configuration. Change the `VirtualHost` container definition according to the “site” configuration template `<bscw-runtime-path>/conf/apache{2,24}/site.conf`, see section 3.3.1 (Unix) resp. section 4.4.2 (Windows).
- The former `<bscw-runtime-path>/apache.conf` file was renamed to `<bscw-runtime-path>/conf/apache{2,24}/bscw.conf`
- Existing entries for the *cron daemon* (Unix) resp. *Task Scheduler* (Windows) have to be adopted to the new location of the `bsadmin` command line script. Most likely you have to exchange `<bscw-runtime-path>/bsadmin` by `<bscw-runtime-path>/`**`bin`**`/bsadmin`. If you configured “folder mail delivery” on Unix the path to the local mail delivery agent in `/etc/aliases` or `.forward` has to be adopted, too, e.g. replace `<bscw-runtime-path>/cgi/bscw.cgi` by `<bscw-runtime-path>/var/www/bscw.cgi`

Important: During the upgrade process to BSCW 5.0 database conversion(s) are necessary. A single database conversion requires beside the conversion process two garbage collection runs; so estimate a downtime of 3 - 4 times the duration of a single garbage collection run. Especially big BSCW database servers with more than 10.000 users should consider this.

For new BSCW instances the default authentication method has been changed to cookie authentication (since BSCW 4.5). It is recommended to manually change the authentication method for existing BSCW instances to cookie authentication within the instance configuration file (`<bscw-runtime-path>/conf/config.py`) using the `COOKIE_AUTHENTICATION` directive (see page 59 for details). After altering the authentication method `bin/bsadmin conf_apache -n` and `bin/bsadmin index_page` must be run for (re-)configuration of the Apache HTTP server and the index page.

The “mobile” package requires cookie authentication as authentication method.

If you are using the BSCW "lucene" indexer (package `PyLucIndex`), an upgrade of "pylucene" to at least version 3.0.1 is required before running the BSCW upgrade procedure. Remind to rebuild the “lucene” index after the upgrade.

If the “ldap” package is enabled the old configuration from `config_ldap.py` must be inserted manually in the new configuration file `<bscw-runtime-path>/conf/ldap/config_ldap.py` after upgrading.

All "moderated" public workspaces are reset to "non-moderated". To restore the previous "moderated" state run `bin/bsadmin dbcheck repair -f m`

You may install the Python *lxml* package to support a version difference view for HTML documents under version control.

Note: The XML-RPC and JSON-RPC API has been extended to require additional authentication information per request if the user-agent is not whitelisted. This prevents potentially injected malicious javascript code in web browsers to utilize the BSCW API. To whitelist your RPC-client user-agent you may add an entry to `<bscw-runtime-path>/conf/config_clientmap.py` for `trusted_json_rpc_client` resp. `trusted_xml_rpc_client`. **Important**: You may **never** add an entry for any available web browser!

Python 2.5 support has been ended.

When **upgrading from BSCW 4.4.6 or lower**

The converter tool configuration file (`<bscw-runtime-path>/conf/config_convert.py`) is automatically generated by using `bin/bsadmin update_defaults -s`. This script will search the local system for archiver, encoder or converter commands (see section 5.6).

The "Flow" package has been replaced by the "Tasks" package and all "Flow" objects will be converted to new project/phase objects. During the conversion the "Process" folders role mapping to restrict inherited roles from the surrounding "Project" folder was reset. Thus it might be possible for **other** members of the project to **change** data in the "Process" folders after the conversion.

When upgrading on a Linux-based OS you must make sure that a working compiler (GCC/CC) is installed (Due to limitations of set-group-id execution for scripts on Linux the compilation of the CGI binary wrapper became mandatory).

Whenever the `SERVER_ROOT` is changed in the instance configuration file (`<bscw-runtime-path>/conf/config.py`) you must call "`bin/bsadmin update_helper`" in order to update the jnlp deployment files with the correct code base URL. Otherwise users may not be able to launch or install the BSCW Desktop application anymore.

BSCW Windows instances require at least Python for Windows Extensions version 2.14. Please upgrade older `pywin32` versions before running the BSCW installer.

Python 2.4 support has been ended in BSCW 4.4.6.

When **upgrading from BSCW 4.4.5 or lower**

Due to a (fixed) bug in the file upload process obsolete files may be still in the `data/Files` area. To remove this superfluous files, please perform the following command (on the server console) after having upgraded:

```
$ cd <bscw-runtime-path>
$ bin/bsadmin fsck -r
```

When **upgrading from BSCW 4.4.4 or lower**

The `POST_AUTHENTICATION` directive in the instance configuration file (`<bscw-runtime-path>/conf/config.py`) was renamed in `POST_AUTH`, which is now enabled by default.

When **upgrading from BSCW 4.3.4 or lower**

Administrator users explicitly need to log in a second time with their password at `[Options > Admin]` to gain BSCW administrator rights. Without this additional administrator authentication no administrative rights are applied to their account. After successful login to the `Admin` page press `[OK]` to keep the administrator rights for your current session or `[Cancel]` to drop the administrator rights again. The administrator status is indicated by a "Admin" label at top of the BSCW user interface.

The syntax of the meta data configuration `<bscw-runtime-path>/conf/config_metadata.py` has been changed. While unmodified meta data definitions are automatically converted to the new syntax, custom meta data definitions will be disabled and need to be converted manually.

The syntax of the action configuration `<bscw-runtime-path>/conf/config_actions.py` has been changed. In particular the syntax of the `Action()` class was altered. If the `Action(...)` definitions of your BSCW instance were changed, these changes must be adopted manually to the new format.

Users can now in addition to their user name log in with one of their allocated email addresses and their password. The “ldap” package has been adopted to support automatic registration for email addresses.

Python 2.3 support has been ended in BSCW 4.3.4

When **upgrading from BSCW 4.3.1 or lower**

BSCW 4.3.2 provides a new module for maintaining BSCW database object tables in an external Berkeley DB (`DBMOD_TAB = 'bsddb4'`). If you used `DBMOD_TAB = 'bsddb3'` in versions before BSCW 4.3.2 upgrade to this new module (by setting `DBMOD_TAB = 'bsddb4'` in the main configuration file `<bscw-runtime-path>/conf/config.py`). This configuration can also be used for upgrading from earlier BSCW releases.

When **upgrading from BSCW 4.2.3 or lower**

The `SERV_UNO_ROOT` directive has been deleted. BSCW services like the user notification service (*UNO*) or the alarm service (*ALARM*) expect now an additional (virtual) HTTP service running on `localhost:<HTTP_LOCAL_PORT>` (default: `HTTP_LOCAL_PORT = 80`).

Note: If you are running several BSCW instances in different virtual hosts you must configure for each BSCW instance a different `HTTP_LOCAL_PORT` number and you must extend the `VirtualHost` directives by these local IP addresses/port pairs.

The `SERVER_ADMINS_IP` directive no longer restricts the user notification service (UNO). You should remove entries from `SERVER_ADMINS_IP` which were made in BSCW 4.2 for `SERV_UNO_ROOT` resp. `SERVER_ROOT`.

When **upgrading from BSCW 4.1.4 or lower**

Important: BSCW 4.2 introduces a new owner assignment. The owner of all newly created objects automatically becomes the owner of the workspace (the owner role is now inherited by the “ambient” folder). This is in opposite to the behavior of previous BSCW versions (< 4.2), where the creator of an object also was the owner of the object. This leads to the following effects:

- Users cannot lose the access path to owned objects by accidental deletion of their workspace membership.
- The quota system assigns utilized resources of all contained objects of a workspace to the owner (and not any longer to the different object creators)
  Attention: After the upgrade you should run one of the following commands to initialize all quota counters:
  1. EDU licensees may only run the command `bin/bsadmin quota fix`.
  2. PRO licensees may run alternatively the command `bin/bsadmin quota report -vL`, which commits changes to the database after each user.
- The actions “cut” and “delete” change the owner of an object: owner becomes the user who cut/deleted the object (the object inherits the owner of the ambient folder (who is in this case the owner of the clipboard resp. the trash)).
  Attention: caused by this owner change the action “destroy” **always** destroys objects contained in the trash. The behavior of previous BSCW versions (< 4.2) to distribute “destroyed” objects first into the trash of the owner is omitted.

Important: BSCW 4.2 implements a new user notification service (UNO) which replaces the workspace activity report and the awareness service of previous BSCW versions. To not interfere with the new user notification service the workspace activity report configuration **must be**

disabled by removing the *crontab* (Unix) or the *task scheduler* (Windows) entry for `bsadmin notify -a`. Additionally remove the entry for `AWSERV` (`bs_servaw`) from the `SERVERS` list in the old instance configuration file `<bscw-runtime-path>/src/config.py` before upgrading. **After** upgrading you might add an entry for `bs_servuno` as described in the comments.

When **upgrading from BSCW 4.0.4 or lower**

The BSCW license server URI has been changed, be sure in `<bscw-runtime-path>/conf/config.py` the `BSCW_LICENSE` variable is set to:

```
BSCW_LICENSE = 'http://bscw.orbiteam.de/pub'               (upgrade BSCW 3.?)
BSCW_LICENSE = 'http://bscw.orbiteam.de/pub/bscw.cgi/'     (upgrade BSCW 4.?)
```

Important: Starting with BSCW 4.0.6 a new license mechanism was introduced. The new mechanism does not longer bind the license to the BSCW servers IP address and installation path. It is name based, which means you have to define in `<bscw-runtime-path>/conf/config.py` the `SERVER_ROOT` variable before applying for a license (cf. section 3.3.2 for Unix or 4.4.1 for Windows)

When **upgrading from BSCW 3.4.1 or lower**

Important: Since version 4.0 BSCW uses roles for access control. This new approach is incompatible with the older access control model. All special access control settings are reset to (hopefully reasonable) defaults during upgrade.

Starting with BSCW 4.0 the document tree layout of the BSCW server has been changed; if you use the Apache HTTP server, please adopt your configuration to the new layout as given in `<bscw-runtime-path>/apache2/bscw.conf` (cf. section 3.3.1 for Unix or 4.4.3 or 4.4.2 for Windows).

When **upgrading from BSCW 3.2 or 3.3**

Important: During upgrade from BSCW 3.2 or 3.3 your current BSCW license becomes invalid and a new evaluation license will be installed. It will be valid for 90 days and 200 users. This might be a problem, if you have already more than 199 registered BSCW users, because new users cannot (be) register(ed) any more. We recommend upgrading your license to the new release as soon as possible. If your old license includes support and upgrading, you will get the new license at no cost (see BSCW license in section 9).

Note: New packages are not automatically enabled after upgrading. You have to add the package names to the `PACKAGES` list in the server settings of the [`Options > Admin`]-page or the file `<bscw-runtime-path>/conf/config.py`. Some of the packages also need installation of extra software and configuration.

When **upgrading from BSCW 2.2 or lower**

Execute the following commands in your existing `BSCW2` instance directory `<bscw-runtime-path>` before installing the new version:

```
$ cd <bscw-runtime-path>
$ start_servers -k
$ mkdir data
$ mv src/.htpasswd data/htpasswd
$ mv src/BSCW_Store data/Store
$ mv src/BSCW_Files data/Files
$ echo > src/config.py
```

Then do the BSCW upgrade and reconfiguration of your HTTP server as described in the subsequent section 3.2 for Unix or section 4 for Windows.

Note: You **may not** replace the upgraded BSCW server instance configuration file (`<bscw-runtime-path>/conf/config.py`) by a `config.py` file of a previous BSCW version! Instead the upgraded BSCW server instance configuration file must be edited manually.

Note: Since the Apache HTTP server configuration `<bscw-runtime-path>/apache2/bscw.conf` is automatically generated all manual changes will be lost after an upgrade.

2.3.1 Upgrading on Unix

The installation program of the BSCW software **must** be run as superuser (root).

```
# tar xf bscw-5.0.9-3????-py26.tar.gz
# cd bscw-5.0.9-3????-py26
# ./install.sh
```

The installation procedure looks for the BSCW system user "`bscw`" (resp. requests the user name of your BSCW user account) and locates all BSCW instances.

If do not want to run the `install.sh` script as superuser or you encounter further problems you may install BSCW completely manual as follows:

- login as "bscw" user
```
# su - bscw
$ id bscw
uid=1234(bscw) gid=1234(bscw) groups=1234(bscw)
```
- create a `$HOME/lib` directory in the "bscw" users' home directory
```
$ cd $HOME
$ mkdir lib
```
- download the BSCW distribution into a temporary directory, extract the archive and extract the BSCW distribution tar file into `$HOME/lib`, e.g.
```
$ cd /tmp
$ tar xf bscw-5.0.9-3????-py26.tar.gz
$ cd $HOME/lib
$ tar xf /tmp/bscw-5.0.9-3????-py26/bscw-5.0.9-3????-py26.tar
```
- run the installation procedure `setup.py <bscw-runtime-path>` and follow the instructions
```
$ cd $HOME/lib/bscw-5.0.9-3????-py26
$ python2.6 ./bin/setup.py <bscw-runtime-path>
```

In particular the installation procedure performs the following steps to upgrade a BSCW instance

```
# ./install.sh

Enter BSCW system user name: [bscw]

Enter BSCW base directory: [/opt/bscw]

Extracting BSCW 5.0.9 distribution in /opt/bscw/lib

Choose one of the following options:
( 0) update BSCW 4.5.9 [/opt/bscw/server]
( 1) update BSCW 5.0.7 [/opt/bscw/srv/bscw.domain.org]
( 2) update other BSCW instance
( 3) create new BSCW instance
Enter a number (0-2): 1

target '/opt/bscw/srv/bscw.domain.org' exists - checking...
conf/config.py : updated...
conf/config_actions.py : created...
...
bsadmin update_defaults
bsadmin manage_servers -u
2012-11-26 15:51:20 bsadmin chkconfig
cc -o var/run/run_bscw var/run/run_bscw.c
2012-11-26 15:51:21 Actual license: OK
2012-11-26 15:51:21 bsadmin start
2012-11-26 15:51:25 Database version >= 2.1
2012-11-26 15:51:25 bsadmin bscw.adm.bs_convert30 -t
2012-11-26 15:51:25 Database version >= 3.0
2012-11-26 15:51:25 bsadmin bscw.adm.bs_convert31 -t
2012-11-26 15:51:25 Database version >= 3.1
2012-11-26 15:51:25 bsadmin bscw.adm.bs_convert33 -t
2012-11-26 15:51:25 Database version >= 3.3
2012-11-26 15:51:25 bsadmin bscw.adm.bs_convert40 -t
2012-11-26 15:51:25 Database version >= 4.0
2012-11-26 15:51:25 bsadmin bscw.adm.bs_convert45 -t
2012-11-26 15:51:25 Database version >= 4.5
2012-11-26 15:51:25 bsadmin bscw.adm.bs_convert50 -t
2012-11-26 15:51:25 Converting to Version 5.0 ...
...
2012-11-26 15:51:25 bsadmin start
2012-11-26 15:51:25 bsadmin bscw.adm.bs_fix_anonymous
2012-11-26 15:51:25 VERSION:  BSCW 5.0.9
                    Released: 20121126-1250-3????
bsadmin convert
Configure 'gzip' compression ...
Configure 'static' resources 'var/www/20121126-1250-3????'...
 (Long time future expire dates)
Configure secure prefix '/bscw/' ...
 (HTTP_AUTHORISATION passed to BSCW)
 (Cookie authentication enabled)
Configure public prefix '/pub/'...
 (No authentication)

Creating Apache HTTP server configuration files in
/opt/bscw/srv/bscw.domain.org/conf/apache{2,24}
  mod.conf ... module configuration file
 site.conf ... virtual host site configuration file
 bscw.conf ... BSCW configuration file
bsadmin conf_apache
bsadmin index_page

BSCW server up and running in '/opt/bscw/srv/bscw.domain.org'

BSCW instance updated: '/opt/bscw/srv/bscw.domain.org'
you may need to restart your web-server

Installation succeeded. For next steps please check
/opt/bscw/lib/bscw-5.0.9-3????-py26/README.txt

Since newer Linux environments do not execute forked processes
set-group-id it is advisable to recursively change the owner the
./var/data/Files directory to the web server user.
Fix file owner/modes for Apache HTTP daemon user? [Y/n]
```

By default, the installation procedure looks in the home directory of the given BSCW user (`$HOME/` and `$HOME/srv/`) to locate a BSCW instance. If you installed your BSCW instance in a non-standard location, the BSCW installation program may not be able to locate the BSCW instance directory. In this case you have two options to upgrade from a previous version to BSCW 5.0:

1. Provide the path to your BSCW instance to the BSCW installer
2. Adopt the new BSCW layout and move your BSCW instance (**recommended**)

How to proceed for each option:

1. *Provide the path to your BSCW instance to the BSCW installer*
   If you want to preserve the old non-standard location for your BSCW instance, it is possible to specify the path to your BSCW instance by choosing the option "`update other BSCW instance`":

```
# ./install.sh

Enter BSCW system user name: [bscw]

Enter BSCW base directory: [/opt/bscw]

Extracting BSCW 5.0.9 distribution in '/opt/bscw/lib'

Choose one of the following options:
 ( 0) update other BSCW instance
 ( 1) create new BSCW instance
Enter a number (0-1): 0

Enter path to BSCW instance: /usr/local/bscw/server
target '/usr/local/bscw/server' exists - checking...
…
```

Alternatively it is possible to specify the path to your BSCW instance as argument of the BSCW installer program:

```
# ./install.sh /usr/local/bscw/server
target '/usr/local/bscw/server' exists - checking...
...
```

This will upgrade your BSCW instance to BSCW 5.0.9 "in-place" and keep the BSCW instance in the old directory.

2. *Adopt the new BSCW layout and move your BSCW instance (**recommended**)*

It is recommended to move the old BSCW instance first to the new standard location `./srv/<hostname>` in the BSCW users' home directory (e.g. `/opt/bscw/srv/<hostname>`). First stop the BSCW server and then move it:

```
# cd /usr/local/bscw
# ./server/start_servers -k

# BSCW_HOME=`su - bscw -c 'echo $HOME'`    # e.g. BSCW_HOME=/opt/bscw

# mkdir -p                        $BSCW_HOME/srv/bscw.domain.org
# chown bscw:bscw                 $BSCW_HOME/srv/bscw.domain.org
# rsync -vaH –del ./server/*      $BSCW_HOME/srv/bscw.domain.org
...
```

Next run the BSCW installer (as root) - with no argument it should find the instance and offer to upgrade it:

```
# ./install.sh
...
Choose one of the following options:
 ( 0) update BSCW 4.5.9 [/opt/bscw/srv/bscw.domain.org]
 ( 1) update other BSCW instance
 ( 2) create new BSCW instance
Enter a number (0-2): 0
...
```

The BSCW installer will update your BSCW instance to BSCW 5.0.9 You finally need to adjust the HTTP server configuration – see configuration section above.

2.3.2 Upgrading on Windows

Before upgrading a BSCW instance ensure to install the "Python for Windows Extensions" (pywin32) at least with Build 214. To upgrade an existing BSCW instance on Windows start the BSCW setup procedure by double-clicking

```
bscw-5.0.9-3????-py26.exe
```

This will (re-)install the BSCW 5.0.9 distribution files in the given location. Next start the BSCW instance setup program by keeping the option *"Install a server instance now"* selected and pressing `[Finish]`. To perform an upgrade, select the BSCW instance to be updated, e.g.

```
        [upgrade BSCW 4.5.9 [c:\bscw\server]]
        install new BSCW instance
```

and click `[Continue]`. A console window is opened and the selected BSCW instance runtime directory is updated (for a detailed description see section 4.2).

Note: During the upgrade procedure the old BSCW service is deleted and a new BSCW service (with new name) is created preserving the old values for

- start type (automatic/manual/...)
- dependencies with other services
- user name under which the old service was run

if the old user name is not the *local system account* (default) the BSCW setup procedure will explicitly ask for the service users' password during the installation.

3 Installation procedure for Unix

These are the installation instructions for BSCW 5.0 on Unix machines. If you are upgrading an existing BSCW server instance please read first the section *2.3 Upgrading to BSCW 5.0.9*.

3.1 System requirements

For approximately 200 users the BSCW server requires the following server hardware:

- Intel Core/Xeon or AMD Phenom/Opteron (>2,5 GHz) 64-bit server system (or comparable systems of other manufacturers).
- 4 GB RAM
- at least 100 GB hard disk space (the BSCW installation requires approx. 75 MB disk space)

Additionally the following software is required:

- Apache HTTP Server 2.2 (2.4)
- Python 2.6 (2.7)
- (optional) extensions for Python

  lxml
  - adds support for a version difference view for versioned HTML documents

  PyLucene
  - required for full text indexing support package *PyLucIndex*

  Python-LDAP
  - required for LDAP/Active Directory bindings (package *ldap*)

Before installing BSCW first install the desired optional Python extension packages (see above).To install BSCW you need a Python interpreter (version 2.7 or 2.6 for your Unix platform.

We recommend to use the Apache HTTP server version 2.2 or 2.4 (`http://httpd.apache.org/`), but any server supporting CGI 1.1 should work. Python 2.6 or 2.7 can be downloaded at `http://www.python.org/`.

Notes:

- BSCW needs Pythons module "`crypt`". This module is not always installed by default. In this case you have to enable the crypt module in Modules/Setup and rebuild Python.
- On systems which do not allow execution of *set-group-id scripts*, e.g. Linux, a C compiler (gcc) with installed system (kernel) C headers is required to compile a binary wrapper.

In order to send registration and report emails BSCW finally needs access (via SMTP) to a mail server (Unix or Windows based).

3.2 Installation

Before installing BSCW ensure the Web server, Python and the desired optional Python extension packages are installed. The BSCW server software distribution is available as *tar* archive

```
bscw-5.0.?-3????-py26.tar.gz
bscw-5.0.?-3????-py27.tar.gz
```

The name of the download file contains BSCW and Python version numbers – e.g. `bscw-5.0.9-3????-py26.tar.gz` contains BSCW version 5.0.9 for Python 2.6. Please make sure to install the latest version of BSCW and always provide your version number when contacting support staff.

There may be additional patch releases available **after** the latest release – check the BSCW product home page http://www.bscw.de/ for latest updates that have been released for download.

The BSCW directory **should not** be accessible via the `DocumentRoot` or any other alias directives of your HTTP server. The path to the BSCW directory needs only "search permission" for the user/group ID that the HTTP server uses.

The BSCW server CGI scripts are executed (set-group-id) with the group ID `bscw`, which is the primary group ID of the BSCW system user. Hence access rights for the group ID `bscw` will be inherited during execution of all BSCW CGI scripts. To ensure an error free operation of the BSCW server:

- the set-group-id bit of the BSCW CGI scripts has to be set (which is done automatically done by the BSCW setup procedure
- the BSCW directory `<bscw-path>` (and all files and directories below) should belong the group ID `bscw`
- the file system of the BSCW directory `<bscw-path>` **must not** be mounted with the `nosuid` option

If the set-group-id execution of the BSCW CGI script fails you will get an "`Error: Wrong group id`" while BSCW operation. To fix this problem see the "note" in the last paragraph of section 3.3.3

Note: When installing on a Linux-based OS you must make sure a working compiler (GCC/CC) is installed (due to limitations of set-group-id execution for scripts on Linux, the compilation of the CGI binary wrapper became mandatory).

The BSCW installation layout and procedure was completely revised. Generally the following file layout is proposed for BSCW version 5 instances:

```
/opt/bscw/                             # BSCW user home directory
                                       # (as defined in /etc/passwd!)

/opt/bscw/.bscw/                       # BSCW instance(s) information
/opt/bscw/.bscw/bscw.conf
/opt/bscw/.bscw/bscw_conf.py

/opt/bscw/lib/                         # BSCW distribution libraries
/opt/bscw/lib/bscw-5.0.9-3????-py26/   # BSCW distribution 5.0.9
/opt/bscw/lib/bscw-5.0.9-3????-py26/bin
/opt/bscw/lib/bscw-5.0.9-3????-py26/bscw  # BSCW executable code
/opt/bscw/lib/bscw-5.0.9-3????-py26/doc   # BSCW documentation
/opt/bscw/lib/bscw-5.0.9-3????-py26/etc
/opt/bscw/lib/bscw-5.0.9-3????-py26/lib   # BSCW third party modules

/opt/bscw/srv/                         # BSCW instances
/opt/bscw/srv/<hostname>/              # BSCW instance runtime
/opt/bscw/srv/<hostname>/bin/          # BSCW instance executables
/opt/bscw/srv/<hostname>/bin/bsadmin
/opt/bscw/srv/<hostname>/conf/         # BSCW instance configuration
/opt/bscw/srv/<hostname>/conf/config.py
/opt/bscw/srv/<hostname>/etc/          # BSCW configuration hints
/opt/bscw/srv/<hostname>/libexec/      # BSCW instance runtime programs
/opt/bscw/srv/<hostname>/var/
```

```
/opt/bscw/srv/<hostname>/var/cache/      # BSCW instance template cache
/opt/bscw/srv/<hostname>/var/data/       # BSCW instance data
/opt/bscw/srv/<hostname>/var/log/        # BSCW instance log files
/opt/bscw/srv/<hostname>/var/run/        # BSCW instance runtime state
/opt/bscw/srv/<hostname>/var/www/        # BSCW instance web resources
```

The BSCW 5 layout allows to install multiple BSCW instances in the runtime directory (`/opt/bscw/srv`), which all share the same BSCW program code located in the library directory (`/opt/bscw/lib`).

As a prerequisite a suitable Python interpreter version and the Apache HTTP server must be available on the system before installing BSCW. For best performance the BSCW libraries and instances should be located on a file system local to the host where your HTTP server runs.

The installation program of the BSCW software **must** be run as superuser (root). The installation procedure will look for the BSCW system user `bscw` and uses the home directory of this user as installation base directory for BSCW (which might alter from `/opt/bscw`). If no BSCW user is found a new BSCW system user `bscw` with an own group `bscw` and a home directory `/opt/bscw` is proposed and then created.

Notes:

- `/opt/bscw` is the proposed location for the BSCW users home directory (resp. the BSCW installation base directory). Generally the installation procedure uses the BSCW users' home directory (as defined in `/etc/passwd`) as default installation base directory.
- If you want to install BSCW in another location different from the home directory of the BSCW user you may want to specify an alternate **base** directory. The base directory of a BSCW installation defines the directory where the installation program will create the `./lib` directory containing the BSCW distribution and the `./srv` directory where to create BSCW runtime instances. Usually the base directory is equal to the BSCW users' home directory and does not need to be changed.
- During the installation procedure you may specify an alternate BSCW system user name or home directory.

After creating or locating the BSCW system user the installation procedure will extract the BSCW distribution archive in the library directory (usually `/opt/bscw/lib`) and the BSCW setup procedure is called and run as BSCW system user `bscw`.

The BSCW setup procedure will allow to update existing BSCW instances or to create new BSCW instances. All required BSCW instance parameters are identified via command line dialogs.

Finally the installation procedure tries to identify the user of the Apache HTTP server and changes the ownership of the upload directory for raw files to the Apache user.

To start the installation extract the BSCW distribution archive and run the `install.sh` script as superuser:

```
# id
uid=0(root) gid=0(root) groups=0(root)
# tar xf bscw-5.0.9-3????-py26.tar.gz
# cd bscw-5.0.9-3????-py26
# ./install.sh
```

If do not want to run the `install.sh` script as superuser or you encounter further problems you may install BSCW completely manual as follows:

- login as "bscw" user

```
# su - bscw
$ id bscw
uid=1234(bscw) gid=1234(bscw) groups=1234(bscw)
```

- create a `$HOME/lib` directory in the "bscw" users' home directory

```
$ cd $HOME
$ mkdir lib
```

- download the BSCW distribution into a temporary directory, extract the archive and extract the BSCW distribution tar file into `$HOME/lib`, e.g.

```
$ cd /tmp
$ tar xf bscw-5.0.9-3????-py26.tar.gz
$ cd $HOME/lib
$ tar xf /tmp/bscw-5.0.9-3????-py26/bscw-5.0.9-3????-py26.tar
```

- run the installation procedure `setup.py <bscw-runtime-path>` and follow the instructions

```
$ cd $HOME/lib/bscw-5.0.9-3????-py26
$ python2.6 ./bin/setup.py <bscw-runtime-path>
```

In particular the installation procedure performs the following steps to create a new BSCW instance:

```
# ./install.sh

Enter BSCW system user name: [bscw]

Enter BSCW user home directory: [/opt/bscw]

Enter BSCW base directory: [/opt/bscw]

Extracting BSCW 5.0.9 distribution in /opt/bscw/lib

Choose one of the following options:
 ( 0) update other BSCW instance
 ( 1) create new BSCW instance
Enter a number (0-1): 1

Please enter the BSCW server root
(use a fully qualified domain name - an IP address is not allowed).
The server root specifies the visible URL for this instance, e.g.
http://host.domain.org or https://host.domain.org
(may be left empty):

 BSCW server root: https://bscw.domain.org

Please enter the name of your BSCW instance directory
(if left empty in directory
 /opt/bscw/srv
the default [bscw.domain.org] is created):

 BSCW instance name: [bscw.domain.org]
```

```
target '/opt/bscw/srv/bscw.domain.org' does not exist - creating...

Please enter the host name (FQDN) or the IP address
of your mail host (MTA) to relay BSCW emails
(may be left empty):

 Mail host name or IP address: mail.domain.org

Please enter email address and login name of the BSCW administrator:

 Email address: bscwadmin@domain.org
 BSCW login name: bscwadmin
 Enter Password:
 Re-type password:

Please enter the BSCW server Realm - used in Authentication dialog
and shown on the welcome page of the server.
(may be left empty and defaults to 'BSCW Shared Workspace Server')
Note: If you are running different BSCW servers on one host
      then you must use a different realm for each server.

 Realm:

Please enter the BSCW public URI prefix as used for public access URL, e.g.
http://my.bscw.de/pub/bscw.cgi
(may be left empty and defaults to 'pub')
Note: If you are running different BSCW servers on one host without using
virtual hosts then you must use a different URI prefix for each server.

 BSCW public prefix:

Please enter the BSCW secure URI prefix as used for secure access URL, e.g.
http://my.bscw.de/bscw/bscw.cgi (requires authentication)
(may be left empty and defaults to 'bscw')
Note: If you are running different BSCW servers on one host without using
virtual hosts then you must use a different URI prefix for each server.

 BSCW secure prefix:

Initial configuration:
SERVER_ROOT = 'http://bscw.domain.org'
SMTP_HOST = 'mail.domain.org'
SERVER_ADMIN = 'bscwadmin@domain.org'
SERVER_ADMINS = [ 'bscwadmin' ]

Are these settings correct (yes/no)? yes
conf/config.py : updated...
```

```
conf/config_actions.py : created...
[...]
bsadmin update_defaults
bsadmin manage_servers -u
2012-11-26 15:10:36 bsadmin chkconfig
cc -o var/run/run_bscw var/run/run_bscw.c
2012-11-26 15:10:37 Actual license: OK (none)
2012-11-26 15:10:37 bsadmin start
2012-11-26 15:10:41 bsadmin garbage -license
2012-11-26 15:10:41 GC actual license: OK (none).
                    is invalid for BSCW 5.0
                    Try installing Evaluation licence
                    Your server: org.domain.bscw:80H.bscw
                    Evaluation licence expires:   20130224
                    Evaluation licence max users: 200
[...]
bsadmin convert
bsadmin manage_servers -u
Configure 'gzip' compression ...
Configure 'static' resources 'var/www/20121126-1306-3????'...
 (Long time future expire dates)
Configure secure prefix '/bscw/' ...
 (HTTP_AUTHORISATION passed to BSCW)
 (Cookie authentication enabled)
Configure public prefix '/pub/'...
 (No authentication)

Creating Apache HTTP server configuration files in
/opt/bscw/srv/bscw.domain.org/conf/apache{2,24}
   mod.conf ... module configuration file
  site.conf ... virtual host site configuration file
  bscw.conf ... BSCW configuration file
bsadmin conf_apache
bsadmin index_page
register admin user
user bscwadmin registered, address:
    bscwadmin@domain.org: (is_owned_by_user)

BSCW server up and running in '/opt/bscw/srv/bscw.domain.org'

BSCW instance created: '/opt/bscw/srv/bscw.domain.org'
Make sure to include the BSCW Apache HTTP server configuration
(see above) in your local Apache HTTP configuration
you may need to restart your web-server

Installation succeeded. For next steps please check
/opt/bscw/lib/bscw-5.0.9-3????-py26/README.txt
```

```
Since newer Linux environments do not execute forked processes
set-group-id it is advisable to recursively change the owner the
./var/data/Files directory to the web server user.
Fix file owner/modes for Apache HTTP daemon user? [Y/n]
```

Note: If the BSCW server does not start up properly, see the file `/tmp/bscw-setup.log` or `<bscw-runtime-path>/var/log/bscw.log` in the instance runtime directory for details and error messages. The frequently asked questions (FAQ) list (`http://www.bscw.de/english/faq.html`) might also be helpful.

3.3 Configuration

The configuration includes the configuration of your Web server and the configuration of the BSCW server.

3.3.1 Apache HTTP server configuration

BSCW requires in addition to a (virtual) web service for user access, a second (virtual) web server running on `localhost` (`127.0.0.1`). This second (virtual) web server enables BSCW services (e.g. the user notification (*UNO*) service or the alarm (*ALARM*) service) to access the BSCW database server via HTTP using the following URL:

```
http://localhost/pub/bscw.cgi
```

Note: The port, the script alias path and the script name may be changed by altering the configuration directives `HTTP_LOCAL_PORT`, `SCRIPTS` and `CREATE_SCRIPTS` in the instance configuration file (`<bscw-runtime-path>/conf/config.py`)

Note: the localhost port to the HTTP server defined in `HTTP_LOCAL_PORT` **must** support HTTP; HTTPS is **not** supported!

The BSCW setup process automatically generates the following Apache HTTP server (version 2.2 and 2.4) configuration files

```
<bscw-runtime-path>/conf/apache{2,24}/mod.conf
<bscw-runtime-path>/conf/apache{2,24}/site.conf
<bscw-runtime-path>/conf/apache{2,24}/bscw.conf
```

which contain all necessary configuration instructions.

The `mod.conf` file ensures the loading of additional modules required by BSCW and must be included in the main Apache HTTP server configuration file. Instead including this file you could enable the loading of the required modules

```
cgid_module (or cgi_module)
expires_module
deflate_module
headers_module
rewrite_module
```

using your platform specific Apache layout.

The `site.conf` file contains several virtual host containers which can be used for Apache layouts which support site configuration files (e.g. Debian Linux `/etc/apache2/sites-available`). Depending on your `SERVER_ROOT` definition in the instance configuration file (`<bscw-runtime-`

path>/conf/config.py) the site.conf file defines the following virtual hosts:

1. if a HTTP server root is defined (e.g. the SERVER_ROOT directive starts with http://...) the site.conf file defines two virtual host containers: a first virtual host container for localhost:80 required by internal BSCW services and a second virtual host container for the server root host name <hostname>:80 for requests.
2. if a HTTPS server root is defined (e.g. the SERVER_ROOT directive starts with https://...) the site.conf file defines three virtual host containers: a first virtual host container for localhost:80 required by internal BSCW services, a second virtual host container for the server root host name <hostname>:80 which redirects all requests to the third virtual host container <hostname>:443 for SSL requests.

Both files include the bscw.conf file with the actual BSCW instance configuration. If you intend to use the site.conf file copy it to your Apache HTTP server configuration. Please note it will most likely not work out of the box, but you have to adopt it to your local Apache HTTP server configuration. Especially you will need to install certificates for your SSL enabled server and adopt the configuration in site.conf.

The bscw.conf file contains the actual BSCW instance configuration for the Apache HTTP server. It may be included in the main configuration file httpd-vhosts.conf resp. httpd-ssl.conf if you manually define virtual hosts (within the standard Apache HTTP server layout) or in the main HTTP server configuration file without defining virtual hosts:

```
Include <bscw-runtime-path>/conf/apache{2,24}/bscw.conf
```

When using virtual web server container (<VirtualHost> ... </VirtualHost>) directives, it is possible to include the <bscw-runtime-path>/conf/apache{2,24}/bscw.conf configuration file in multiple virtual web server containers. An example for a virtual web server definition in the Apache HTTP server configuration file should look as follows

```
<VirtualHost bscw.domain.org:80>
    ServerName  bscw.domain.org
    ServerAlias localhost
    ServerAdmin hostmaster@domain.org

    ErrorLog    logs/bscw_domain_org_error_log
    CustomLog   logs/bscw_domain_org_access_log common
    ScriptLog   logs/bscw_domain_org_error_log

    DocumentRoot "<bscw-path>/var/www"
    <Directory "<bscw-path>/var/www">
        Options                  ExecCGI FollowSymLinks MultiViews
        AllowOverride            None
        DirectoryIndex           index.html default.htm
        LanguagePriority         en de es fr
        ForceLanguagePriority    Fallback
        # Apache 2.2
        Order                    deny,allow
        Allow                    from all
        # Apache 2.4
        #Require                  all granted
    </Directory>

    Include     "<bscw-runtime-path>/conf/apache{2,24}/bscw.conf"
```

```
        </VirtualHost>


        <VirtualHost bscw.domain.org:80>
            ServerName  bscw.domain.org
            ServerAdmin hostmaster@domain.org

            ErrorLog    logs/bscw_domain_org_error_log
            CustomLog   logs/bscw_domain_org_access_log common
            ScriptLog   logs/bscw_domain_org_error_log

            DocumentRoot "<bscw-path>/var/www"
            <Directory "<bscw-path>/var/www">
                Options                 ExecCGI FollowSymLinks MultiViews
                AllowOverride           None
                DirectoryIndex          index.html default.htm
                LanguagePriority        en de es fr
                ForceLanguagePriority   Fallback
                # Apache 2.2
                Order                   deny,allow
                Allow                   from all
                # Apache 2.4
                #Require                 all granted
            </Directory>

            Include     "<bscw-runtime-path>/conf/apache{2,24}/bscw.conf"
        </VirtualHost>
```

To provide a SSL encrypted web site your virtual web server definition should look like (Note: additionally you will still require a HTTP web server on `localhost` as defined above).

```
        <VirtualHost bscw.domain.org:80>
            ServerName  bscw.domain.org
            ServerAdmin hostmaster@domain.org

            ErrorLog    logs/bscw_domain_org_error.log
            CustomLog   logs/bscw_domain_org_access_log common
            ScriptLog   logs/bscw_domain_org_script.log

            <IfModule alias_module>
                RedirectMatch permanent ^/(.*)$ https://bscw.domain.org/$1
            </IfModule>
        </VirtualHost>

        <VirtualHost bscw.domain.org:443>
            ServerName  bscw.domain.org
            ServerAdmin hostmaster@domain.org

            ErrorLog    logs/bscw_domain_org_error.log
            CustomLog   logs/bscw_domain_org_access_log common
            ScriptLog   logs/bscw_domain_org_script.log
            DocumentRoot "<bscw-runtime-path>/var/www"
            <Directory "<bscw-runtime-path>/var/www">
                Options                 ExecCGI FollowSymLinks MultiViews
                AllowOverride           None
                DirectoryIndex          index.html default.htm
                LanguagePriority        en de es fr
                ForceLanguagePriority   Fallback
                # Apache 2.2
                Order                   deny,allow
                Allow                   from all
                # Apache 2.4
                #Require                 all granted
            </Directory>

            SSLEngine on
            SSLVerifyDepth 5

            #SSLCACertificateFile     conf/ssl/ca-bundle.crt
            #SSLCertificateChainFile  conf/ssl/bscw_domain_org_root.crt
            SSLCertificateKeyFile    conf/ssl/bscw_domain_org.key
            SSLCertificateFile       conf/ssl/bscw_domain_org.crt

            Include     "<bscw-runtime-path>/conf/apache{2,24}/bscw.conf"
        </VirtualHost>
```

You may change the BSCW Apache HTTP server configuration file by using the `bin/bsadmin conf_apache` script. To adopt the generated Apache configuration file to your local web server settings use one of the following options:

- If no option is used `bin/bsadmin conf_apache` tries to read the old option setting from `bscw.conf` (if exists). Use option `-n` or remove `bscw.conf` if you want to avoid this.
- If option `-r` is used (requires "`rewrite`" module) the user credentials are passed that the authentication is handled by the BSCW server (this is the default case).
- If option `-a` is used, BSCW allows to let the Apache HTTP server perform authentication (**note**: certain configurations (such as cookie authentication) imply option `-r`).
- If option `-s` is used the Apache HTTP server is configured for authentication via client certificates. This option includes the `-r` option and requires a SSL enabled server.
- If option `-o` is used client certificates authentication optional. This option includes the `-r` option and requires a SSL enabled server.
- If the `-D` or `-E` options are used the Apache HTTP server is configured to compress (`gzip`) BSCW resources (`-D`) or to cache resources due to a long time future expiry date (`-E`). This options require the "`deflate`" (`-D`) or the "`expires`" (`-E`) modules (these options are enabled by default).

Notes:

- If you are running several BSCW instances in different virtual hosts you must configure for each BSCW instance a different `HTTP_LOCAL_PORT` number and you must extend the `VirtualHost` directives by these local IP addresses/port pairs.
- It might be necessary to add an extra `Listen 127.0.0.1:<HTTP_LOCAL_PORT>` directive to the

main Apache HTTP server configuration file.

Remember to always **restart** your Apache HTTP server whenever the `bsadmin conf_apache` script was run.

Please note the following relations between HTTP server directives and the BSCW server instance configuration file (`<bscw-runtime-path>/conf/config.py`) variable settings:

- the BSCW server instance `SERVER_ROOT` definition must correspond at least with one (virtual) server name (as specified in the `ServerName` directive), e.g.:

  ```
  SERVER_ROOT = 'https://bscw.domain.org'
  ⇔
  ServerName "bscw.domain.org"
  Port 443
  ```

- the BSCW server instance value for the `BSCW_REALM` variable corresponds with the setting of the HTTP servers `AuthType` and `AuthName` directives, e.g.:

  ```
  BSCW_REALM = 'BSCW Shared Workspace Server'
  ⇔

  AuthType = Basic
  AuthName = "BSCW Shared Workspace Server"
  ```

  Otherwise problems with user authentication might occur: typically, users are asked twice for their passwords during registration or when switching user id.

3.3.2 BSCW instance configuration

You might skip the next parts of the configuration if you just upgraded your old BSCW server. The old configuration should be OK.

Local configuration details of your BSCW instance are held in the configuration file at `<bscw-runtime-path>/conf/config.py` (cf. section 5.2). The minimum you need to do is to configure "`Section 1: MANDATORY server settings`" of this file:

- The "server root" - the host name (and port) part of your BSCW servers URL - is specified in the variable `SERVER_ROOT` contains the absolute URL of your BSCW server and an optional port. If no port is specified the standard ports 80 (for HTTP) or 443 (for HTTPS) are assumed:

  ```
  SERVER_ROOT = 'http://bscw.domain.org/'
  SERVER_ROOT = 'http://bscw.domain.org:123/'
  SERVER_ROOT = 'https://bscw.domain.org/'
  ```

  A fully qualified host name is required as server name "bscw.domain.org", in order to allow the BSCW server to resolve its name to an IP address (`SERVER_ROOT` may **not** contain an IP address any more!).

  Ideally you define a host name/nickname (`A/CNAME`) in your DNS zone, which points to your BSCW server host, e.g.

  ```
  server1.domain.org     A         1.2.3.4
  server2.domain.org     A         1.2.3.5
  bscw.domain.org        CNAME     server1.domain.org
  ```

  Proceeding this way a future migration of your BSCW server from `server1` to `server2` will keep the well known URL http://bscw.domain.org/ and your license will not be invalidated by the migration.

  Note: whenever the `SERVER_ROOT` is changed in the instance configuration file (`<bscw-runtime-path>/conf/config.py`) you must call `bin/bsadmin update_helper` in order to update the jnlp deployment files with the correct codebase URL. Otherwise users may not be able to launch or install the BSCW Desktop application anymore.

- SERVER_ADMIN contains the **valid** email address of the server administrator, e.g.

  ```
  SERVER_ADMIN = 'bscw@domain.org'
  ```

- SERVER_ADMINS defines a list of BSCW users that have administrator rights, e.g.

  ```
  SERVER_ADMINS = [ 'bscw-admin', 'YourName' ]
  ```

  You will most likely want to add your BSCW login name to SERVER_ADMINS to give yourself administrator rights (and maybe the login names of other BSCW users who should have administrator rights).

- SMTP_HOST contains a host name or an IP-address of a mail host, that accepts mail posting by SMTP, e.g.

  ```
  SMTP_HOST = 'mail.domain.org'
  ```

  The BSCW system can use the local mail transfer agent (MTA), such as *sendmail* to send email (e.g. registration invitations), which should be fine for most installations. However, it may be better if BSCW directly uses your *smart mailhost* via SMTP. In general we recommend to use SMTP_HOST rather than SENDMAIL.

  To do this, set the SMTP_HOST directive in <bscw-runtime-path>/conf/config.py to the IP address (or fully qualified domain name) of the machine that hosts your *smart mailhost*. **Note**: if you you are using MS Exchange as MTA, you must explicitly allow the IP address of your BSCW server host to ***relay*** email.

3.3.3 Administrator account

After your BSCW instance is running you can log in with the administrator account registered during the setup process (mind login name and password are case sensitive!) by opening the URL:

```
http://bscw.domain.org/bscw/bscw.cgi
```

Actually to gain administrator rights you have to login a second time with your password by opening

```
[Options > Admin]
```

If you open the URL

```
http://bscw.domain.org/pub/
```

you get a BSCW overview which contains links to your BSCW instance.

Note: If you get an “Error: Wrong group id” during this steps the BSCW CGI scripts are not executed with the group ID bscw. This may happen because of the following reasons:

1. The set-group-id bit of the BSCW CGI script is not set. In this case, please execute the following command in your BSCW instance directory:

   ```
   $ cd <bscw-runtime-path>
   $ ./bin/bsadmin chkconfig
   ```

2. You have installed BSCW on a file system that is mounted with the nosuid option. In this case you have to remount the filesystem without the nosuid option.
3. Your operating system does not support the set-group-id bit for scripts (eg. Linux, BSD). In this case you have to compile a binary wrapper program and to reinstall the CGI scripts. Please ensure a C-compiler (cc, gcc) is available in the path and execute the following command in your BSCW instance directory again:

   ```
   $ cd <bscw-runtime-path>
   $ ./bin/bsadmin chkconfig
   ```

3.4 Database Server, Garbage Collection and Backup

All data of the BSCW server is held in the BSCW data store and handled through the BSCW database server. The BSCW database server is managed with the `start_servers` script, which is located in the BSCW instance `<bscw-runtime-path>/bin` directory:

- to start up BSCW database server use

```
$ <bscw-runtime-path>/bin/start_servers
```

- to stop BSCW database server use

```
$ <bscw-runtime-path>/bin/start_servers -k
```

- to run the garbage collector use

```
$ <bscw-runtime-path>/bin/start_servers -gc
```

The state and errors of the BSCW database server are logged in the file `<bscw-runtime-path>/var/log/bscw.log`. We recommend that `start_servers` should be executed at system boot and `start_servers -k` at shut-down.

The BSCW distribution provides boot scripts for several POSIX systems. Choose the according directory for your system. E.g. for Debian Linux copy the files from

```
/opt/bscw/lib/bscw-5.0.9-3????-py26/etc/posix/debian


/opt/bscw/lib/bscw-5.0.9-3????-py26/etc/posix/debian/etc/logrotate.d/bscw
/opt/bscw/lib/bscw-5.0.9-3????-py26/etc/posix/debian/etc/init.d/bscw
/opt/bscw/lib/bscw-5.0.9-3????-py26/etc/posix/debian/etc/default/bscw
```

to your etc directory. Next edit the `/etc/default/bscw` file to set your BSCW user and the paths to your BSCW instances runtime directories. Finally you have to enable the `bscw` boot script for the different runlevels. Refer the boot script comments how to obtain this for your system. On Debian Linux you have to run

```
# insserv bscw
```

You will need to set up the system to **garbage collect every day**. The task of the garbage collector is to find unreferenced, e.g., obsolete objects in the data store and remove them. For performance reasons, a “delete” operation on an object may not remove the respective object physically from the store. If you do not run the garbage collector periodically, the BSCW data store will grow constantly although many of its objects are obsolete. This would waste disk space and may substantially reduce the performance of the BSCW server.

We recommend that you set up a *cron* job for running the `start_servers -gc` script, though you can do it manually. An example crontab entry for daily garbage collection at 06:05 looks like:

```
# garbage collection
5 6 * * *  <bscw-runtime-path>/bin/start_servers -gc
```

Do not stop the BSCW database server before garbage collection, the garbage collection **needs** a running server!

Additionally it is **urgently** recommended to have regular **BACKUPS** (e.g. daily) of the configuration and the data store to avoid loss of data, e.g., because of a disk crash. The recommended time for backup is just after garbage collection.

The garbage collection creates alternating a garbage collected version of the BSCW database in the files `<bscw-runtime-path>/var/data/StoreA` or `StoreB` (Note: these locations can be overridden by editing `<bscw-runtime-path>/conf/config.py`). Generally you should consider the following files or directories of your BSCW instance (relative to your <bscw-runtime-path>) for backup:

- BSCW instance configuration files located in the ./conf/ directory
- BSCW instance data files and directories such as

```
./var/data/
./var/log/
./var/www/
```

Best you backup your complete BSCW instance directory <bscw-runtime-path>. This task can be done automatically after garbage collection if there is a file start_servers.conf in the instance directory containing the following:

```
save=<safe-path>/bscw_backup.tar.gz
backup_files='<bscw-runtime-path>'      # must be absolute
backup='tar cf - $backup_files | gzip -c > $save &&
        rm `./bin/bsadmin getconfig SAVE`'
```

(The file start_servers.conf is loaded by the start_servers -gc script and $backup is evaluated after successful garbage collection.)

Using this hook the complete BSCW instance directory is saved to the archive file <safe-path>/bscw_backup.tar.gz. To restore the system make sure you stop the BSCW database server and extract the archive relative to the root (/) directory (assuming 'backup_files' above was set to the absolute path of the BSCW instance directory):

```
$ cd <bscw-runtime-path>
$ ./bin/start_servers -k
$ cd /
$ gunzip < $save | tar xpf -

$ cd <bscw-runtime-path>
$ mv `./bin/bsadmin getconfig SAVE RESTORE`
$ ./bin/start_servers
```

(Which restores the BSCW instance directory <bscw-runtime-path>, then moves the (restored) ./var/data/Backup file to the active database file ./var/data/StoreA or ./var/data/StoreB.)

Notes:

- The data/Text and data/Index directories may be skipped while backup, because the contents may be reconstructed after restoration of a backup.
- Alternatively you can use any incremental backup method.

3.5 Folder Mail Delivery

Sending email to a BSCW folder is an alternative to the usual HTML/HTTP interface where users create content, e.g., via “Add Document” or “Add Note” actions using a Web browser. To enable folder mail delivery the following configuration steps have to take part:

- the BSCW mail delivery agent (MDA) has to be configured
- the local mail transfer agent (MTA) mail has to be configured to deliver incoming mails for the BSCW server mailbox to the BSCW MDA

Note: Your MTA must support VERP (variable envelope return paths) to allow the individual addressing of single folders; BSCW folder delivery is known to work with recent versions of sendmail, Postfix or qmail).

3.5.1 BSCW mail delivery agent (MDA)

The BSCW mail delivery agent (MDA) is configured by setting the following entries in the BSCW server instance configuration file `<bscw-runtime-path>/conf/config.py`:

```
#    MDA_MTA
#       Specifies the local mail transfer agent (MTA), currently
#       supported are:
#           MDA_MTA = 'qmail'
#           MDA_MTA = 'postfix'
#           MDA_MTA = 'sendmail'
#       Setting MDA_MTA = ''  or any unknown MTA will disable the
#       BSCW mail delivery feature (this is the default).
#    MDA_MBOX
#       Local mailbox name for BSCW mda (this is normally the BSCW
#       user id name)
#    MDA_DOMAIN
#       Domain name of the BSCW MDA (which is the delivery domain of
#       the local MTA for the local BSCW MDA mailbox)
#    MDA_HDRDESCR
#       Defines which headers are shown in the description of an uploaded
#       email, e.g. MDA_HDRDESCR = ['From', 'To', 'Cc']
#    MDA_DELIMITER = None (optional)
#       allows to override the MTA default recipient delimiter
#       MDA_DELIMITER = '+' (sendmail/postfix)
#       MDA_DELIMITER = '-' (qmail)
#    MDA_EXT = True (optional)
#       appends the extension for the MIME type 'message/rfc822' (as
#       defined in config_mime.py: .eml or .mht) to the email name.
MDA_MTA = ''    # disabled (default)
MDA_MBOX = 'lab'
MDA_DOMAIN = 'bscw.de'
MDA_HDRDESCR = []
```

In the given example, the local BSCW mailbox is set to "`lab`" and the delivery domain name of the local MTA is "`bscw.de`". Hence, a folder mail address has the form "`lab+1234@bscw.de`" (for sendmail and postfix) and "`lab-1234@bscw.de`" (for qmail).

To ensure consistent mail addresses, when local BSCW mail delivery is enabled, the BSCW server should only use the local mail server, therefore it is advisable to set

```
SMTP_HOST = ''
```

3.5.2 Local Mail Transfer Agent (MTA)

To deliver mail into a BSCW folder the localhost mail transfer agent has to deliver mail messages to a "program", namely to the BSCW mail deliver agent. This is achieved by "piping" the message into the BSCW main CGI script:

```
"|<bscw-runtime-path>/var/www/bscw.cgi"
```

Sendmail

To enable the BSCW MDA to deliver mails into folder for sendmail the following

/etc/mail/sendmail.cf configuration must be ensured:

- to allow sendmail program message delivery to the BSCW MDA the sendmail "prog" mailer has to be defined in /etc/mail/sendmail.cf as follows:

  ```
  Mprog,        P=/bin/sh, F=lsDFMPoqeu9,
                S=EnvFromL/HdrFromL, R=EnvToL/HdrToL, D=$z:/,
                T=X-Unix/X-Unix/X-Unix,
                A=sh -c $u
  ```

  The F and P flags in the "prog" mailer flag list F= are required, to ensure the message contains a "From:" and "Return-Path:" header line.

  Note: you may not use "smrsh" (restricted shell for sendmail) as "prog" mailer for sendmail, since it does not permit the delivery into the BSCW MDA script. Alternatively you might link the bscw.cgi script from /etc/smrsh.
- to enable the BSCW MDA to determine a well-defined recipient of a message you have to ensure the header definition HReceived in /etc/mail/sendmail.cf contains a

  ```
  for $u; $|;
  ```

  line (which is the default setting in newer sendmail versions).
- To make multiple recipients work with sendmail add a Delivered-To: header by enter the following configuration line to /etc/mail/sendmail.cf:

  ```
  H?J?Delivered-To: $u
  ```

After editing /etc/mail/sendmail.cf your sendmail needs to be restarted before changes become effective.

After successful sendmail configuration, the program delivery to the BSCW MDA is enabled by choosing one of the following alternatives:

- enter the following line into BSCW users ID $HOME/.forward file:

  ```
  "|<bscw-runtime-path>/var/www/bscw.cgi"
  ```

or

- add an alias to the sendmail aliases database /etc/aliases file

  ```
  bscw:    "|<bscw-runtime-path>/var/www/bscw.cgi"
  ```

and run the "newaliases" program.

Finally to enable folder mail delivery in BSCW set in the BSCW server instance configuration file <bscw-runtime-path>/conf/config.py (beside the other settings described above)

```
MDA_MTA = 'sendmail'
```

To test the folder mail delivery create a folder (within BSCW) and trigger the action "Open to Mail". Choose in the form the "enabled for anybody" option. After enabling the mail upload look at the folders info page to determine the folders email address. (If in the "Details" table a "Email address" row is missing, the BSCW MDA was not properly configured, check again your BSCW MDA configuration).

To debug the mail delivery enter the following entry into the BSCW_LOGGING directive in the BSCW server instance configuration file <bscw-runtime-path>/conf/config.py:

```
BSCW_LOGGING = {
    'mda': ('DEBUG', 'mda.log'),
}
```

Send a mail message to the prepared folder address and check in /var/log/syslog (or wherever your sendmail writes its log entries) if the local sendmail program received the message and delivered it to the BSCW MDA. Typical log entries of a successful delivery look like:

```
Nov 15 15:29:17 maestral sendmail[5801]: g97G0Kp05801:
     from=<info@orbiteam.de>, size=551, class=0, nrcpts=1,
     msgid=<201205151600.g97G0DW08799@tormenta.orbiteam.de>,
     proto=ESMTP, daemon=MTA-IPv4, relay=mail [195.127.160.172]
Nov 15 15:29:17 maestral sendmail[5802]: g97G0Kp05801:
     to=|/opt/bscw/srv/lab.bscw.de/var/www/bscw.cgi,
     ctladdr=<lab+1234@bscw.de> (523/57), delay=00:00:01,
     xdelay=00:00:00, mailer=prog, pri=30015, dsn=2.0.0, stat=Sent
```

Next check the log file (default: `<bscw-runtime-path>/var/log/mda.log`). A successful delivery log entry for a sendmail MTA looks like:

```
2012-11-15 15:29:18 mda    INFO   invoked as 523/57
2012-11-15 15:29:18 mda    DEBUG
    MDA_MTA    = 'sendmail'
    MDA_MBOX   = 'lab'
    MDA_DOMAIN = 'bscw.de'
2012-11-15 15:29:18 mda    INFO   start delivery
2012-11-15 15:29:18 mda    INFO   sender addr in 'from': header.
2012-11-15 15:29:18 mda    INFO   recipient in header: <lab+1234@bscw.de>
2012-11-15 15:29:18 mda    INFO   set domain to 'bscw.de'
2012-11-15 15:29:18 mda    INFO   store document
2012-11-15 15:29:18 mda    INFO   message loaded
2012-11-15 15:29:18 mda    INFO   message stored size=2028
2012-11-15 15:29:18 mda    INFO   virus check OK
2012-11-15 15:29:18 mda    INFO   msg for Folder#118433 (access 'anybody');
2012-11-15 15:29:18 mda    INFO   msg from info <info@orbiteam.de> delivered.
```

Postfix

To enable the BSCW MDA to deliver mails into folders for the Postfix MTA add the line

```
recipient_delimiter = +
```

to the Postfix configuration file `/etc/postfix/main.cf`.

After Postfix configuration, the program delivery to the BSCW MDA is enabled by choosing one of the following alternatives:

- enter the following line into BSCW users ID `$HOME/.forward` file:

```
"|<bscw-runtime-path>/var/www/bscw.cgi"
```

or

- add an alias for the `MDA_MBOX` (e.g. `bscw`) directive to the sendmail aliases database `/etc/aliases` file

```
bscw:    "|<bscw-runtime-path>/var/www/bscw.cgi"
```

and run the “`newaliases`” program.

Finally to enable folder mail delivery in BSCW set in the BSCW server instance configuration file `<bscw-runtime-path>/conf/config.py` (beside the other settings described above)

```
MDA_MTA = 'postfix'
```

To test the folder mail delivery create a folder (within BSCW) and trigger the action “Open to Mail”.

Choose in the form the “enabled for anybody” option. After enabling the mail upload look at the folders info page to determine the folders email address. (If in the “Details” table a “Email address” row is missing, the BSCW MDA was not properly configured, check again your BSCW MDA configuration).

To debug the mail delivery enter the following entry into the `BSCW_LOGGING` directive in the BSCW server instance configuration file `<bscw-runtime-path>/conf/config.py`:

```
BSCW_LOGGING = {
    'mda': ('DEBUG', 'mda.log'),
}
```

Send a mail message to the prepared folder address and check in `/var/log/syslog` (or wherever your `postfix` writes its log entries) if the local `postfix` program received the message and delivered it to the BSCW MDA. Typical log entries of a successful delivery look like:

```
Nov 15 15:29:18 hosting-b24d7f41 postfix/smtpd[27822]: 786AD18660BA:
   client=localhost[127.0.0.1]
Nov 15 15:29:18 hosting-b24d7f41 postfix/cleanup[27823]: 786AD18660BA:
   message-id=<20120301142916.GA10103@orbigate.orbiteam.de>
Nov 15 15:29:18 hosting-b24d7f41 postfix/smtpd[27822]:
   disconnect from localhost[127.0.0.1]
Nov 15 15:29:18 hosting-b24d7f41 postfix/qmgr[2714]: 786AD18660BA:
   from=<paulsen@orbiteam.de>, size=1791, nrcpt=1 (queue active)
...
Nov 15 15:29:18 hosting-b24d7f41 postfix/local[27841]: 786AD18660BA:
   to=<lab+1234@mail.orbiteam.de>, relay=local, delay=0.38,
   delays=0.01/0.01/0/0.36, dsn=2.0.0, status=sent (delivered to command:
   /opt/bscw/srv/lab.bscw.de/var/www/bscw.cgi)
Nov 15 15:29:18 hosting-b24d7f41 postfix/qmgr[2714]: 786AD18660BA: removed
```

Next check the log file (default: `<bscw-runtime-path>/var/log/mda.log`). A successful delivery log entry for a postfix MTA looks like:

```
2012-11-15 15:29:18 mda    INFO   invoked as 523/57
2012-11-15 15:29:18 mda    DEBUG
    MDA_MTA    = 'postfix'
    MDA_MBOX   = 'lab'
    MDA_DOMAIN = 'bscw.de'
2012-11-15 15:29:18 mda    INFO   start delivery
2012-11-15 15:29:18 mda    INFO   sender addr in 'from': header.
2012-11-15 15:29:18 mda    INFO   recipient in header: <lab+1234@bscw.de>
2012-11-15 15:29:18 mda    INFO   set domain to 'bscw.de'
2012-11-15 15:29:18 mda    INFO   store document
2012-11-15 15:29:18 mda    INFO   message loaded
2012-11-15 15:29:18 mda    INFO   message stored size=2028
2012-11-15 15:29:18 mda    INFO   virus check OK
2012-11-15 15:29:18 mda    INFO   msg for Folder#118433 (access 'anybody');
2012-11-15 15:29:18 mda    INFO   msg from info <info@orbiteam.de> delivered.
```

Qmail

To assign a local BSCW user mailbox, enter the following lines into your `/var/qmail/users/assign` file

```
=<bscw-mbox>:<bscw-user>:<bscw-uid>:<bscw-gid>:<bscw-runtime-path>:::
+<bscw-mbox>-:<bscw-user>:<bscw-uid>:<bscw-gid>:<bscw-runtime-path>:-::
```

where

`<bscw-mbox>` = local BSCW server mailbox as defined in `MDA_MBOX`, e.g. `bscw`
`<bscw-user>` = user name of the BSCW server user, e.g. `bscw`
`<bscw-uid>` = user ID of the BSCW server user, e.g. `523`
`<bscw-gid>` = group ID of the BSCW server user, e.g. `57`
`<bscw-runtime-path>` = path to your BSCW instance, e.g. `/opt/bscw/srv/lab.bscw.de`

While the configuration line starting with a `=` character defines the handling of the local address bscw, the line starting with a `+` character handles all extension addresses `bscw-*` (for further details consult the qmail-users man page or the “Life with qmail” documentation). After changing the contents of the `/var/qmail/users/assign` file you have to run the `qmail-newu` command to update the assignments database.

To enable BSCW MDA program delivery for all extension addresses `bscw-*`, create the file `<bscw-runtime-path>/.qmail-default` (with the BSCW user ID and group ID as owner) and enter one single line

```
|<bscw-runtime-path>/var/www/bscw.cgi
```

Finally set in the BSCW server instance configuration `<bscw-runtime-path>/conf/config.py` (beside the other settings described above)

```
MDA_MTA = 'qmail'
```

To test the folder mail delivery create a folder (within BSCW) and trigger the action “Open to Mail”. Choose in the form the “enabled for anybody” option. After enabling the mail upload look at the folders info page to determine the folders email address. (If in the “Details” table a “Email address” row is missing, the BSCW MDA was not properly configured, check again your BSCW MDA configuration in the BSCW server instance configuration).

To debug the mail delivery enter the following entry into the `BSCW_LOGGING` directive in the BSCW server instance configuration file `<bscw-runtime-path>/conf/config.py`:

```
BSCW_LOGGING = {
    'mda': ('DEBUG', 'mda.log'),
}
```

Send a mail message to the prepared folder address and check in `/var/log/syslog` (or wherever your `qmail-send` writes its log entries). If the `qmail-send` program delivered it to the BSCW MDA. Typical (sys)log entries of a successful delivery look like:

```
Nov 15 15:29:18 maestral qmail: [ID 748625 mail.info] 1029764356.914769
    info msg 236165: bytes 653 from <info@orbiteam.de> qp 4281 uid 503
Nov 15 15:29:18 maestral qmail: [ID 748625 mail.info] 1029764356.915894
    starting delivery 22: msg 236165 to local lab-1234@bscw.de
Nov 15 15:29:18 maestral qmail: [ID 748625 mail.info] 1029764356.916318
    status: local 1/10 remote 0/20
Nov 15 15:29:18 maestral qmail: [ID 748625 mail.info] 1029764357.554749
    delivery 22: success: did_0+0+1/
```

```
Nov 15 15:29:18 maestral qmail: [ID 748625 mail.info] 1029764357.555183
     status: local 0/10 remote 0/20
Nov 15 15:29:18 maestral qmail: [ID 748625 mail.info] 1029764357.555524
     end msg 236165
```

Check the log file (default: `<bscw-runtime-path>/data/mda.log`). A successful delivery log entry for a qmail MTA looks like:

```
2012-11-15 15:29:18 mda    INFO   invoked as 523/57
2012-11-15 15:29:18 mda    DEBUG
    MDA_MTA    = 'qmail'
    MDA_MBOX   = 'lab'
    MDA_DOMAIN = 'bscw.de'
2012-11-15 15:29:18 mda    INFO   start delivery
2012-11-15 15:29:18 mda    INFO   sender addr in 'from': header.
2012-11-15 15:29:18 mda    INFO   recipient in header: <lab+1234@bscw.de>
2012-11-15 15:29:18 mda    INFO   set domain to 'bscw.de'
2012-11-15 15:29:18 mda    INFO   store document
2012-11-15 15:29:18 mda    INFO   message loaded
2012-11-15 15:29:18 mda    INFO   message stored size=2028
2012-11-15 15:29:18 mda    INFO   virus check OK
2012-11-15 15:29:18 mda    INFO   msg for Folder#118433 (access 'anybody');
2012-11-15 15:29:18 mda    INFO   msg from info <info@orbiteam.de> delivered.
```

4 Installation procedure for Windows 7, 8/Server 2012, 2008, 2003

These are the installation instructions for BSCW 5.0 on Windows 7, 8/Server 2012, 2008, 2003 machines. If you are upgrading an existing BSCW server instance please read first the section *2.3 Upgrading to BSCW 5.0.9*

4.1 System requirements

For approximately 200 users BSCW requires the following server hardware on a Windows 7, 8/Server 2012, 2008, 2003 installation:

- Intel Core/Xeon or AMD Phenom/Opteron (>2,5 GHz) 64-bit server system.
- 4 GB RAM
- at least 100 GB hard disk space (the BSCW installation requires about 75 MB disk space)
- Windows 7, 8/Server 2012, 2008, 2003 with
  - Apache HTTP Server 2.2 or 2.4

  or
  - Microsoft Internet Information Server (IIS 6, 7 or 8)

To use BSCW you will need the “Python” interpreter software and extensions:

- Python 2.6 (2.7)
- “pywin32” Build 216 (Win32 Extensions and API for Python)
- (optional) extensions for Python:

  lxml
  - adds support for a version difference view for versioned HTML documents

  PyLucene
  - required for full text indexing support package *PyLucIndex*

  Python-LDAP
  - required for LDAP/Active Directory bindings (package *ldap*)

The “Python” interpreter and the “Python for Windows Extensions” (`pywin32`) are copyrighted, but freely usable and can be downloaded from

```
http://www.python.org/
http://pywin32.sourceforge.net/
```

Notes:

- Starting with the Python for Windows Extensions `pywin32-210.win32-py2.6.exe` the installation of the additional library `mfc71.dll` may be necessary depending on your Windows version.
- BSCW requires at least Python for Windows Extensions Build 214. Please upgrade older `pywin32` versions before running the BSCW installer `bscw-5.0.9-3????-py26.exe`.
- Before installing BSCW first install the desired optional Python extension packages (see above).
- If the installer fails with an error message like

  ```
  IOError: [Errno 13] Permission denied:
        'C:\BSCW\srv\<runtime>\conf\config.py'
  ```

  please disable your virus scanner before running the BSCW installer `bscw-5.0.9-3????-py26.exe`.

- BSCW requires the use of a NTFS (local directory).

Additionally you require a CGI 1.1 compliant Web server. BSCW supports

- Apache HTTP Server 2.2 or 2.4
- Microsoft Internet Information Server (IIS)

To use the BSCW WebDAV functionality, you must use the Apache HTTP server version 2.2 or 2.4. The Apache HTTP server is copyrighted, but is freely usable and can be downloaded from the Apache HTTP server project (`http://httpd.apache.org/`).

In order to send registration and report emails BSCW finally needs access (via SMTP) to a mail server (Unix or Windows based).

4.2 Installation and Configuration

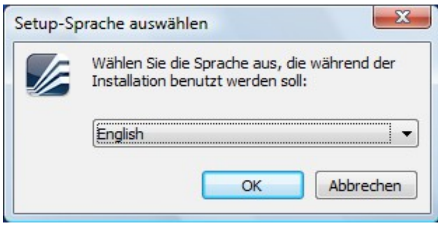
Before installing BSCW ensure the Web server, Python and the desired optional Python extension packages are installed. The name of the download installer contains BSCW and Python version numbers – e.g. `bscw-5.0.9-3????-py26.exe` contains BSCW version 5.0.9 for Python 2.6. Please make sure to install the latest version of BSCW and always provide your version number when contacting support staff.

Note: If you want to deploy BSCW with IIS the CGI support must be manually enabled *before* the BSCW installer is started, otherwise the automatic configuration of IIS may fail.

Start the BSCW setup procedure by double-clicking the installer (according to your Python version):

```
bscw-5.0.9-3????-py26.exe
bscw-5.0.9-3????-py27.exe
```

The BSCW installer first asks for the language used in the current setup procedure. Select the desired language and press `[OK]`.

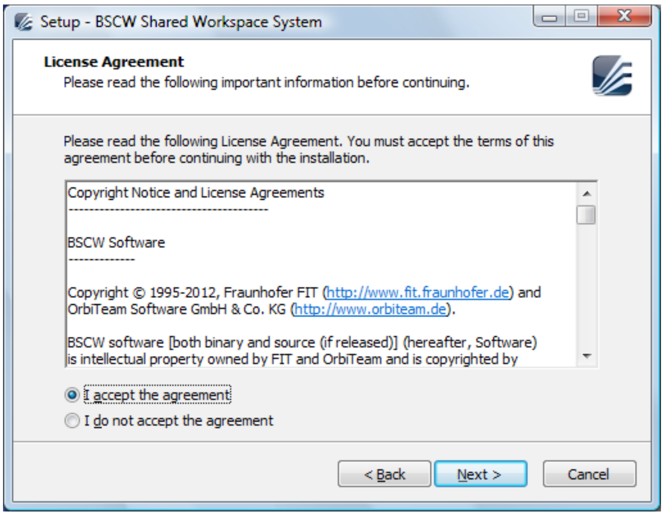


Setup-Sprache auswählen

Wählen Sie die Sprache aus, die während der Installation benutzt werden soll:

English

Then the setup program will try to install the BSCW version 5.0.9. Click `[Next]` and accept the license agreement



Setup - BSCW Shared Workspace System

License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

Copyright Notice and License Agreements

BSCW Software

Copyright © 1995-2012, Fraunhofer FIT (http://www.fit.fraunhofer.de) and OrbiTeam Software GmbH & Co. KG (http://www.orbiteam.de).

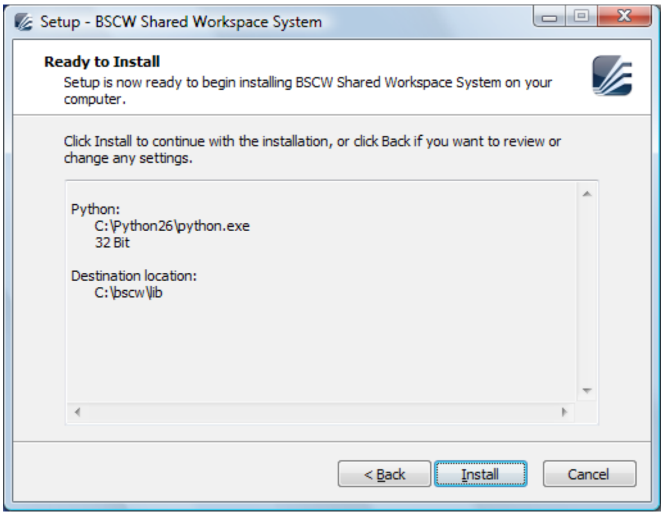
BSCW software [both binary and source (if released)] (hereafter, Software) is intellectual property owned by FIT and OrbiTeam and is copyrighted by

● I accept the agreement

○ I do not accept the agreement

and continue with `[Next]`

By default BSCW setup will install the BSCW program code in the `C:\BSCW\lib` directory. Accept this pre-selection or select a different directory and click `[Next]` to see a summary of the chosen locations:



Setup - BSCW Shared Workspace System

Ready to Install

Setup is now ready to begin installing BSCW Shared Workspace System on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

Python:
    C:\Python26\python.exe
    32 Bit

Destination location:
    C:\bscw\lib

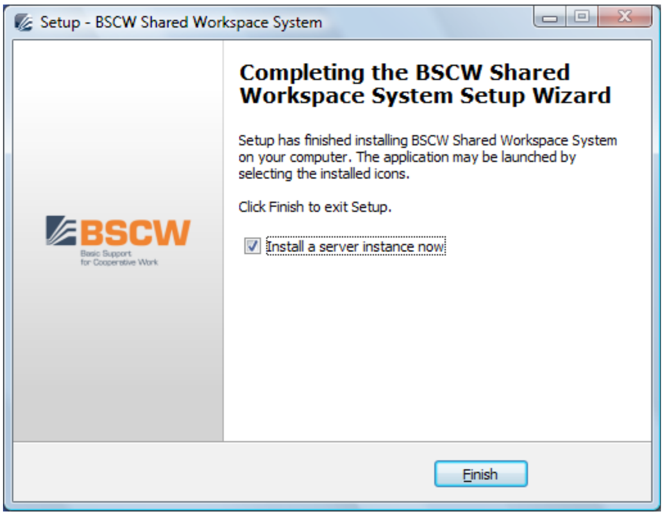
To accept this click `[Install]` which will extract the BSCW 5.0.9 distribution files in the following locations

```
C:\BSCW\lib\bscw-5.0.9-3????-py26\          # BSCW distribution 5.0.9
C:\BSCW\lib\bscw-5.0.9-3????-py26\bin
C:\BSCW\lib\bscw-5.0.9-3????-py26\bscw      # BSCW executable code
C:\BSCW\lib\bscw-5.0.9-3????-py26\doc       # BSCW documentation
C:\BSCW\lib\bscw-5.0.9-3????-py26\etc
C:\BSCW\lib\bscw-5.0.9-3????-py26\lib       # BSCW third party modules
```

The BSCW 5 layout allows to install multiple BSCW instances in the runtime directory (`C:\BSCW\srv`), which all share the same BSCW program code located in the library directory (`C:\BSCW\lib`).

Important: BSCW distribution *must* reside in the same partition as all BSCW instances. For instance it is *not* possible to install the BSCW distribution on drive `C:\` and a BSCW instance on another drive (e.g. `D:\`).

After installing the BSCW distribution files the setup program will run the "instance setup" to complete the installation. To start the BSCW instance setup program keep the option *"Install a server instance now"* selected and press `[Finish]`


Setup - BSCW Shared Workspace System
BSCW
Basic Support for Cooperative Work

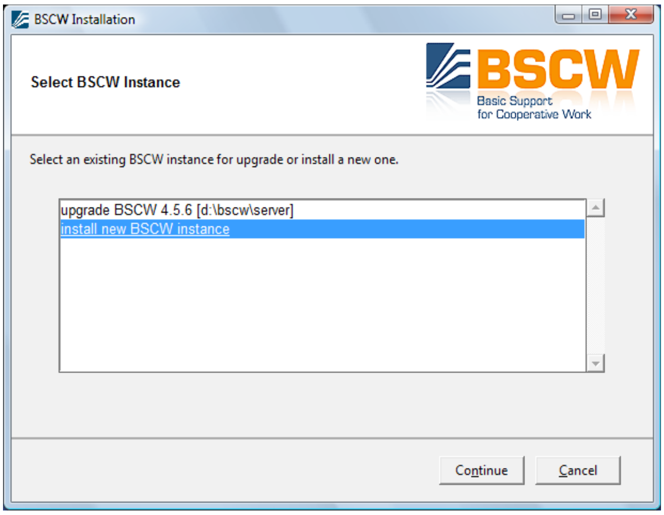
Completing the BSCW Shared Workspace System Setup Wizard

Setup has finished installing BSCW Shared Workspace System on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

☑ Install a server instance now

The BSCW instance setup program will examine your system, and if a BSCW instance runtime is found the following selection is shown:



Select BSCW Instance

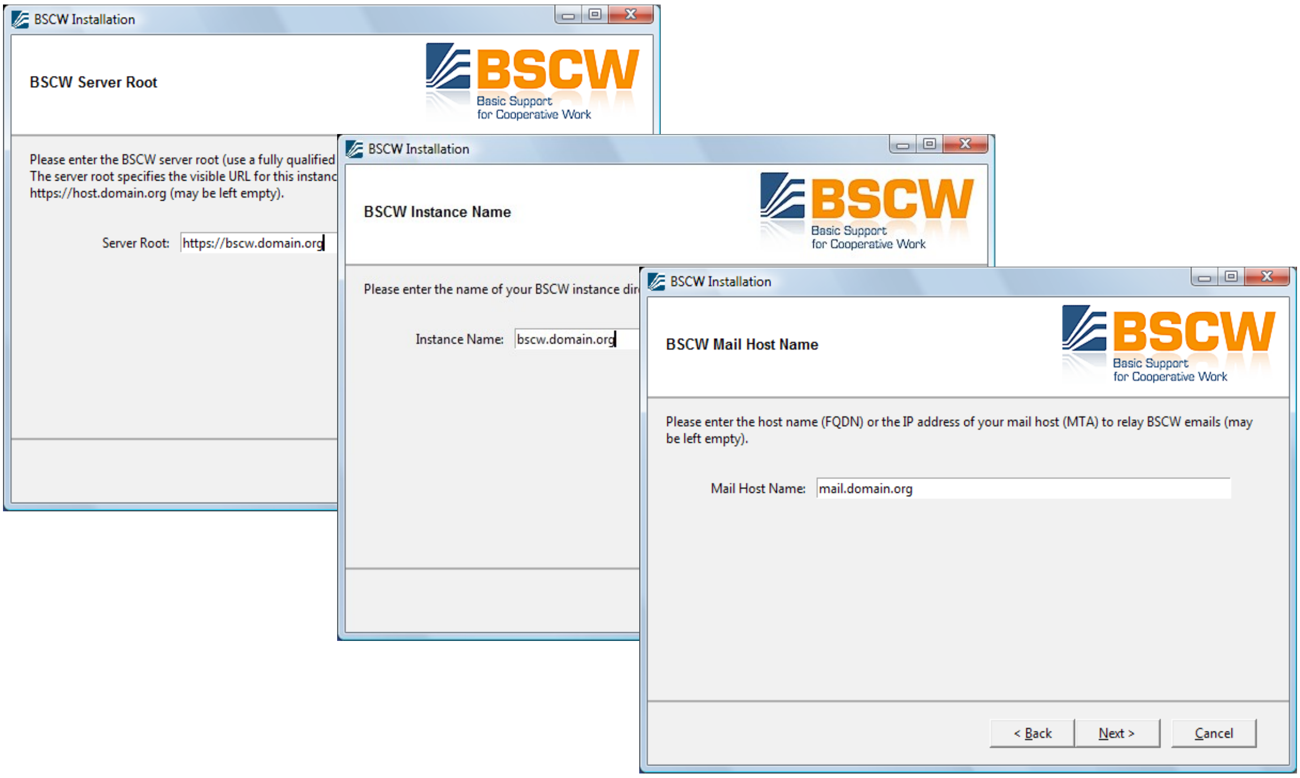
Select an existing BSCW instance for upgrade or install a new one.

upgrade BSCW 4.5.6 [d:\bscw\server]
install new BSCW instance

select "install new BSCW instance" and click `[Continue]`. If no BSCW instance runtime is found this step is omitted. Next a console window and a second setup window are opened. To perform an initial instance configuration the following configuration details must be entered:

- *BSCW server root, instance name and mail host name*
  The server root specifies the visible URL for this instance, while the instance name specifies the directory name of the BSCW instance. Next the mail host name is required by BSCW to relay emails. Please enter the (FQDN) or the IP address of your mail host (MTA). While a working MTA is **mandatory** for BSCW operation, you may leave the server root definition empty for later configuration (see also section 4.4.1).**Note**: if you you are using MS Exchange as MTA,

you must explicitly allow the IP address of you BSCW server host to ***relay*** email.



BSCW Installation

BSCW Server Root

Please enter the BSCW server root (use a fully qualified
The server root specifies the visible URL for this instanc
https://host.domain.org (may be left empty).

Server Root: https://bscw.domain.org

BSCW Installation

BSCW Instance Name

Please enter the name of your BSCW instance dir

Instance Name: bscw.domain.org

BSCW Installation

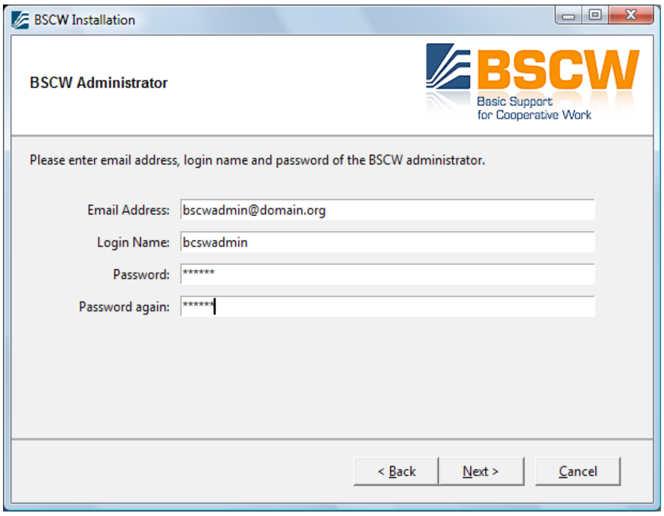
BSCW Mail Host Name

Please enter the host name (FQDN) or the IP address of your mail host (MTA) to relay BSCW emails (may be left empty).

Mail Host Name: mail.domain.org

- *BSCW administrator*

  Enter a **valid** BSCW server administrator email address, an user name and a password of the user who shall become a BSCW server administrator



BSCW Installation

BSCW Administrator

Please enter email address, login name and password of the BSCW administrator.

Email Address: bscwadmin@domain.org

Login Name: bcswadmin

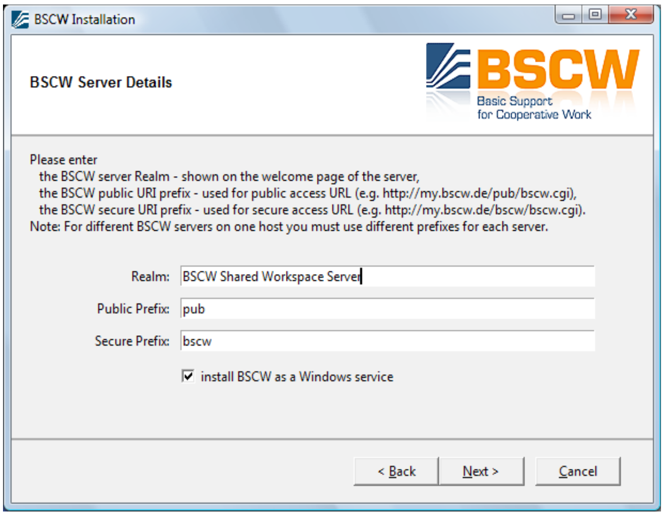
Password: ******

Password again: ******

Note: The login name denotes the account of the BSCW administrator and not any Windows account.

- *BSCW server details*

  Finally you have to define the following server detail information. The BCW server realm is shown in authentication dialogs or on the welcome page of the BSCW instance. The BSCW public prefix defines the path after the server root used to allow (unauthenticated) access to published BSCW contents, e.g `https://bscw.domain.org/pub/`, while the BSCW secure prefix defines the path after the server root used to access personal (authenticated) BSCW contents, e.g `https://bscw.domain.org/bscw/`. It is recommended to install BSCW as a service. Further information can be found in section 4.3.1.

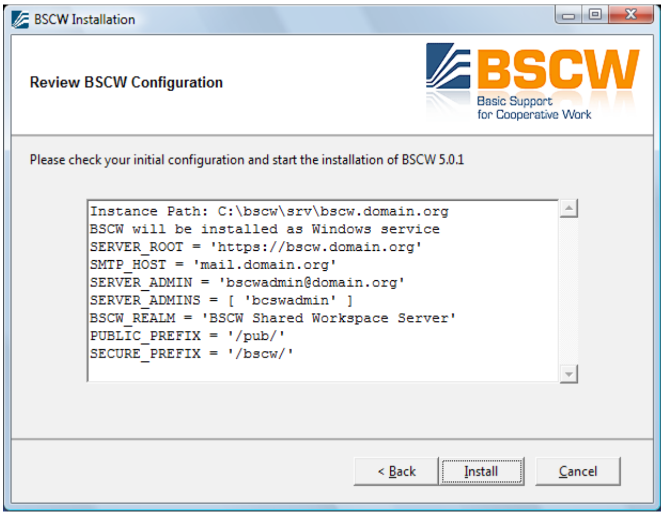


Please enter
the BSCW server Realm - shown on the welcome page of the server,
the BSCW public URI prefix - used for public access URL (e.g. http://my.bscw.de/pub/bscw.cgi),
the BSCW secure URI prefix - used for secure access URL (e.g. http://my.bscw.de/bscw/bscw.cgi).
Note: For different BSCW servers on one host you must use different prefixes for each server.

Realm: BSCW Shared Workspace Server
Public Prefix: pub
Secure Prefix: bscw

☑ install BSCW as a Windows service

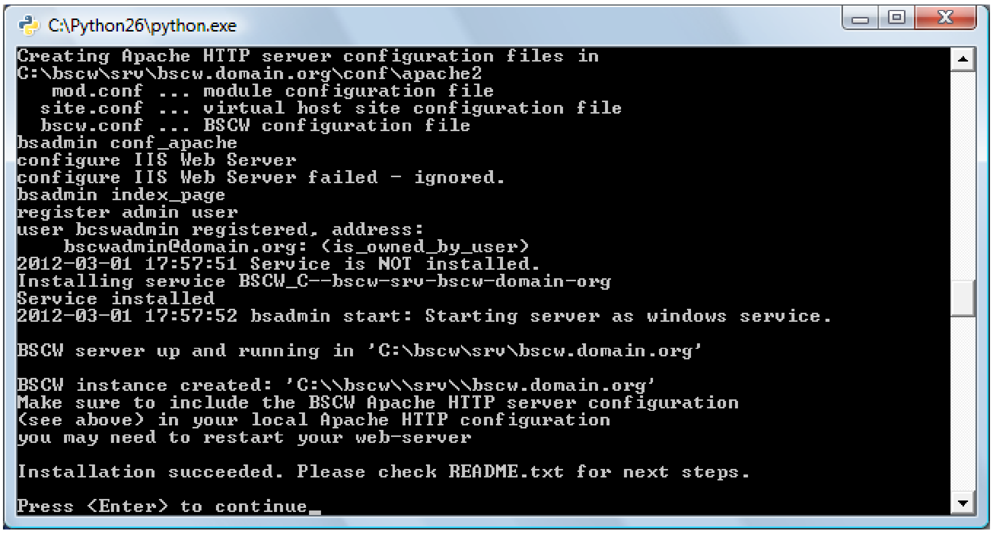
Finally a summary of your settings is shown. You can correct any wrong settings by using the `[Back]` button.



Please check your initial configuration and start the installation of BSCW 5.0.1

```
Instance Path: C:\bscw\srv\bscw.domain.org
BSCW will be installed as Windows service
SERVER_ROOT = 'https://bscw.domain.org'
SMTP_HOST = 'mail.domain.org'
SERVER_ADMIN = 'bscwadmin@domain.org'
SERVER_ADMINS = [ 'bcswadmin' ]
BSCW_REALM = 'BSCW Shared Workspace Server'
PUBLIC_PREFIX = '/pub/'
SECURE_PREFIX = '/bscw/'
```

If all settings are correct press the `[Install]` button, which will then start the BSCW instance configuration. The progress is shown in a console window



```
Creating Apache HTTP server configuration files in
C:\bscw\srv\bscw.domain.org\conf\apache2
   mod.conf ... module configuration file
  site.conf ... virtual host site configuration file
  bscw.conf ... BSCW configuration file
bsadmin conf_apache
configure IIS Web Server
configure IIS Web Server failed - ignored.
bsadmin index_page
register admin user
user bcswadmin registered, address:
    bscwadmin@domain.org: (is_owned_by_user)
2012-03-01 17:57:51 Service is NOT installed.
Installing service BSCW_C--bscw-srv-bscw-domain-org
Service installed
2012-03-01 17:57:52 bsadmin start: Starting server as windows service.

BSCW server up and running in 'C:\bscw\srv\bscw.domain.org'

BSCW instance created: 'C:\\bscw\\srv\\bscw.domain.org'
Make sure to include the BSCW Apache HTTP server configuration
(see above) in your local Apache HTTP configuration
you may need to restart your web-server

Installation succeeded. Please check README.txt for next steps.

Press <Enter> to continue_
```

Depending on the deployed HTTP server, you have to choose one of the following configuration alternatives:

- **Apache HTTP Server**
  If you use the Apache HTTP server (version 2.2 or 2.4) the setup process automatically generated configuration files in the directory `<bscw-runtime-path>/conf/apache{2,24}/`, which contains all necessary configuration instructions. For a more complete Apache HTTP server configuration discussion see the corresponding configuration section 4.4.2. While using “virtual hosts” is recommended, the easiest option is to include the directive
  ```
  Include <bscw-runtime-path>/conf/apache{2,24}/bscw.conf
  ```
  to the file `httpd.conf`.
- **Microsoft Internet Information Server (IIS)**
  If you use the Microsoft Internet Information Server (IIS) all necessary IIS configuration for BSCW is done by the setup script automatically for Windows 7, 8/Server 2012, 2008, 2003. Finally the setup script launches your default Web browser to connect to your BSCW server (see also section 4.4.3 for further IIS configuration details).

Notes:

- It is recommended to use the Apache HTTP 2.2 (or 2.4) server on Windows; when using the Microsoft IIS the WebDAV ("web folders") functionality of BSCW will not be available.
- When using the Microsoft Internet Information Server (IIS) CGI support must be enabled manually **before** the BSCW installer is started, otherwise the automatic configuration of IIS may fail.

After your BSCW instance is running you can log in with the administrator account registered during the installation procedure (see above) by opening the URL (mind login name and password are case sensitive!):

```
http://<server>[:port]/bscw/bscw.cgi
(e.g. http://bscw.domain.org/bscw/bscw.cgi)
```

Actually to gain administrator rights you have to login a second time with your password by opening

```
[Options > Admin]
```

If you open the URL

```
http://<server>[:port]/pub/
(e.g. http://bscw.domain.org/pub/)
```

you get a BSCW overview which contains links to your BSCW instance.

4.3 Database Server and Garbage Collection

All data of the BSCW server is held in the BSCW data store and handled through the BSCW database server. The BSCW database server is managed with the `bsadmin` script, which is located in the BSCW instance directory `<bscw-runtime-path>\bin`. The BSCW server can be administered by executing the `bsadmin` script from a DOS shell as follows:

```
> cd <bscw-runtime-path>
> bin\bsadmin start
```

Starts the BSCW server. If it is registered as a Windows service, the service is run, otherwise the server is started directly. Note: controlling Windows services requires administrative privileges. To avoid calling `bsadmin start` manually, you can set up the Windows service to start up at system boot or use the task scheduler instead (see below).

```
> bin\bsadmin stop
```

Stops the BSCW server. If it is registered as a Windows service, the service is stopped, otherwise the server is stopped directly. Note: controlling Windows services requires administrative privileges.

```
> bin\bsadmin garbage
```

Runs the garbage collection on the BSCW database. Note: the garbage collection requires the BSCW server to run!

```
> bin\bsadmin
```

Lists further administration functions.

We recommend that `bsadmin start` should be executed at system boot. To achieve this you have to register BSCW as Windows service. Furthermore for the garbage collection a task job must be set up, which calls these functions periodically (see below).

The BSCW database garbage collection **must** be run daily. The task of the garbage collector is to find unreferenced, e.g., obsolete objects in the data store and remove them. For performance reasons, a "delete" operation on an object may not remove the respective object physically from the store. If you do not run the garbage collector periodically, the BSCW data store will grow constantly although many of its objects are obsolete. This would waste disk space and may substantially reduce the performance of the BSCW server.

4.3.1 Windows Service

On Windows 7, 8/Server 2012, 2008, 2003 the BSCW server can be run as a *Windows service*. This is an optional, convenient way to launch it in the background without showing a DOS shell.

The BSCW installer offers to register a Windows service, starting up at boot time. If you have chosen to do this, you can use `bsadmin start` and `bsadmin stop` to start/stop the service assuming you are working with administrative privileges.

Additionally you may register resp. removed the Windows service later by executing the `bsadmin` script from a DOS shell as follows:

```
> cd <bscw-runtime-path>
> bin\bsadmin service
```

Displays usage hints.

```
> bin\bsadmin service install
```

Registers the BSCW Windows service (manual startup).

```
> bin\bsadmin service --startup auto install
```

Registers the BSCW Windows service (startup at boot time).

```
> bin\bsadmin service remove
```

Removes the BSCW Windows service

4.3.2 Task Scheduler

On Windows 7, 8/Server 2012, 2008, 2003 use the *task scheduler* to schedule periodic system commands (such as the garbage collection).

At least you have to schedule one job to run the BSCW garbage collector (e.g. once per night). Use the following command line to run the garbage collection:

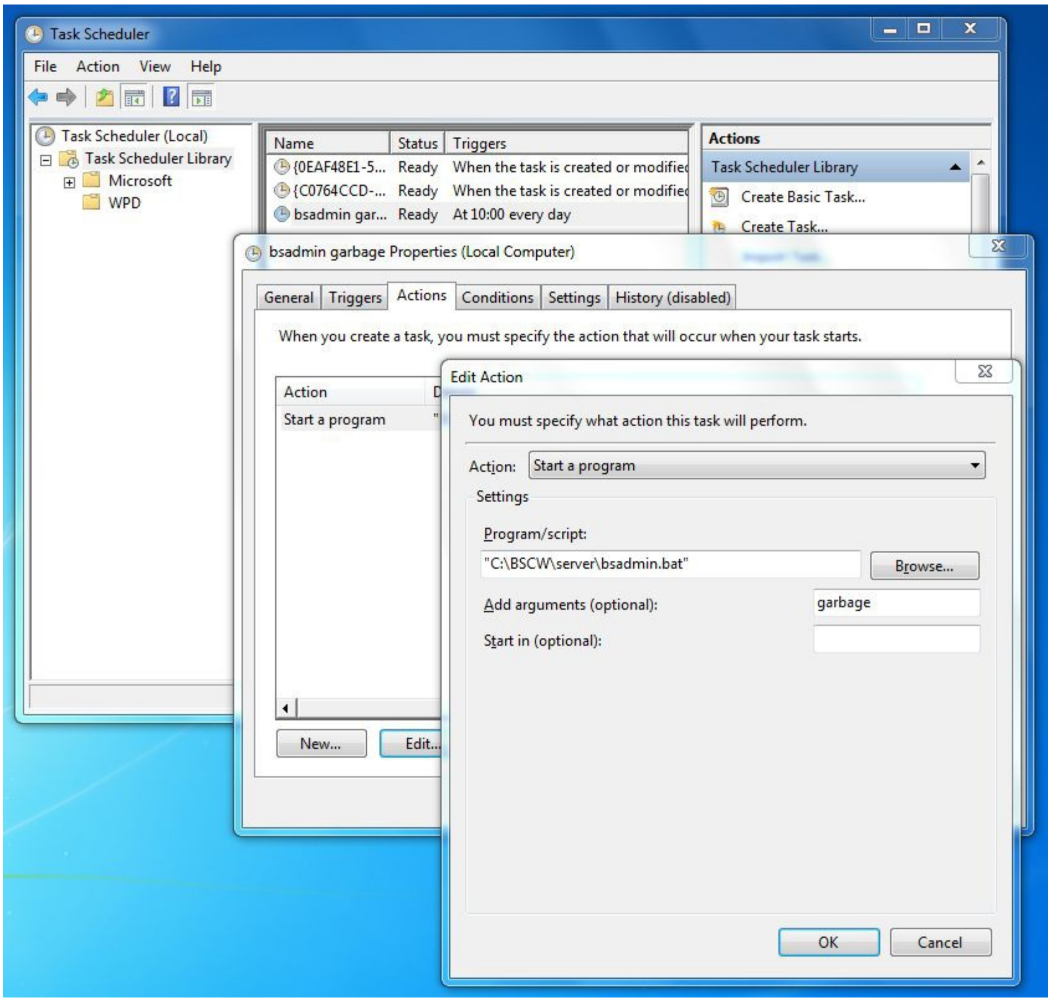
```
"<bscw-runtime-path>\bin\bsadmin.bat" garbage
```

If you do not want to run the BSCW server as a Windows service, you may alternatively use the task scheduler to start it. Use the following command line to define a scheduled job to start BSCW at system boot:

```
"<bscw-runtime-path>\bin\bsadmin.bat" start
```

Note: You may use the command above without quotes if the path names does not contain any spaces. **Important**: the task scheduler requires `bsadmin.bat`.

Note: Do not run the same BSCW server instance more than once! This may seriously damage the BSCW database.



Name	Status	Triggers
{0EAF48E1-5...	Ready	When the task is created or modified
{C0764CCD-...	Ready	When the task is created or modified
bsadmin gar...	Ready	At 10:00 every day

When you create a task, you must specify the action that will occur when your task starts.

You must specify what action this task will perform.

Action: Start a program

Program/script: "C:\BSCW\server\bsadmin.bat"

Add arguments (optional): garbage

Start in (optional):

4.4 Further Configuration Details

All BSCW configuration parameters are stored (similar to the Unix version) in configuration files (see also section 5). These configuration files will be updated during the installation and can be changed by a BSCW administrator on the `[Options > Admin]`-page within the item "BSCW Server Settings" or by directly editing the respective configuration files (see below for further details).

The standard set up should create an installation which should be appropriate in most cases. However, if you want to modify the default settings, you will find respective information in this section. Please note in this section only Windows 7, 8/Server 2012, 2008, 2003 specific configuration options are explained:

- BSCW server root definition
- IIS configuration
- Apache HTTP server configuration
- BSCW registry settings
- De-Installation

4.4.1 BSCW Server Root Definition

The "server root" - the hostname (and port) part of your BSCW servers URL - is specified in the BSCW server instance configuration file at `<bscw-runtime-path>/conf/config.py`. The variable `SERVER_ROOT` contains the absolute URL of your BSCW server and an optional port. If no port is specified the standard ports 80 (for HTTP) or 443 (for HTTPS) are assumed:

```
SERVER_ROOT = 'http://bscw.domain.org/'
SERVER_ROOT = 'http://bscw.domain.org:123/'
SERVER_ROOT = 'https://bscw.domain.org/'
```

A fully qualified host name is required as server name "bscw.domain.org", in order to allow the BSCW server to resolve its name to an IP address (`SERVER_ROOT` may **not** contain an IP address anymore!).

Ideally you define a host name/nickname (`A/CNAME`) in your DNS zone, which points to your BSCW server host, e.g.

```
server1.domain.org     A          1.2.3.4
server2.domain.org     A          1.2.3.5
bscw.domain.org        CNAME      server1.domain.org
```

Proceeding this way a future migration of your BSCW server from `server1` to `server2` will keep the well known URL http://bscw.domain.org/ and your license will not be invalidated by the migration.

Note: whenever the `SERVER_ROOT` is changed in the instance configuration file (`<bscw-runtime-path>\src\config.py`) you must call "`bsadmin update_helper`" in order to update the jnlp deployment files with the correct codebase URL. Otherwise users may not be able to launch or install the BSCW Desktop application anymore.

4.4.2 Apache HTTP Server Configuration

BSCW requires in addition to a (virtual) web server for user access, a second (virtual) web server running on `localhost` (`127.0.0.1`). This second (virtual) web server enables BSCW services (e.g. the user notification (*UNO*) service or the alarm (*ALARM*) service) to access the BSCW database server via HTTP using the following URL:

```
        http://localhost/pub/bscw.cgi
```

Note: The port, the script alias path and the script name may be changed by altering the configuration directives `HTTP_LOCAL_PORT`, `SCRIPTS` and `CREATE_SCRIPTS` in the instance configuration file (`<bscw-runtime-path>/conf/config.py`)

Note: the localhost port to the HTTP server defined in `HTTP_LOCAL_PORT` **must** support HTTP; HTTPS is **not** supported!

The BSCW setup process automatically generates the following Apache HTTP server (version 2.2 or 2.4) configuration files

```
        <bscw-runtime-path>/conf/apache{2,24}/mod.conf
        <bscw-runtime-path>/conf/apache{2,24}/site.conf
        <bscw-runtime-path>/conf/apache{2,24}/bscw.conf
```

which contain all necessary configuration instructions.

The `mod.conf` file ensures the loading of additional modules required by BSCW and must be included in the main Apache HTTP server configuration file (`httpd.conf`). Instead including this file you could enable the loading of the required modules

```
        cgid_module (or cgi_module)
        expires_module
        deflate_module
        headers_module
        rewrite_module
```

using your platform specific Apache layout.

The `site.conf` file contains several virtual host containers. Depending on your `SERVER_ROOT` definition in the instance configuration file (`<bscw-runtime-path>/conf/config.py`) the `site.conf` file defines the following virtual hosts:

1. if a HTTP server root is defined (e.g. the `SERVER_ROOT` directive starts with `http://...`) the `site.conf` file defines two virtual host containers: a first virtual host container for `localhost:80` required by internal BSCW services and a second virtual host container for the server root host name `<hostname>:80` for requests.
2. if a HTTPS server root is defined (e.g. the `SERVER_ROOT` directive starts with `https://...`) the `site.conf` file defines three virtual host containers: a first virtual host container for `localhost:80` required by internal BSCW services, a second virtual host container for the server root host name `<hostname>:80` which redirects all requests to the third virtual host container `<hostname>:443` for SSL requests.

Both files include the `bscw.conf` file with the actual BSCW instance configuration. If you intend to use the `site.conf` file copy it to your Apache HTTP server configuration. Please note it will most likely not work out of the box, but you have to adopt it to your local Apache HTTP server configuration. Especially you will need to install certificates for your SSL enabled server and adopt the configuration in `site.conf`.

The `bscw.conf` file contains the actual BSCW instance configuration for the Apache HTTP server. It may be included in the main configuration file `httpd-vhosts.conf` resp. `httpd-ssl.conf` if you manually define virtual hosts (within the standard Apache HTTP server layout) or in `httpd.conf` without defining virtual hosts:

```
        Include <bscw-runtime-path>/conf/apache{2,24}/bscw.conf
```

When using virtual web server container (`<VirtualHost> ... </VirtualHost>`) directives, it is possible to include the `<bscw-runtime-path>/conf/apache{2,24}/bscw.conf` configuration file in multiple virtual web server container. An example for a virtual web server definition in the Apache HTTP server configuration file file should look like

```
<VirtualHost bscw.domain.org:80>
    ServerName  bscw.domain.org
    ServerAlias localhost
    ServerAdmin hostmaster@domain.org

    ErrorLog    logs/bscw_domain_org_error_log
    CustomLog   logs/bscw_domain_org_access_log common
    ScriptLog   logs/bscw_domain_org_error_log

    DocumentRoot "<bscw-path>/var/www"
    <Directory "<bscw-path>/var/www">
        Options                 ExecCGI FollowSymLinks MultiViews
        AllowOverride           None
        DirectoryIndex          index.html default.htm
        LanguagePriority        en de es fr
        ForceLanguagePriority   Fallback
        # Apache 2.2
        Order                   deny,allow
        Allow                   from all
        # Apache 2.4
        #Require                 all granted
    </Directory>

    Include     "<bscw-runtime-path>/conf/apache{2,24}/bscw.conf"
</VirtualHost>

<VirtualHost bscw.domain.org:80>
    ServerName  bscw.domain.org
    ServerAdmin hostmaster@domain.org

    ErrorLog    logs/bscw_domain_org_error_log
    CustomLog   logs/bscw_domain_org_access_log common
    ScriptLog   logs/bscw_domain_org_error_log

    DocumentRoot "<bscw-path>/var/www"
    <Directory "<bscw-path>/var/www">
        Options                 ExecCGI FollowSymLinks MultiViews
        AllowOverride           None
        DirectoryIndex          index.html default.htm
        LanguagePriority        en de es fr
        ForceLanguagePriority   Fallback
        # Apache 2.2
        Order                   deny,allow
        Allow                   from all
        # Apache 2.4
        #Require                 all granted
    </Directory>
```

```
        Include     "<bscw-runtime-path>/conf/apache{2,24}/bscw.conf"
    </VirtualHost>
```

To provide a SSL encrypted web site your virtual web server definition should look like (Note: additionally you will still require a HTTP web server on `localhost` as defined above).

```
    <VirtualHost bscw.domain.org:80>
        ServerName  bscw.domain.org
        ServerAdmin hostmaster@domain.org

        ErrorLog    logs/bscw_domain_org_error.log
        CustomLog   logs/bscw_domain_org_access_log common
        ScriptLog   logs/bscw_domain_org_script.log

        <IfModule alias_module>
            RedirectMatch permanent ^/(.*)$ https://bscw.domain.org/$1
        </IfModule>
    </VirtualHost>

    <VirtualHost bscw.domain.org:443>
        ServerName  bscw.domain.org
        ServerAdmin hostmaster@domain.org

        ErrorLog    logs/bscw_domain_org_error.log
        CustomLog   logs/bscw_domain_org_access_log common
        ScriptLog   logs/bscw_domain_org_script.log

        DocumentRoot "<bscw-runtime-path>/var/www"
        <Directory "<bscw-runtime-path>/var/www">
            Options                  ExecCGI FollowSymLinks MultiViews
            AllowOverride            None
            DirectoryIndex           index.html default.htm
            LanguagePriority         en de es fr
            ForceLanguagePriority    Fallback
            # Apache 2.2
            Order                    deny,allow
            Allow                    from all
            # Apache 2.4
            #Require                  all granted
        </Directory>

        SSLEngine on
        SSLVerifyDepth 5

        #SSLCACertificateFile     conf/ssl/ca-bundle.crt
        #SSLCertificateChainFile  conf(ssl/bscw_domain_org_root.crt
        SSLCertificateKeyFile    conf/ssl/bscw_domain_org.key
        SSLCertificateFile       conf/ssl/bscw_domain_org.crt

        Include     "<bscw-runtime-path>/conf/apache{2,24}/bscw.conf"
        </VirtualHost>
```

You may change the BSCW Apache HTTP server configuration file by using the `bin/bsadmin conf_apache` script. To adopt the generated Apache configuration file to your local web server settings use one of the following options:

- If no option is used `bin/bsadmin conf_apache` tries to read the old option setting from `bscw.conf` (if exists). Use option `-n` or remove `bscw.conf` if you want to avoid this.
- If option `-r` is used (requires "`rewrite`" module) the user credentials are passed that the authentication is handled by the BSCW server (this is the default case).
- If option `-a` is used, BSCW allows to let the Apache HTTP server perform authentication (**note**: certain configurations (such as cookie authentication) imply option `-r`).
- If option `-s` is used the Apache HTTP server is configured for authentication via client certificates. This option includes the `-r` option and requires a SSL enabled server.
- If option `-o` is used client certificates authentication optional. This option includes the `-r` option and requires a SSL enabled server.
- If the `-D` or `-E` options are used the Apache HTTP server is configured to compress (`gzip`) BSCW resources (`-D`) or to cache resources due to a long time future expiry date (`-E`). This options require the "`deflate`" (`-D`) or the "`expires`" (`-E`) modules (these options are enabled by default).

Notes:

- If you are running several BSCW instances in different virtual hosts you must configure for each BSCW instance a different `HTTP_LOCAL_PORT` number and you must extend the `VirtualHost` directives by these local IP addresses/port pairs.
- It might be necessary to add an extra `Listen 127.0.0.1:<HTTP_LOCAL_PORT>` directive to the Apache HTTP server configuration file `httpd.conf`.

Remember to always **restart** your Apache HTTP server whenever the `bsadmin conf_apache` script was run.

4.4.3 IIS Configuration

The BSCW server requires additional virtual directory mappings of your Web server. They depend on the values specified for the `RESOURCES_PREFIX` and the `SCRIPTS` variables in the BSCW server configuration. By default the virtual directory mappings are

```
/bscw              C:\BSCW\srv\<runtime>\var\www
/pub               C:\BSCW\srv\<runtime>\var\www
```

These directory mappings are set and configured **automatically** for IIS by BSCW on Windows 7, 8/Server 2012, 2008, 2003 using the installation program `bsadmin conf_iis`.

Note: *before* running `bsadmin conf_iis` you have to manually activate the CGI support for your IIS.

Manual IIS 6 configuration

If the Internet Information Services 6 are not installed on your system follow these instructions to install IIS6:

- open `Control Panel > Add Remove Programs`
- enter `Add/Remove Windows Components`
- select `Internet Information Services`

Press `OK` to install IIS6. After installation configure IIS6 (on Windows XP) as follows:

- open `Control Panel > Administrative Tools > Internet Information Services`
- select `<computer> (local computer) > Web Sites > Default Web Site`
- add new virtual directories
  - select in the context menu of the *`Default Web Site`*
    `New > Virtual Directory...`
  - step through the *`Virtual Directory Creation Wizard`* and add the following directory mappings:

    ```
    pub            -> <bscw-runtime-path>\var\www
    bscw           -> <bscw-runtime-path>\var\www
    ```

    On the last tab (`Access Permissions`) allow `Read` for all virtual directories and `Execute` for `pub` and `bscw`.
- configure virtual directories
  - select in the context menu for each virtual directory `pub` respective `bscw`
    `Properties > Virtual Directory > Configuration...`
  - open *`Application Configuration`* `> Mappings > Add` and enter in *`Add/Edit Application Mapping`*
    Executable: `"<python-path>\python.exe" -u "%s"`
    Extension: `.cgi`
    deselect the option *`Check that files exists`*
  - select *`Properties`* `> Directory Security`
    `> Anonymous access and authentication control > Edit...`
  - in *`Authentication Methods`*
    select *`Anonymous access`*
    deselect *`Authenticated access – Integrated Windows authentication`*
  - in *`Properties > Custom Errors`* `> HTTP Error 401;5` click on `Edit Properties...` and set in *`Error Mapping Properties`* to
    message type: `Default`

Manual IIS 7 or 8 configuration

If the Internet Information Services 7 or 8 are not installed on *Windows Server 2008, 2012* follow these instructions to install IIS:

- open the Server Manager, Roles section or open
  `Control Panel > Programs > Programs and Features`
  `> Turn Windows features on or off`
- select `Add Roles` and follow the wizard
- activate `[X] Web Server (IIS)` on the "Server Roles" page. In case you are asked to install more required features on *Windows Server 2012* (e.g. IIS Management Console) agree to include *management tools.*
- activate

  ```
  Web Server
     Application Development
        [x] CGI
  ```

  on the "Role Services" page

Complete the installation steps of the wizard to install IIS. After installation configure IIS as follows:

- (*Windows Server 2008*) open `Start > Administrative Tools > Internet Information Services (IIS) Manager`
- (*Windows Server 2012*) open `Server Manager > Tools > Internet Information Services (IIS) Manager`
- select `<computer> > Sites > Default Web Site`
- add new virtual directories
  - select in the context menu of the *`Default Web Site`*

```
        Add Virtual Directory...
```

- add the following directory mappings:

```
        pub           -> <bscw-runtime-path>\var\www
        bscw          -> <bscw-runtime-path>\var\www
```

- configure virtual directories
  - open the virtual directory for `pub` resp. `bscw` and choose for each directory

```
     Feature View  > Handler Mappings
```

  - add in *`Handler Mappings`* a script mapping with `Add Script Map...`

    Request Path: `*.cgi`
    Executable: `"<python-path>\python.exe" -u "%s"`
    Name: `Python Script`

  - open `Request Restrictions... > Mapping` and deselect

```
        Invoke handler only if request is mapped to:
```

  - confirm pop up windows of each *`Add Script Map`* question with *`OK`*.

BSCW uses its own built-in authentication scheme to check the access for the virtual directory `bscw`. Therefore no authentication filter is necessary. These configurations will be done for Windows 7, 8/Server 2012, 2008, 2003 by the BSCW administration command `bsadmin conf_iis`.

Notes:

- `bsadmin conf_iis` should be sufficient for automatic configuration of IIS version 6, 7 or 8
- The path to the Python interpreter `<python-path>` may **not** contain any space character (such as "`C:\Program Files\Python26\python.exe`"), otherwise IIS will generate a broken configuration for itself!

Hint: If your path `<python-path>` contains space characters alternatively you can use the Windows short filename (SFN or "8.3 filename") to the `python.exe` executable.

4.4.4 De-Installation

The BSCW de-installation procedure only allows to remove BSCW libraries which are no longer in use by any installed BSCW instance. To de-install old BSCW libraries, start the BSCW de-installer program in the systems control panel.

4.5 Folder Mail Delivery

BSCW does not support BSCW folder mail delivery on Windows.

5 Configuration of BSCW Servers

The BSCW server can be configured by a set of configuration files which are stored in the instance configuration directory `<bscw-runtime-path>/conf/`. The standard configuration files in the instance configuration directory `<bscw-runtime-path>/conf/` are:

• `config.py`	General configuration of the BSCW server
• `config_actions.py`	Customization to default action or role definitions
• `config_cal.py`	Configuration of the calendar
• `config_clientmap.py`	Configuration of web browser capabilities
• `config_controls.py`	Customization to default operation control definitions
• `config_convert.py`	Specification of archivers, encoders and converters
• `config_countries.py`	Specification of country codes
• `config_ext.py`	Extensions configuration
• `config_help.py`	Contains online help mappings
• `config_html_ui.py`	HTML user interface of the BSCW server
• `config_icons.py`	Icons in the user interface of the BSCW server
• `config_meet.py`	Configuration of synchronous meeting facilities
• `config_menu.py`	Configuration of the menu layout
• `config_metadata.py`	Configuration of meta data
• `config_mimegroups.py`	Application MIME type grouping
• `config_mimeicons.py`	MIME type icons
• `config_mimemsg.py`	Additional translations for MIME type specification
• `config_mime.py`	MIME type specifications
• `config_mobile_ui.py`	Mobile user interface configuration
• `config_mpick.py`	Define class substitutions for deactivated packages
• `config_prio_categ.py`	Configuration of priorities and categories

Please note the editing instructions within these files carefully when making any modifications. It should be noted that all configuration files are Python modules and on thus subject to Python's programming language syntax. After an overview of different user authentication possibilities, the above configuration files are described in this section.

5.1 Authentication

BSCW provides for each user a personal view of accessible data. To access this personal view on the BSCW system every user needs to authenticate with an individual user name or an email address (allocated to an user name) and a password. In general BSCW offers two possibilities to perform this authentication

1. *BSCW Authentication*

   In general BSCW authentication is passing user credentials via an environment variable to BSCW (*cookie / basic / digest authentication*). Passing the users' credentials to BSCW gives the most flexibility to react on authentication challenges (see also section 2.2. *Security considerations*).

   Using BSCW authentication enables the usage of different (configured) methods, for example such as querying an LDAP service for users with an LDAP binding, or redirect to a single sign

on service to perform an external user authentication.
Additional features like

- authentication with (registered) email address and password
- BSCW logout
- automatic fall-back to *basic / digest authentication* for WebDAV clients (since the WebDAV protocol does not support cookie authentication).

are only available using BSCW authentication.

BSCW uses cookie authentication as default authentication method.

2. *Web Server Authentication*
   is the "classical" way BSCW handles authentication. To utilize the Apache web serves' basic authentication module the (encrypted) user credentials are stored within a file `htpasswd` which is shared between the BSCW server and the Apache web server. While BSCW maintains this file, the Apache web server uses it to check the given user credentials before BSCW may be accessed via its `bscw.cgi` script.

   Using web server based authentication allows a "cascading" use of diverse Apache authentication modules. This enables for instance the implementation of an efficient authentication lookup against an organizational LDAP service (using the Apache `mod_ldap.so`). For failed LDAP authentication attempts then as second stage the standard basic authentication method is invoked using the shared `htpasswd` file.

   Nevertheless Apache web server authentication may not be flexible enough and has the following limitations:

   - authentication with email address and password is not possible, since at login time the web server cannot check the association between allocated email address and user name
   - the BSCW logout feature is not available
   - the usage of the ZOPE external editor is not possible (due to the used authentication mechanism).

BSCW instances enable *BSCW authentication* using Cookie Authentication as default setting. On older BSCW instances *BSCW authentication* may be explicitly enabled by running the command line script

```
$ ./bin/bsadmin conf_apache -n
```

See section 3.3.1 for details.

5.2 conf/config.py

This file defines the general server settings and server configuration of the BSCW server instance. Please note that all relative file and directory names are resolved using instance runtime directory `<bscw-runtime-path>`.

Below the names of the configuration variables, their meaning and their default settings are given. At least the variables mentioned in the "`Section 1: MANDATORY server settings`" the configuration file **must be set** since their default setting is not sufficient.

5.2.1 Section 1: MANDATORY server settings

`SERVER_ROOT`	The Web servers' root address. This should be an absolute URL specifying

- the protocol (`http` or `https`)
- the fully qualified domain name of the server (a numeric IP address is not allowed here)
- and (optionally) the port number

See section 3.3.2 for configuration hints of `SERVER_ROOT`. For example:

```
SERVER_ROOT = 'http://bscw.domain.org'
SERVER_ROOT = 'http://bscw.domain.org:8000'
SERVER_ROOT = 'https://bscw.domain.org'
```

Important Notes:

- You have to set `SERVER_ROOT` to before you apply for a BSCW license.
- A granted BSCW license (not the evaluation license) will become invalid if you change `SERVER_ROOT` or the standard `SCRIPTS` prefix (see below). In this case BSCW will complain with a “license error” message after the BSCW database server is restarted or the garbage collector has run. Hence, you need to apply for a new (royalty-free) “change server”- license after changing the values of `SERVER_ROOT` or the standard `SCRIPTS` prefix. Of course, you might also reset `SERVER_ROOT` and `SCRIPTS` to the old values and restart (stop and start) the BSCW database server.
- Whenever the `SERVER_ROOT` is changed you must run "`bsadmin update_helper`" in order to update the jnlp deployment files with the correct code base URL. Otherwise users may not be able to launch or install the BSCW Desktop application anymore.

```
SERVER_ROOT = 'http://bscw.domain.org'
```

`SERVER_ADMIN`

The mail address of the BSCW administrator. `SERVER_ADMIN` **must** be set to a valid and complete mail address.

```
SERVER_ADMIN = 'bscwadmin@domain.org'
```

`SERVER_ADMINS`

List of BSCW users that have administrator rights, e.g.:

```
SERVER_ADMINS = [ 'bscwadmin', 'diana', 'peter' ]
```

Note:The users listed here **must** be registered BSCW users and the names must match exactly.

For domain restrictions see also `SERVER_ADMINS_IP`

```
SERVER_ADMINS = [ 'bscwadmin' ]
```

`SERVER_ADMIN_`
`CONTACT`

The mail contact address of the BSCW administrator. This is used to `'mailto:SERVER_ADMIN_CONTACT'` in index_page and in Help menu for end users to contact their BSCW administrator by e-mail.

If left empty the `SERVER_ADMIN` e-mail address is used.

```
SERVER_ADMIN_CONTACT = '' or 'mailto:SERVER_ADMIN
SERVER_ADMIN_CONTACT = ''
```

`HTTP_LOCAL_PORT`

`localhost` port to HTTP server. BSCW needs local access to the standard `bscw.cgi` script. The HTTP server must listen to `localhost:<HTTP_LOCAL_PORT>` and must provide access to the BSCW Server via this port.

Notes:

- the '`localhost'` port to the HTTP server **must** support HTTP; HTTPS is **not** supported.
- If you use the Apache `<VirtualHost>` container to accommodate the BSCW script path it may be necessary define an extra virtual host for `localhost:<HTTP_LOCAL_PORT>` where the Apache configuration file `<bscw-runtime-path>/conf/apache{2,24}/bscw.conf` also is included. See also the virtual host container template file `<bscw-instance-path>/conf/apache{2,24}/site.conf` for examples.

```
HTTP_LOCAL_PORT = 80
```

`SMTP_HOST`
`SMTP_AUTH`

A host name or an IP-address of a mail host that accepts mail posting by SMTP.

Using the `SMTP_HOST` option is recommended, because it allows to set sender addresses correctly. If empty, the local mail delivery command as defined in `SENDMAIL` is used. (see also local BSCW mail delivery (`MDA_MTA`)). A non-default port may be specified by appending `:<port>`

```
SMTP_HOST = 'mail.bscw.de:225'
```

Finally `@TLS` or `@SSL` may be appended to `SMTP_HOST`, in order to switch over to TLS (see `smtplib starttls`) or to start right away with SSL (see `smtplib SMTP_SSL`, not supported by python < 2.6). Invalid `SMTP_HOST` setting and BSCW mail transport in general can be debugged with

```
bsadmin sendmail -vv
```

If `SMTP_AUTH` is set to a non-empty string `'login:passwd'` the RFC 2554 SMTP authentication mechanism is used after connecting the mail host. This string contains a pair of login name and clear text password separated by `':'`, e.g. `SMTP_AUTH = 'bscw:secret'`

```
SMTP_HOST = 'mail.domain.org'
SMTP_AUTH = ''
```

5.2.2 Section 2: Mail handling

`SENDMAIL`

A command line accepting mail (header+body) for posting via standard input. The patterns `%(from)s` and `%(to)s` in the `SENDMAIL` string are substituted by the sender and the recipients of the mail respectively (the recipients are separated by spaces).

Note: You may also use a list with arguments which is more secure. The arguments `%(from)s` and `%(to)s` are substituted by the sender and recipients.

```
SENDMAIL = ['/usr/lib/sendmail',  '-f',  '%(from)s',  '%(to)s']
```

```
MDA_MTA
MDA_MBOX
MDA_DOMAIN
MDA_HDRDESCR
MDA_DELIMITER
MDA_EXT
```

Settings for the local BSCW mail delivery agent, which delivers mail directly into folders.

Note: When local BSCW mail delivery is enabled, the BSCW server should only use the local mail server, therefore it is advisable to set `SMTP_HOST = ''`

`MDA_MTA` specifies the local Mail Transfer Agent (MTA), currently supported are:

```
MDA_MTA = 'qmail'
MDA_MTA = 'postfix'
MDA_MTA = 'sendmail'
```

Setting `MDA_MTA = ''` or any unknown MTA will disable the BSCW mail delivery feature (this is the default).

`MDA_MBOX` defines the local mailbox name for BSCW mda (this is normally the BSCW user ID name)

`MDA_DOMAIN` defines the domain name of the BSCW MDA (which is the delivery domain of the local MTA for the local BSCW MDA mailbox)

`MDA_HDRDESCR` defines which headers are shown in the description of an uploaded email, e.g. `MDA_HDRDESCR = ['From', 'To', 'Cc']`

`MDA_DELIMITER = None` (optional) allows to override the MTA default recipient delimiter:

```
MDA_DELIMITER = '+' (sendmail/postfix)
MDA_DELIMITER = '-' (qmail)
```

`MDA_EXT = True` (optional) appends the extension for the MIME type `'message/rfc822'` (as defined in `config_mime.py`: `.eml` or `.mht`) to the email name.

```
MDA_MTA = ''
MDA_MBOX = 'bscw'
MDA_DOMAIN = 'domain.org'
MDA_HDRDESCR = [ 'From', 'To', 'Cc' ]
```

```
SEND_LIMIT
```

`SEND_LIMIT` is a tuple of `(soft_limit, hard_limit)`. If an email should be send by the send operation and the message becomes larger than the `soft_limit`, the user gets an hint that, he will send a large email. If the message is larger then the `hard_limit`, the sending of the message will be rejected. If one or both of the limits are `0`, the test or both tests will be suppressed.

Possible values for the sizes are strings which may be specified in bytes or kilo- (mega-, giga-, tera-) bytes with an additional `k` (`K`), `M`, `G` or `T` suffix.

E.g. valid values for ten mega-bytes are `10485760` or `'10M'`.

```
SEND_LIMIT = ('10M', '20M')
```

`TOKEN_EXP`

Send documents as links by email. When sending documents as links (send operation), email recipients will be sent the link including a temporary token. This token grants "get" access to the document for anonymous users, until the token expires. `TOKEN_EXP` is the tokens' life span in seconds.

- possible values for the interval may be specified in seconds or minutes (hours, days, weeks) with an additional `'m'` (`'h'`, `'d'`, `'w'`) suffix. E.g. valid values which specify one week are `604800` (or `'604800'`), `'7d'` or `'1w'`.
- `TOKEN_EXP = None` will entirely disable option to send tokens; links can then only be sent to registered users with "get" right.

Note: `TOKEN_EXP < 600` sec (`10` min) will entirely disable option

```
TOKEN_EXP = '4w'
```

`SEND_ADMIN`

If set it specifies an *email address* which will be used as sender instead of the `SERVER_ADMIN`, when an user sends an email via the send operation.

```
SEND_ADMIN = ''
```

`SEND_RETURN_PATH`

If set it specifies an *email address*, which will be used as `Return-Path` in the envelope when mails are sent. Otherwise the email address from the `From` field is used (which is considered to be a valid email address, because it has been verified by BSCW).

```
SEND_RETURN_PATH = ''
```

`FAX_SERVER`
`FAX_ATTACHMENTS`

`FAX_SERVER` is the IP address (or hostname) of a fax server with a mail gateway (for example the HylaFAX server, http://www.hylafax.org/). If it is not empty, BSCW offers an interface to send faxes with this server, i.e. the BSCW server sends either a `text/plain` or a `multipart/mixed` message with a fax to this server. To configure this message edit the template `fax.mail` in the BSCW messages directory:

```
FAX_SERVER = 'fax.domain.org'
```

`FAX_ATTACHMENTS` contains a list of mime types for documents, that can be send in a `multipart/mixed` message to the fax server.

```
FAX_SERVER = ''
FAX_ATTACHMENTS = [ 'text/plain', 'application/postscript' ]
```

5.2.3 Section 3: Server access

`SERVER_ADMINS_IP`

List (or tuple) of IP domain addresses. If not empty the remote address must match one of the given domains for a user in SERVER_ADMINS to become BSCW Administrator (see below).

```
SERVER_ADMINS_IP = [
   '1.2.3.4',          # administrator IP address
   '127.0.0.1',        # localhost
]
```

```
SERVER_ADMINS_IP = []
```

MAY_REGISTER
ALLOW_MAIL_AUTH

MAY_REGISTER defines a list of BSCW users who have the right to register mail addresses - i.e. to invite new users to the system or to a workspace. This is in addition to SERVER_ADMINS who have this right anyway.

There are three special cases: if MAY_REGISTER is

- [] Registration of new email addresses is allowed for all users. This allows all registered users to invite new users to the system. Also self-registration is possible.
- None Registration is allowed for all registered users, but self-registration is forbidden.

Note: Only MAY_REGISTER=[] allows self-registration by URL

```
http://bscw.domain.org/pub/bscw.cgi?op=rmail
```

If ALLOW_MAIL_AUTH is set True (default) users may reset their password via mail token authentication. If set False mail token authentication is disallowed (and only the administrator may reset forgotten passwords).

```
MAY_REGISTER = None
ALLOW_MAIL_AUTH = True
```

REGISTER_DETAILS
TERMS_AND_CONDITIONS
PRIVACY_POLICY

A list of user details that must be filled in at registration time by new users. Select a subset of the following user attributes:

fullname (users' full name), *org* (Organization), *phone* (work phone). *fax* (work fax), *homephone* (private phone), *mobile* (mobile phone), *post* (postal address). In order to inquire the user for his full name and phone number at registration you would configure:

```
REGISTER_DETAILS = [ 'fullname','org', 'phone', 'fax',
#                    'homephone', 'mobile', 'post' ]
```

or allow registration without additional details, simply set

```
REGISTER_DETAILS = None
```

If TERMS_AND_CONDITIONS or PRIVACY_POLICY are defined and point to a link (URL), the registration form will be extended by check-boxes which the user has to confirm before finishing the registration process. To support language dependent links, add the language shortcut (uppercase) to the variable name, e.g. use TERMS_AND_CONDITIONS_DE for a German page. **Note**: English is the default language which is displayed for all languages without language dependent link.

```
REGISTER_DETAILS = None
TERMS_AND_CONDITIONS = ''
PRIVACY_POLICY = ''
```

`MAY_UNREGISTER`

Defines if BSCW users have the right to unregister (i.e. self-destroy their account).

`MAY_UNREGISTER` may not be configured for individual accounts but only be set to `True` or `False`, i.e. enabled or disabled for all users (except for administrators).

```
MAY_UNREGISTER = False
```

`RESTRICT_MAIL`

A list (or tuple) of pairs (pattern, boolean) which restrict the set of email addresses that are accepted by BSCW for e.g. registration or invitation.

If the list is not empty, then a new mail address is translated to lower case and matched against the patterns (see Python module `re`). The boolean of the first matching pattern decides, if the mail address is accepted by BSCW. For example

```
RESTRICT_MAIL = [('[^@]*@orbiteam.de', 1), ]
```

restricts accepted email addresses to the single domain `orbiteam.de`.

Note: `RESTRICT_MAIL` does not apply, if the inviting user is in the list of `SERVER_ADMINS`.

```
RESTRICT_MAIL = []
```

`RESTRICT_SEND`

A list (or tuple) of pairs (pattern, boolean) which restricts the set of users that are allowed to send email via BSCW. If the list is not empty, then the primary mail address of a user is translated to lower case and is matched against the patterns (see Python module `re`). The boolean of the first matching pattern decides, if a user may send email. For example

```
RESTRICT_SEND = [('[^@]*@orbiteam.de', 1), ]
```

restricts users who may send email to all users who have registered with a primary email address in the domain `orbiteam.de`.

```
RESTRICT_SEND = []
```

`RESTRICT_FAX`

If set, it specifies a list (or tuple) of pairs (pattern, boolean) which restrict the set of users that are allowed to send a fax via BSCW. If the list is not empty, then the primary mail address of a user is translated to lower case and is matched against the patterns (see Python module `re`). The boolean of the first matching pattern decides if a user may send email. For example

```
RESTRICT_FAX = [('[^@]*@orbiteam.de', 1), ]
```

restricts users who may send fax to all users who have registered with a primary email address in the domain `orbiteam.de`.

```
RESTRICT_FAX = ()
```

```
MINPASSLEN
EXPPASS
EXPACCT
LOG_EXPIRED_USERS
BADPASS
CRYPT_SALT
```

User password quality and user account expiry settings

`MINPASSLEN` defines the minimal length of a users local password. **Note**: A dictionary search to avoid the selection of weak passwords is available if the

- '`cracklib`' (http://sourceforge.net/projects/cracklib/)
- '`python-crack'` package (http://www.nongnu.org/python-crack/)

are installed. To enable this feature set

```
MINPASSLEN = 'libcrack'
```

`EXPPASS` defines the time interval users are required to change their passwords.

`EXPACCT` defines the time interval after which users are expired, if they did not login previously.

Possible values for the `EXPPASS` and `EXPACCT` intervals may be specified in seconds or minutes (hours, days, weeks) with an additional `'m'` (`'h'`, `'d'`, `'w'`) suffix. E.g. valid values which specify one week are `604800` (or `'604800'`), `'7d'` or `'1w'`. A value of `0` disables enforced password changing resp. general user expiration.

`LOG_EXPIRED_USERS` specifies the log file where all expired (and re-enabled) user accounts are logged.

`BADPASS` defines the number of failed authentication attempts after an user is locked. A value of 0 disables user password mismatch locking (Note: the use is not advised, enabling this feature allows easy denial of service attacks).

`CRYPT_SALT` specifies the `crypt()` algorithm used when storing password hashes. Possible values for `CRYPT_SALT` are:

- `''` to use the traditional format
- `'_'` to use the BSD DES Extended Format
- `'$1$'` to use the Linux MD5 Modular Format (default)

```
MINPASSLEN = 5
EXPPASS = 0
EXPACCT = 0
LOG_EXPIRED_USERS = 'expired_users'
BADPASS = 0
CRYPT_SALT = '$1$'
```

```
SCAN_FILE
NO_VIRUSES_FOUND
VIRUS_FOUND
```

Define a virus scanner which scans files during upload. The upload of files with a virus will be denied.

`SCANFILE` specifies the command string to scan a file. Use the pattern '`%(file)s'` for the file name. Consider in the parameters of the scan command to scan archive files (like zip) or emails. An empty value will deactivate the scanning.

`NO_VIRUS_FOUND` gives a list of result codes which the scan command will return if no virus is found.

VIRUS_FOUND gives a list of result codes which the scan command will return if a virus is found.

Example for NAI McAffee VirusScan:

```
SCANFILE = 'uvscan --mime --unzip %(file)s'
NO_VIRUS_FOUND = [0,]
VIRUS_FOUND = [13,]
```

Example for ClamAV:

```
SCANFILE = 'clamscan --no-summary %(file)s'
SCANFILE = 'clamdscan --fdpass --no-summary %(file)s'
NO_VIRUS_FOUND = [0,]
VIRUS_FOUND = [1,]
```

```
SCAN_FILE = ''
NO_VIRUS_FOUND = []
VIRUS_FOUND = []
```

5.2.4 Section 4: web/proxy server settings

BSCW_REALM

HTTP Authentication parameter - as set in the Web server configuration. **Note**: If you are running different BSCW servers on one host then you must use a different realm for each server.

```
BSCW_REALM = 'BSCW Shared Workspace Server'
```

USE_HTTP_HOST

If not zero and the Host: header is sent by the client, then the BSCW server will use this header for generation of absolute server URLs.

Otherwise the URL will be taken from the SERVER_ROOT or from the environment variable SERVER_NAME or from the socket.gethostname() method (in this order).

```
USE_HTTP_HOST = 1
```

GZIP_RESPONSE

Compress HTML pages if longer than GZIP_RESPONSE bytes and client supports 'gzip_resonse' (see config_clientmap.py).

GZIP_RESPONSE = 0 is no compression

```
GZIP_RESPONSE = 0
```

COOKIE_ AUTHENTICATION
AUTH_MODE
OPEN_ID_DEFAULT
FEDERATIONS

By default BSCW authenticates user via HTTP cookies. This overrides the user authentication (perhaps already) done by the HTTP server. In order to avoid confusion, the HTTP server should **not** be configured to do authentication when COOKIE_AUTHENTICATION is enabled.

COOKIE_AUTHENTICATION is a triple

```
(tagname, location, timeout)
```

with the following settings

`timeout = None`	do not verify authentication tag (low security!)
`timeout = 0`	authentication tag does not expire
`timeout = n`	authentication tag expires after n minutes
`location = None`	use op_login for BSCW authentication
`location = URL`	jump to URL for authentication

e.g. `COOKIE_AUTHENTICATION = ('bscw_auth', None, 120)`

Cookie authentication is diabled with

```
COOKIE_AUTHENTICATION = None
```

If the MS Internet Information Server (IIS) is used please see: http://support.microsoft.com/support/kb/q176113/

`AUTH_MODE` (authorization mode) specifies the authentication method BSCW uses when user credentials are passed to BSCW. `AUTH_MODE` must be `'Basic'` for basic access authentication or `'Digest'` for digest access authentication.

Attention: "*digest authentication*" is not possible in combination with LDAP or with email address login. If you use one of these features `AUTH_MODE` must be '`Basic'`

`OPEN_ID_DEFAULT` is set in order to enable OpenID registration and authentication (see `http://openid.net/`). For example, set

```
OPEN_ID_DEFAULT = ("openid.net",
                   "http://openid.net/get-an-openid/")
```

This will show a link to the "default provider" `openid.net` in the login page. This enables a user to get an OpenID URL if he does not have one. If you do not want to give a link to a default provider set

```
OPEN_ID_DEFAULT = ("", "")
```

Note: `COOKIE_AUTHENTICATION` must be set and location (see above) must be `None` when OpenIDs are used. Also the python-openid package must be installed (tested with Python OpenID 2.2.1, `http://openidenabled.com/python-openid/releases/`).

OpenID registration and authentication is disabled with

```
OPEN_ID_DEFAULT = None
```

`FEDERATIONS` enables support for federation access and single-sign-on. This Feature works in conjunction with path prefixes for anonymous users defined in `SCRIPTS` (see below) and authentication modules provided for the Apache HTTP server (only `mod_shib` is currently supported).

`FEDERATIONS` must be the empty tuple `()` or a dictionary `{}`

```
anonymous_user_name: (login_module, icon, restrict_mail)
```

For example, if you have defined a path prefix for the (anonymous) SnakeOilProviders

```
'/pub/snakeoil/': ('SnakeOilProviders', ...)
```

in the `SCRIPTS` dictionary, you can enforce Shibboleth authentication for accessing `/pub/snakeoil/` with the following entry in `FEDERATIONS`:

```
  'SnakeOilProviders': ('login_shib', '<icon-url>', ()),
```

Parameters:

- `'login_shib'` refers to the BSCW Shibboleth authentication module (the only one that is currently supported in this context).
- BSCW uses the `'<icon-url>'` for a login-button button which is displayed in the login page. A user might click on it in order to authenticate via the federation.
- the third parameter is a list of pairs (see `RESTRICT_MAIL`) that restrict the trusted email addresses for user identification and automatic registration via the federation. You might use `()` if you trust all authenticated email addresses.
- the third parameter may also be set to `RESTRICT_MAIL`. In this case the normal `RESTRICT_MAIL` and `MAY_REGISTER` settings apply. Especially if no self-registration is allowed (`MAY_REGISTER != []`) then only invited users may auto-register via Shibboleth

Notes:

- use `bsadmin conf_apache` and `bsadmin index_page` for (re-)configuration of the Apache HTTP server and the index page if you have changed `COOKIE_AUTHENTICATION`, `SCRIPTS` or `FEDERATIONS`.
- test if authentication is correctly enforced by accessing `<SERVER_ROOT>/pub/snakeoil/bscw.cgi/`
- `COOKIE_AUTHENTICATION` must be enabled to display the the login-page in the first place.
- this kind of login requires that the authentication process provides an authenticated email address of the user. For `mod_shib` this means that it must set the environment variable `mail` resp. `HTTP_SHIB_INETORGPERSON_MAIL` after successful authentication. To test open: `<SERVER_ROOT>/pub/snakeoil/bscw.cgi/?op=env`

See also section 6.16.3 for further configuration hints.

```
AUTH_MODE = 'Basic'
COOKIE_AUTHENTICATION = None
OPEN_ID_DEFAULT = None
FEDERATIONS = ()
```

`POST_AUTH`

When enabled all `POST` requests must be authenticated by a hidden variable. To avoid cross-side attacks this must be enabled.

```
POST_AUTH = 1
```

`POST_CHECK_REFERER`

For `POST` requests the `Referer` header should start with the server root (of the BSCW web server). If this is not the case and when `POST_CHECK_REFERER` is `True` an error is raised.

Note: some proxies and browser might suppress the `Referer` header (they should not if the `Referer` URL is on the target host, since there is no privacy compromise in that case).

```
POST_AUTHENTICATION = 1
```

`LOG_REMOVED_USERS`

If not empty then all users that are removed from the system are logged in the file `LOG_REMOVED_USERS`. An entry is a line of the form:

```
user_name:user_id[:?]mail_address
```

An email address proceeded by `'?'` denotes an email address that was not owned by the user at the time of removal (hence the user has not received an email notification).

```
LOG_REMOVED_USERS = 'removed_users'
```

`PASSWD`
`PASUPD`
`PASLOG`

`PASSWD` - Web server password file
`PASUPD` - Web server password database update program
`PASLOG` - Web server password database update logfile (only used if `PASUPD` is set)

If `PASSWD` is not empty the password file `PASSWD` is automatically managed by BSCW.

Notes:

- If you change the `PASSWD` file, you must also point your HTTP server to the new file (e.g. run `bsadmin conf_apache` and restart `httpd` with the new configuration).
- Some HTTP servers need a password database. In this case `PASSUPD` must be set to a password update program.

See the sections for the Web server configuration for details (3.3.1 (Unix), 4.4.3 or 4.4.2 (Windows).

```
PASSWD = 'htpasswd'
PASUPD = ''
PASLOG = 'pasupd.log'
```

`SCRIPTS`
`SECURE_SCRIPTS`

Define the CGI scripts that may be called the HTTP server. Given the URL http://bscw.domain.org/testing/bscw/bscw.cgi/25, the HTTP server will split this URL into
the `SCRIPT_NAME` `"/testing/bscw/bscw.cgi"` and
the `PATH_INFO` `"/25"`
The BSCW server splits the script name further into
the `prefix` `"/testing/bscw/"` and
the `script` `"bscw.cgi"`
Note: the prefix always starts and ends with a `'/'`.

The BSCW accepts a `SCRIPT_NAME`, if the prefix is found in the `SCRIPTS` dictionary:

```
prefix->(username, '', create_scripts, extra_scripts)
```

The script is found in one of the two lists `create_scripts` or `extra_scripts`. If username is `None`, the user must authenticate.

Otherwise the BSCW server assumes that the client may use the script without authentication (e.g. for anonymous access or access controlled by the client's host domain). In the latter case the user will get access according to username.

Notes:

- **Important**: You have to apply for a new ("move server") license if you have a granted BSCW license (not an evaluation license) and change the prefix for the entry with username `None` (the standard `SCRIPTS` prefix). See also `SERVER_ROOT` above.
- The usernames must be different in all tuples `(username, ...)` and there must exist at least a tuple with username `None` and a tuple with username “`anonymous`”.
- The command “`bsadmin chkconfig`” needs the dictionary entry (which might be the same for all prefixes) in order to automatically create the scripts listed in `create_scripts`. For special purposes you might also create your own CGI scripts that eventually call the BSCW service. These scripts must be listed in `extra_scripts`.
- The `nj_*` scripts that are used for non-JavaScript access are also created automatically by “`bsadmin chkconfig`“they should **not** occur in `create_scripts` or in `extra_scripts`.
- The user objects for all `usernames != None` are automatically created and registered as anonymous users when the path prefix is accessed. If a non-anonymous user is found by username then the a `"Bad script name"` error will be raised.
- The user objects for all `usernames != None` are automatically created and registered as anonymous users when the path prefix is accessed. If a non-anonymous user is found by `username` then the a `"Bad script name"` error will be raised.
- The `extra_script` and `SECURE_SCRIPTS` feature is intended for experts only. `CREATE_SCRIPTS` is not used elsewhere. It is only defined for convenience.

```
SECURE_SCRIPTS = []
CREATE_SCRIPTS = ['bscw.cgi']
SECURE_PREFIX = '/bscw/'
PUBLIC_PREFIX = '/pub/'
SCRIPTS = {
    SECURE_PREFIX:
        (None,               '', CREATE_SCRIPTS, SECURE_SCRIPTS),
    PUBLIC_PREFIX:
        ('anonymous',        '', CREATE_SCRIPTS, SECURE_SCRIPTS),
#   '/pub/snakeoil/':
#       ('SnakeOilProviders', '', CREATE_SCRIPTS, SECURE_SCRIPTS),
}
```

`SILENT_ERROR_FOR`
`NO_SILENT_ERROR`

If a script prefix is listed in `SILENT_ERROR_FOR`, any error message will be replaced by `SILENT_ERROR.html` which then must reside in the directory `bscw/templates` of the server. For example

```
        SILENT_ERROR_FOR = (PUBLIC_PREFIX,)
```

will prevent error related information from being collected by misuse of public access.

NO_SILENT_ERROR defines a list of error exceptions, where an error will be raised any way, eg.

```
        NO_SILENT_ERROR = ('Account_Locked', 'virus_found')
```

```
SILENT_ERROR_FOR = ()
NO_SILENT_ERROR = ('Account_Locked', 'virus_found')
```

SCRIPTS_OTHER_ROOTS

If you have more than one BSCW server instance on a single host, you must select one of them to handle all document root '/' WebDAV PROPFIND requests for all BSCW server instances on this host. E.g. if you have another BSCW server that handles requests on /bscw1 and /pub1 (see SCRIPTS above), you might set

SCRIPTS_OTHER_ROOTS = ('/bscw1', '/pub1')	on this server and
SCRIPTS_OTHER_ROOTS = None	on the other server.

Note: You have to re-generate the Apache configuration with with "bsadmin conf_apache" when you make changes to SCRIPTS or SCRIPTS_OTHER_ROOTS and restart the web server.

```
SCRIPTS_OTHER_ROOTS = ()
```

SSO_PREFIX
SSO_SCRIPT
CAS_URI
SCRIPTS_ALIASES

To enable Single Sign On support for the Central Authentication System (CAS; http://www.ja-sig.org/products/cas/) an alternate secure path prefix (SSO_PREFIX), an CAS service URL (CAS_URI) and an alternate secure script path (SCRIPTS_ALIASES) must be defined:

SSO_PREFIX defines a path prefix which is redirected to the SSO authentication service. If undefined or empty SSO support is disabled (default).

SSO_SCRIPT (optional) defines an additional alternate script name of the CGI script which is redirected to the SSO authentication service. If undefined or empty CREATE_SCRIPTS[0] is used (default)

CAS_URI defines the URL of the CAS Single Sign On service, e.g.

```
   CAS_URI = 'http://sso.domain.org:8080/cas'
```

SCRIPTS_ALIASES define alternate script alias prefix paths for the secure prefix:

```
   SCRIPTS_ALIASES = {
        SECURE_PREFIX: [
            (SSO_PREFIX, {
               'mode': AUTH_MODE,
               'cookie': ('bscw_cas', None, 120) }), ],
   }
```

An script alias prefix path definition is a list of tuples [('path_alias', auth_dict)]. For every script alias prefix path the authentication dictionary `'auth_dict'` defines the authentication mode and (if needed) cookie authentication, e.g.

```
auth_dict = {'mode':   'Digest', 'cookie': None}

auth_dict = {'mode':   'Basic',
             'cookie': ('bscw_cas', None, 120)}
```

For available '`mode`' values see the `AUTH_MODE` and for available '`cookie`' values see the `COOKIE_AUTHENTICATION` configuration directives above

```
CAS_URI = ''
SSO_PREFIX = ''
SSO_COOKIE = ('bscw_cas', None, 120)
SCRIPTS_ALIASES = {
    SECURE_PREFIX: [
        (SSO_PREFIX, { 'mode': AUTH_MODE, 'cookie': SSO_COOKIE }), ],
}
```

`PATH_INFO_SLASH`

Must have the values `''` or '`%2f'` or `'%2F'`. This should only be set not empty if the Apache HTTP server is used and `"AllowEncodedSlashes On"` is set. It must be equal to the encoding that the Apache HTTP server uses for URI path segments (`'%2f'`).

Warning: This is an experimental feature. Many DAV clients do not work with encoded slashes in URI path segments. Better leave it empty by now.

```
PATH_INFO_SLASH = ''
```

`HTTP_PROXY`
`FTP_PROXY`
`GOPHER_PROXY`

Specify proxies for various server types (i.e. http, ftp, gopher) by defining a variable named `<TYPE>_PROXY`. This variable denotes the proxy server for this type by the form '`<name>:<port>`'. For example

```
HTTP_PROXY   = 'proxy.orbiteam.de:3128'
FTP_PROXY    = 'proxy.orbiteam.de:3128'
GOPHER_PROXY = 'proxy.orbiteam.de:3128'
```

The proxies are used by the BSCW server if it fetches or verifies an URL or if it connects to a web search engine within a www search.

```
HTTP_PROXY = ''
FTP_PROXY = ''
GOPHER_PROXY = ''
```

`BYPASS_PROXY`

Gives a list for domains, where the proxy should be bypassed, i. e., a host whose end of its name matches one of the domains, will be connected directly. Normally it should be set to the local domain. For example

```
BYPASS_PROXY = ['fit.fraunhofer.de', 'orbiteam.de' ]
```

```
BYPASS_PROXY = []
```

`FTP_GATEWAY`

Sets a FTP firewall gateway IP address (or hostname)for the export operation.

```
FTP_GATEWAY = ''
```

`WEBDAV_EDITORS`
`WEBDAV_APP_MODE`

Control the `get` and `edit` actions for documents, which can be opened by applications via WebDAV. For example with MS Office 2003, 2007 or 2010 documents may be saved immediately on the BSCW server.

Prerequisites: WebDAV enabled, MS Internet Explorer 8 or above, MS Office 2003, 2007 or 2010, the document name must have the right extension

`WEBDAV_EDITORS` defines a dictionary MIME-types -> MS-ProgIds which define the editor application to be used on the client. If the MS-ProgId is empty ('') the default editor will be used (this is known to fail sometimes).

`WEBDAV_APP_MODE`

- `0` - disable this feature.
- `1` - action '`get`' with old behavior, action '`edit`' calls the application with a WebDAV link.
- `2` - action '`get`' calls the application with a WebDAV link, no action '`edit`'
- `3` - action '`get`' calls the application with a WebDAV link (in view mode for MS Office), action '`edit`' (in edit mode).

```
WEBDAV_EDITORS = {
  'application/vnd.ms-excel': 'Excel.Sheet',
  'application/vnd.ms-project': 'MSProject.Project',
  'application/vnd.ms-powerpoint': 'PowerPoint.Slide',
  'application/vnd.ms-word': 'Word.Document',
  'application/vnd.visio': 'Visio.Drawing',
  'application/rtf': 'Word.Document',
  'application/vnd.ms-excel.macroenabled.12': 'Excel.Sheet',
  'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet': 'Excel.Sheet',
  'application/vnd.openxmlformats-officedocument.presentationml.presentation': 'PowerPoint.Slide',
  'application/vnd.openxmlformats-officedocument.wordprocessingml.document': 'Word.Document',
}
WEBDAV_APP_MODE = 1
```

`GET_MIME_TYPES_`
`AS_ATTACHMENT`

You may want to configure the system to get/open certain document types (i.e. MIME Types) as attachments and not directly within the browser (which is the default behaviour when users click on a link). Especially for newer versions of Microsoft Office, attachments can avoid strange effects and frustrating user experience: Links to MS-Office documents (Word, Excel, Powerpoint...) will usually let MS-Office open the document directly from the BSCW server and so ask for username / password, even though the document is opened read-only (i.e. for reading or printing only) and the 'save to server' feature is not available. As a workaround, MS suggests to explicitly mark the content as a read-only download (i.e., as an "attachment"). In that case, the browser will download the document with the known login information and afterward open MS-Office on the downloaded file. Also see http://support.microsoft.com/kb/899927/en-us

For all document types configured below, BSCW will add the "`Content-Disposition: Attachment`" header to the response whenever the user clicks a link to the document in a container page.

```
GET_MIME_TYPES_AS_ATTACHMENT = (
  'application/vnd.ms-excel',
  'application/vnd.ms-project',
  'application/vnd.ms-powerpoint',
  'application/vnd.ms-word',
  'application/vnd.visio',
  'application/rtf',
  'application/vnd.ms-excel.macroenabled.12',
  'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet',
  'application/vnd.openxmlformats-officedocument.presentationml.presentation',
  'application/vnd.openxmlformats-officedocument.wordprocessingml.document',
)
```

5.2.5 Section 5: BSCW appearance settings

`USER_SEARCH_LIMIT` Maximum number of matching hits by a User Search.

```
USER_SEARCH_LIMIT = 100
```

`MEMBER_SEARCH` Defines, if the search for BSCW users, is allowed on the add member form.

```
MEMBER_SEARCH= 1
```

`MAX_VERSIONS`

- if `1` (default), new created documents are not set under version control.
- if `0`, new created documents are automatically set under version control and all revised versions will be stored.
- if `<n>`, new created documents are automatically set under version control, but only the latest `<n>` revised versions will be stored. Revising version `<n>+1` will automatically remove the oldest revision.

```
MAX_VERSIONS = 1
```

`SCROLL_LIMIT`
`SCROLL_DEFAULT`

Many browsers crash, when they should display long container listings. To prevent users from requesting such precarious pages, the number of entries in a listing can be limited. If set to `0`, there is no restriction to the number of entries.

`SCROLL_DEFAULT` is taken as default setting for new users. Can be `500`, `300`, `200`, `150`, `100`, `80`, `50` or `20` as long as `<= SCROLL_LIMIT`. Can also be `0` (but only if `SCROLL_LIMIT` is `0`), meaning "all entries".

```
SCROLL_LIMIT = 500
SCROLL_DEFAULT = 100
```

`HELPER`
`EDITOR`

Mime type for BSCW uploading and edit helper. On client side, this Mime type should be configured to one of our BSCW uploader and generic editor programs:

```
EDITOR = 'application/x-bscw-edit'
```

The user must have an application (i.e. script) bound to this mime type and should have selected "external editors" (in the `[Options>Preferences]`

[General] [File Handling] menu). Upon "Edit" action, BSCW serves a file that contains four lines:

- the URL, where the file can be downloaded (GET) or uploaded (PUT)
- the mime type of the file
- the encoding of the file
- the name of the file

EDITOR = 'application/x-zope-edit'

The Zope External Editor is used for editing cycle, i.e. upon "edit" action, BSCW serves a file with according mime type and content (as expected by Zope External Editor). The user must have the Zope External Editor installed and should have selected some "external editors" (in the [Options>Preferences] [General] [File Handling] menu).

```
HELPER = 'application/x-java-jnlp-file'
EDITOR = 'application/x-zope-edit'
```

ConversionMethode

Parameter for file conversion: all possible conversions for a file should be shown (0) or only the best one (1).

```
ConversionMethode = 1
```

STYLE_NOW_BROWSER_SCALE
STYLE_COLORS

STYLE_NOW_BROWSER_SCALE Defines a subdirectory of the directory STYLES_DIR, in which changes to the default style sheets and changes in config_styles.py for different font sizes can be located. It is used only if browser scale is switched off.

STYLE_COLORS defines a list of subdirectories of the directory STYLES_DIR, in which changes to the default style sheets and changes in config_styles.py for different color schemes can be located.

```
STYLE_NOW_BROWSER_SCALE = 'small'
STYLE_COLORS = ()
```

THEMES

Available themes to be chosen by users. Every theme is based on a config_theme.py, which contains values for placeholders. The final .css files are then generated using bsadmin themes.

```
THEMES = ('bscw', 'bw')
```

RATE_COLORS

Colors for URL ratings:

```
[ none, poor, passable, fair, good, excellent ]
```

```
RATE_COLORS = ['#000000','#404878','#6068a0','#7680d0','#ff8000','#ff0000']
```

REFTYPES
DOTDIR

REFTYPES is a list of MIME Types of documents that may contain URLs with relative anchors, especially '../'-references. If a web browser resolves such '../'-references it removes elements at the end of the documents URL path. Hence a sufficient number of (dummy DOTDIR) directories must be inserted into the document's URL.

Note: DOTDIR must contain exactly one '/' and it must be the last character.

```
REFTYPES = [ 'text/html' ]
DOTDIR   = '*/'
```

JAVA_DRAGDROP

Enable/disable internal "Java drag&drop uploader" for all users by default.

Note: In some modern browsers, the drag&drop uploader actually works without using Java, but this setting applies anyway!

```
JAVA_DRAGDROP = True
```

INDEX_PAGE_EXT

specifies a name pattern and works like the 'index.html' feature in most HTTP servers: if a Folder contains an object matching the given pattern ('*' matches '-<language>' or '') then the object is presented to the anonymous user instead of the folder listing. E.g. if INDEX_PAGE_EXT='index*.html' and an anonymous user has selected the German language (de) in his browser, then the Folder contents are looked up for the names 'index-de.html' and 'index.html' (in this sequence).

For compatibility with old BSCW installations, if INDEX_PAGE_EXT starts with '.' (a file extension .<ext> is specified) it will match the names index-<language>.<ext>, index.<ext>, and english.<ext>.

```
INDEX_PAGE_EXT = ''
```

LOCAL_URL_PREFIX

It is possible (for administrators only) to make URL links into the local file system. If LOCAL_URL_PREFIX is not empty and the URL has the form LOCAL_URL_PREFIX:<local file path> then the file or directory on the local file system is accessed by the GET operation on the URL (a relative local file path is interpreted relative to the runtime directory). If the directory contains a file named index.html (recommended!) the contents of index.html are returned instead of a directory listing.

Example: set LOCAL_URL_PREFIX = 'local' and add (as admin) a URL named local:var/log/sys.log to a workspace to provide access to your /opt/bscw/srv/my.bscw.org/var/log/sys.log

Note:

- a leading slash will be interpreted as absolute file path i.e. a URL 'local:/etc/passwd' will result in access to the file '/etc/passwd' on the server
- this feature is experimental, and has obvious security implications! It is disabled by default.

```
LOCAL_URL_PREFIX = ''
```

SYS_MSG

Display system messages. SYS_MSG denotes the number of last system message. If `SYS_MSG > 0`, you must have files

```
   sys_msg1.html, sys_msg2.html ... sys_msg<SYS_MSG>.html
```

in `conf/msg/en/` (and optional the corresponding translated files in other language dependant directories `conf/msg/<lang>/`). All files must be encoded in UTF-8. See `conf/msg/en/sys_msg1.html` as an example.. The files are displayed to users as system messages and must be confirmed.

Note: Newly registered users will see `sys_msg0.html` (if provided).

```
SYS_MSG = 0
```

`SYS_BUSY`

Set, if the server is unavailable for processing requests. The message in file `conf/msg/en/<SYS_BUSY>.html` will be returned. (See `conf/msg/en/sys_busy.html` as an example)

Note: the path and the filename ending `'.html'` is appended to `SYS_BUSY`.

```
SYS_BUSY = ''
```

`SYS_BANNER`

Display system banner. Here urgent messages or announcements can be placed just above the user banner between the navigation and the content area in a folder listing. The string must be some HTML code, e.g.

```
        SYS_BANNER = '<h1>System Banner</h1>'
```

```
SYS_BANNER = ''
```

`INDEX_MSG`

Display a custom welcome message on the index page. Here important messages (e.g. terms of use of the server) can be linked/announced. The string may include HTML code, e.g.

```
   INDEX_MSG = '<h3>Efficient collaboration service.</h3>'
```

Note:

- you need to run `bsadmin index_page` in order to update the index page
- you may also define language dependant welcome messages for DE, FR etc. by defining `INDEX_MSG_DE`, `INDEX_MSG_FR` etc. (`INDEX_MSG` should be default/EN)

```
INDEX_MSG = ''
```

```
SERVER_HOME
SERVER_HELP
SERVER_INFO
SERVER_CANCEL
SERVER_LOGOUT
```

Locations of various resources for the URLs in the BSCW Banner.

- `SERVER_HOME` - BSCW server home page
- `SERVER_HELP` - BSCW Help files (English)
  `SERVER_HELP_DE` - BSCW Help files (German)
  `SERVER_HELP_FR` - BSCW Help files (French)
  by default the help is served from local BSCW server, alternatively you may use the publicly available help:

```
      SERVER_HELP    = 'http://www.bscw.de/help-5.0'
      SERVER_HELP_DE = 'http://www.bscw.de/hilfe-5.0'
```

- SERVER_INFO - BSCW server info page - by default it shows the index page in the scripts directory for anonymous (see SCRIPTS).
- SERVER_CANCEL - Defines an URL, to which will be redirected if cancel is pressed (currently chpwd and rmail). Default is SERVER_INFO.
- SERVER_LOGOUT - Defines an URL, to which will be redirected after logout (currently for COOKIE_AUTHENTICATION only). Default is SERVER_INFO.

```
SERVER_HOME    = '/'
SERVER_HELP    = '' # local help
SERVER_HELP_DE = '' # local help
SERVER_HELP_FR = '' # local help
SERVER_INFO    = ''
SERVER_CANCEL  = ''
SERVER_LOGOUT  = ''
```

SERVER_TIMEZONE

Should be set to the time zone that corresponds to time.localtime and should be given in the form “Continent/City”. If you are not sure, use the special value “localtime”.

```
SERVER_TIMEZONE = 'localtime'
```

BSCW_LICENSE

URL used for requesting BSCW license upgrades. This should not be changed.

```
BSCW_LICENSE = 'https://license.bscw.de/pub/bscw.cgi/'
```

5.2.6 Section 6: Optional BSCW packages

PACKAGES

A list of directories containing BSCW extension packages. List of available packages:

```
'blog',         # Blogs
'case',         # File Synchronisation Tool (Java Applet)
'exportpdf',    # Export views to PDF (requires reportlab)
'factory',      # Document Generator
'FlowFolder',   # Flow Folder
'ldap',         # LDAP interface
'mobile',       # Mobile interface
'poll',         # Opinion polls and schedules
'portal',       # Portal and Widgets
'presence',     # Presence Awareness
'PyLucIndex',   # PyLucene Indexer
'readers',      # Event View
'rss',          # RSS Newsfeeds
'Secure',       # Public key support
'SMS',          # SMS interface
'sync',         # Outlook Synchronisation Tool (Java Applet)
'Tasks',        # Work Flow
'Timeline',     # Graphical overview of activities
```

```
'WebFolder',     # Web Folder
'wsmap'          # Visualisation of workspaces and memberships
```

Attention: The `PACKAGES` list is maintained by the `bsadmin package` command. Do **not** manually edit the `PACKAGES` list.

To enable a package run

```
bsadmin package -e <pkg-name>
bsadmin package -e ldap
```

To disable a package run

```
bsadmin package -d <pkg-name>
bsadmin package -d ldap
```

```
PACKAGES = [
    'blog',             # Blogs
    'case',             # File Synchronisation Tool (Java Applet)
    'FlowFolder',       # Flow Folder
    'mobile',           # Mobile interface
    'poll',             # Opinion polls
    'portal',           # Portal and Widgets
    'rss',              # RSS Newsfeeds
    'sync',             # Outlook Synchronisation Tool (Java Applet)
    'Tasks',            # Work Flow
    'WebFolder',        # Web Folder
]
```

`SERV_UNO_STATE`
`SERV_UNO_TIMES`

`WSREPORT`
`WSREPORT_DIRECT`

`AUTOSUBSCRIBE_REPORT`
`AUTOSUBSCRIBE_REPORT_DIRECT`

`DEFAULT_EVENTMASK`
`DEFAULT_EVENTMASK_DIRECT`

`REPORTLOG`

The user notification services (UNO) perform the following tasks:

- sending periodical workspace activity reports via email to give the users an overview about recent activities in a specific time period (e.g. daily)
- sending direct email notifications to inform the users instantly about recent events (optional)

Using the user notification services a BSCW user does not need to contact its BSCW server(s) so often to check for new events. If the user notification services are activated, the users' event preference page provides a column for subscription of the “Daily Report”, the “Direct Email” notification (depending on the UNO service configuration).

By default no report is sent to new users, so each user may decide to subscribe to the workspace report by himself. The server administrator can change this behavior using the `AUTOSUBSCRIBE_REPORT = 1` flag. If this is enabled new users will automatically be subscribed to the user notification service. (Each user may then unsubscribe from the service.)

Accordingly the direct email notification is disabled by default for each user so each user may decide to enable the direct email. The server administrator can change this behavior using the `AUTOSUBSCRIBE_REPORT_DIRECT = 1` flag. (Again this affects all new users and those users who have not yet changed their event preferences.)

To activate the user notification service the BSCW administrator has to start an additional UNO server (`bscw.adm.bs_servuno`). See section

"SERVERS" for details on how to start bscw.adm.bs_servuno.

Note: UNO accesses the BSCW database server via an extra (virtual) HTTP server running on 'http://localhost:<HTTP_LOCAL_PORT>'. See HTTP_LOCAL_PORT for details.

Also, the following variables have to be set for configuration of bscw.adm.bs_servuno:

SERV_UNO_STATE: A file name for saving the state of the bscw.adm.bs_servuno service must be given here. The file is written, when the bscw.adm.bs_servuno is stopped and read when the server is started again.

SERV_UNO_TIMES: A dictionary containing fine tuning parameters for bscw.adm.bs_servuno; if is set to SERV_UNO_TIMES = None the default settings are used (as shown below). To overwrite the default settings for fine tuning parameters use e.g.:

```
SERV_UNO_TIMES = {
    'TdelayDirect': 10.,
    'MaxRetry': 20,
}
```

The defaults are:

TdelayDirect': 60.: Delay direct notification one minute for the first affected user. This is to accumulate more events in the direct notification mail.

'TdelayNextProc': 3.: Add a delay of 3 seconds for the next affected user. This is to avoid an overload of the mail server if a lot of users are affected,

'TdelayNextDirect': 120.: Delay the next direct notification for the same user two minutes. This is to avoid an overload of the user.

'TdelayDayly': 10.: Add a delay of three seconds between daily notification mails. This is to avoid an overload of the mailserver if the service has to send the notification to a lot of users.

'TdelayRetry': 600.: Add a delay of 10 minutes after the notification has failed and retry then.

'MaxRetry': 3: 3 retries that are delayed with TdelayRetry.

'TdelayFailed': 3600.: After MaxRetry the notification is delayed 1 hour (0 may be assigned here, then there will be no retry upto next midnight).

'FailMessagesAt': 10: Log an error message every 10th failure (first, 11th, 21st ...) **Note**: No error messages are logged after MaxRetry (special values 1: each message 0: never)

'MaxJobs': 5: Maximum number of parallel running mail processes. **Note**: For more throughput on big server machines this value might be increased (20)

'QueueInfo': 10: Show job queue status after 10 jobs are queued

(use values: `2`^{n} `* MaxJobs`)

`'ReportTime': '01:31':` Start daily report at 01:31 (must be >= 00:00 and < 07:00)

`WSREPORT = 1 (= 0)` enable (disable) the daily report.

`WSREPORT_DIRECT = 1 (= 0)` enable (disable) the direct email report.

Note: When `bscw.adm.bs_servuno` does not run the daily report and the direct email report are disabled. Whenever the values of `WSREPORT` or `WSREPORT_DIRECT` are altered `bscw.adm.bs_servuno` must be restarted to take these changes into effect.

`AUTOSUBSCRIBE_REPORT` defines the daily report default subscription for all users

`AUTOSUBSCRIBE_REPORT_DIRECT` defines the direct email report default subscription for all users

Note: once a user has changed its subscription preferences this flag will have no further effect - but the administrator may use the '`bsadmin report'` command to change a user's report subscription later).

`DEFAULT_EVENTMASK` defines the event type subscription mask, with the values

```
read = 1; create = 2; move = 4; change = 8
```

By default all event types are subscribed (`read + create + move + change = 15`)

`DEFAULT_EVENTMASK_DIRECT` defines the default event type subscription mask for the direct email notification. By default no event types are selected, so users can enable the direct email service and then choose to select some event types for certain folders of interest only.

For example, set

```
WSREPORT_DIRECT = 1
AUTOSUBSCRIBE_REPORT_DIRECT = 1
DEFAULT_EVENTMASK_DIRECT = 2
```

to enable the direct email notification service for all users by default, so each user will receive an email for each newly created object! (We do not recommend this setting though.)

`REPORTLOG`: points to a file in which a protocol about the reports is logged. For example:

```
REPORTLOG = 'report.log'
```

```
SERV_UNO_STATE = 'ServUnoState'
SERV_UNO_TIMES = None

WSREPORT = 1
WSREPORT_DIRECT = 0

AUTOSUBSCRIBE_REPORT = 0
AUTOSUBSCRIBE_REPORT_DIRECT = 0
```

```
DEFAULT_EVENTMASK = 15
DEFAULT_EVENTMASK_DIRECT = 0

REPORTLOG = ''
```

ALWAYS_CREATE_
READEVENTS
ACCUMULATE_CHANGE_
EVENTS

BSCW Awareness Service configuration

Setting ALWAYS_CREATE_READEVENTS > 0 enables the creation of read events, even if a user has already read the document and the document was not modified in between. This is sometimes needed for enhanced awareness.

Setting ALWAYS_CREATE_READEVENTS = 0 creates read events, after the first time a user reads a (unchanged) document. For successive reads no further read event is created (default).

Setting ALWAYS_CREATE_READEVENTS < 0 suppresses generation of read events.

If ACCUMULATE_CHANGE_EVENTS is enabled, two subsequent and similar modifications of the same user are are accumulated to one event (i.e. only the last event is kept), unless the object has been read or modified in between.

```
ALWAYS_CREATE_READEVENT = 0
ACCUMULATE_CHANGE_EVENTS = 1
```

AW_MONITOR_ENABLED

Monitor Applet configuration

Setting AW_MONITOR_ENABLED to 1 enables the Monitor Applet, which provides awareness of on-line users and recent events. The Applet will be accessible on the home folder of each user.

Note: The Monitor Applet requires the BSCW web services API (XML-RPC). Setting AW_MONITOR_ENABLED to 1 **mandatory** requires to enable ACCEPT_WEBSERVICES (see below).

```
AW_MONITOR_ENABLED = 0
```

JAVA_APPLETS

Java Applet configuration

JAVA_APPLETS is a list of (available) Java Applets related JAR-files are stored in resources/static/java version details should only be updated in case of an applet update (i.e. new JAR file installed)

```
JAVA_APPLETS = {
    'uploader': ('de.bscw.uploader.UploaderApplet', '5.0.6.0'),
    'dropper' : ('de.bscw.dropper.DropApplet',      '5.0.6.0'),
    'monitor' : ('de.bscw.monitor.MonitorApplet',   '5.0.5.0'),
    'client'  : ('de.bscw.client.*',                '5.0.5.0'),
    'util'    : ('de.bscw.util.*',                  '5.0.5.0')
}
```

SERVERS

The SERVERS list is used for starting (and stopping) BSCW servers. Only

extra server addresses and implementation modules should be specified here (`GSERV`, `SERV_ALARM`, `SERV_ACCESS` are defined below). The extra servers are specified by a pair `(address, module)` or a triple `(address, service_module, protocol_module)`. The default protocol_module is '`bscw.core.cl_servublk`'. The following RPC protocol modules are available:

`bscw.core.cl_servublk`

- standard rpc module, can be used with either a file path (recommended) or a (host, port) TCP/IP address. In the first case the module uses a unix socket if possible or selects a free port for a local IP connection and stores it in the given file.

`bscw.core.cl_servinet_ext`

- for non Python services, (host, port) addresses only.

Example for starting the user notification server:

```
SERVERS = [
  ('UnoSocket','bscw.adm.bs_servuno'),
]
```

```
SERVERS = []
```

5.2.7 Section 7: BSCW database server settings

```
STORE
STORE_PAIR
TABLES
CLEAN
SAVE
```

Persistent object store and garbage collection. Relative paths are relative to the BSCW database directory (`<bscw-runtime-path>/var/data`)

`STORE`

Normally contains actual sizes of files in `STORE_PAIR` and a garbage collection (GC) counter. **Only** for disaster recovery, put a backup into this file and start the server.

`STORE_PAIR`

The BSCW database server uses one of the files in this pair as the current StoreFile. The other one is free for garbage collection (called the GcFile). These files contain data of all persistent objects and will grow **big**. The database server only appends data to the the current StoreFile. So it must be garbage collected from time to time (every day is recommended!). The garbage collector copies actual data from StoreFile to GcFile and then swaps the files.

`TABLES`

BSCW database table information used for fast database server restart.

`CLEAN`

Prefix for some temporary files during garbage collection. The database server moves `STORE` to `CLEAN` during initialization.

`SAVE`

For the purpose of backup, the garbage collector copies the new current StoreFile to this file just before it finishes.

Note: `bsadmin getconfig STORE` shows the current StoreFile from `STORE_PAIR`

It is strongly recommended that `STORE`, `TABLES` and `CLEAN` are in the same directory and the (base) file names are not changed. For increased reliability, put `SAVE` and the files in `STORE_PAIR` on different devices. **Never** put `STORE`, `TABLES`, `CLEAN` and the files in `STORE_PAIR` on a shared file system.

On Unix systems, `STORE_PAIR` may also point to (properly configured) block devices e.g. `STORE_PAIR = ('/dev/sdb3', '/dev/sdb4')`.

```
STORE = 'Store'
STORE_PAIR = ('StoreA', 'StoreB')
TABLES = 'Tables'
CLEAN  = 'Garbage'
SAVE   = 'Backup'
```

`ALARM_DIR`
`FILES`
`TEMP`

Persistent alarm store, file store and temporary files. Relaive pathes are relative to the BSCW database directory (`<bscw-runtime-path>/var/data`)

`ALARM_DIR`

Directory for scheduled alarms

`FILES`

Root directory of document file tree

`TEMP`

Directory for temporary files

We recommend that the directories `TEMP` and `FILES` are **on the same file system**. In this case only a link (instead of a copy) is necessary to put a temporary file in the right place, e.g. after document upload.

Note: You can find the file for a BSCW Document with id `12345` at `FILES/01/23/45F`, probably with some extension `.xxx` depending on the file type.

```
ALARM_DIR = 'Alarm'
FILES = 'Files'
TEMP = 'Temp'
```

`FILES_SWITCH`

Simulates “soft links” in the BSCW file store on Windows 7, 8/Server 2012, 2008, 2003. A list (or tuple) of pairs (path-pattern, substitute) determines the actual location of a BSCW file. E.g. if

```
FILES == 'D:\\files'
```

(see below), then

```
FILES_SWITCH = (('D:\\files\\01', 'E:\\files\\01'))
```

will substitute all BSCW file paths starting with `D:\\files\\01` by file paths starting with `E:\\files\\01`. This may be used for distributing the BSCW file store on different disks etc.

Note: Some `bsadmin` tools like `bsadmin fsck` do not support this feature and may give wrong results.

```
FILES_SWITCH = ()
```

`RMUSER_DIR`

Archive configuration for (optional) archiving of removed user artifacts:

`RMUSR_DIR` - Directory for archives of removed users

`RMUSR_ARC` - Format for removed users archives: `'zip'` (default) or `'tar'`

`RMUSR_ENC` - Pathname encoding: `'utf-8'` (default)

```
RMUSR_DIR = 'rmuserarc
```

`SERV_ACCESS`
`SERV_ACCESS_STATE`

`SERV_ACCESS` specifies the address of the access server `bscw.adm.bs_servaccess`. The `bscw.adm.bs_servaccess` service is an optional accelerator for searches. It implements fast access filtering. Disabled if empty.

In order to enable this service use

```
SERV_ACCESS = 'AccessSocket'
```

Note: If you enable this service on an upgraded Server you might get an error (e.g. in `bscw.log`) like:

```
mm-dd hh:mm:ss  ACCESS watch died:
RuntimeError: Old pickle not supported
```

In this case

```
$ ./bin/bsadmin garbage bs_classtabe0
$ rm var/data/ServAccessState
$ ./bin/bsadmin start
```

will solve the problem. In the case of

```
mm-dd hh:mm:ss  ACCESS position nnnnn:
ValueError: bad marshal data
```

a clean restart without reading the saved state might help:

```
$ rm var/data/ServAccessState
$ ./bin/bsadmin start
```

`SERV_ACCESS_STATE` - File to save state of `bscw.adm.bs_servaccess` when the access server is shut down. This file is only used when `bscw.adm.bs_servaccess` is enabled.

```
SERV_ACCESS = ''
SERV_ACCESS_STATE = 'ServAccessState'
```

`SERVER_LOG`
`BSCW_LOG`
`ALARM_LOG`

All requests to BSCW are logged in this file. Should be set for analyzing purposes only. A log entry contains the following information (divided by blanks):

- request date (local time)
- remote host
- remote user
- request method
- BSCW operation
- response code
- request duration (CPU time)
- request duration (real time)
- request path

Server activities (e.g. start, stop, gc) and errors will logged in `BSCW_LOG`. Non-recoverable `alarm()` failures will be logged in `ALARM_LOG`.

```
SERVER_LOG = 'server.log'
BSCW_LOG = 'bscw.log'
ALARM_LOG = 'alarm.log'
```

`BSCW_UMASK`

`BSCW_UMASK` restricts access to owner and group by default (i.e. mask out read, write and execute bits for "other" users). This mask is used only on UNIX systems.

```
BSCW_UMASK = 7
```

`DBMOD_TAB`
`DBMOD_CACHESIZE`
`DBMOD_PAGESIZE`
`DBMOD_HASH`
`DBMOD_MINKEY`

`DBMOD_TAB`: definition of the BSCW database table type:

- the default value '`dict`' uses Python dictionaries and should be used for small BSCW databases only, because the key and offset tables are hold in memory of the `bs_servdb` process and must be loaded to memory and stored to disk on server start and shutdown respectively.
- the value '`bsddb4`' uses an external Berkeley DB to store the BSCW database tables and requires an installed Berkeley DB (http://sleepycat.com/) and the additional bsddb module. Python >= 2.3 contains the bsddb module by default and supports Berkeley DB versions up to version 4.4.

If you use a newer version of Berkeley DB install the actual version of the bsddb module, see http://pybsddb.sourceforge.net/.

The following configuration parameters `DBMOD_CACHESIZE`, `DBMOD_PAGESIZE`, `DBMOD_HASH`, `DBMOD_MINKEY` are only used with `DBMOD_TAB='bsddb4'`:

`DBMOD_CACHESIZE`: defines the cache size of the Berkeley DB. The recommended cache size is about 10-15% of the actual `STORE` size. Note:

`DBMOD_CACHESIZE <  100` defines cache size in giga bytes (GB)
`DBMOD_CACHESIZE >= 100` define cache size in bytes (B)

`DBMOD_PAGESIZE`: specifies the size of a single cache page. Do not choose too big values to avoid high I/O load (default: `8192`)

DBMOD_HASH: uses Berkley DB HASH access method (instead of the default BTREE access method). Note: this option is **not** recommended!

DBMOD_MINKEY: is a pair of two values which are only used with the BTREE access method (the default). The values depend on the maximum key + data size of the offset table (StoreOff) and the key table (StoreKey) respectively. The values should be less than

```
DBMOD_PAGEZIZE / 2*<max key+data size>
```

A good working heuristic value pair seems to be

```
DBMOD_MINKEY = (9*(DBMOD_PAGEZIZE/1024),
                5*(DBMOD_PAGEZIZE/1024))
```

```
DBMOD_TAB = 'dict'              # Python dictionary (default)
DBMOD_CACHESIZE = 2097152       # cache size in bytes
DBMOD_PAGESIZE = 8192           # page size in bytes
DBMOD_MINKEY = 72, 40           # heuristic values for page size 8192
```

GSSERV
SERV_ALARM

GSERV - address of database server socket DBMOD
SERV_ALARM - address of alarm server socket (bscw.adm.bs_servalarm)

Filenames are recommended here. Fixed local port addresses like ('localhost', 12966) should only be used if there are problems with UNIX sockets or the automatic TCP/IP port selection does not work.

The bscw.adm.bs_servalarm service schedules alarms for persistent objects.

```
GSERV            = 'DbSocket'
SERV_ALARM       = 'AlarmSocket'
```

ACCEPT_WEBSERVICES

BSCW offers a range of services via different web service protocols: XML-RPC, JSON, SOAP

Basically most of the actions available on the user interface (like "add folder") are accessible via a web service API. Of course access to API is restricted via access control as in the regular user interface (i.e. authentication and BSCW internal roles and rights are respected).

For documentation on the web services API see the BSCW distribution

```
bscw-5.0.9-3????-py2?/doc/devel/BSCW50-API-doc.zip
```

or our web site:

```
http://www.bscw.de/english/documentation.html
```

Availability of the web service API on different user levels can be configured by adding the respective flags:

ACCEPT_WEBSERVICES = 0 disable all web service calls

ACCEPT_WEBSERVICES = 1 enable standard web service calls for registered users

ACCEPT_WEBSERVICES = 2 enable additional web service calls for registered administrators

ACCEPT_WEBSERVICES = 4 enable standard web service calls for public access (anonymous)

`ACCEPT_WEBSERVICES = 8` enable standard web service calls for anonymous users with special authentication (see `SCRIPTS`)

Note:

- By default the web service API is enabled for registered users only.
- Certain built-in components of BSCW (like the `Portal`) do require web services for client-server communication and won't work if you disable this feature!
- If disabled, all requests will be rejected by BSCW rendering an error response (e.g. HTTP error code 501: content_unsupported in case of XML-RPC API).

```
ACCEPT_WEBSERVICES = 1
```

`BSCW_LOGGING`

BSCW offers detailed logging information on various components of the system. The purpose of logging is mainly for debugging and problem identification. It should be noted that system logs may contain personal detail and sensitive information.

Logging is enabled by default to log `error` conditions:

```
BSCW_LOGGING = {
    'sys':  ('ERROR', 'sys.log'),
}
```

This creates the log file `sys.log` where records from all loggers with the log level `ERROR` will be logged. To create log files for specific loggers with more detailed log levels add the following entries, for example:

```
BSCW_LOGGING = {
    'sys':  ('INFO', 'sys.log'),
    'ldap': ('DEBUG', 'ldap.log'),
    'mda': ('DEBUG', 'mda.log')
}
```

Where `'ldap'` specifies the logger for the BSCW ldap package, and `'mda'` the logger for the BSCW Mail Delivery Agent.

```
#BSCW_LOGGING = {
#    'sys': ('INFO', 'sys.log'),
#    'arc': ('ERROR', 'arc.log'),
#}
```

`WAIT_ARCHIVING`
`ARCHIVE_LIMIT`

BSCW allows archiving of large workspaces by starting a background process for the archive task (and likewise for extraction of archives). For smaller archives the task is coupled with the CGI process so that the user will see immediate feedback.

`WAIT_ARCHIVING` defines the time in seconds that a CGI process will wait for the archive or extract task to complete before it returns with an adhoc response to the user. The time must be less than the http server's timeout

`WAIT_ARCHIVING = 10`
`ARCHIVE_LIMI = '2G'`
	(typically 300 sec.).
	`ARCHIVE_LIMIT` defines the maximum size of downloadable archives. You may use this to prevent denial-of-service attacks caused by users creating archive-bombs. By default `ARCHIVE_LIMIT` is set to `2G`. Possible values for the sizes are strings which may be specified in bytes or kilo- (mega-, giga-, tera-) bytes with an additional `k` (`K`), `M`, `G` or `T` suffix. E.g. valid values for ten mega-bytes are `10485760` or `'10M'`.

`packages_state`
`packages_state = 2`
	Please do not change `packages_state`. It controls automatic enabling/disabling of new/obsolete `PACKAGES` in `bsadmin update_defaults`

5.3 conf/config_cal.py

This is the configuration file for the calendar. Please note that not all entries are meant to be configured by the administrator here. Especially the settings of `flags`, `categories` and `appoint_status` should **not be changed**.

BSCW administrators may change the default preferences for each user's calendar here – the `calendar_flags` contains the sum of all enabled calendar flags (cf. list of flags). The file also contains defaults for the display of appointments in different views. For each view (`y = year, m = month, w = week, d = day`) a list of potential (`'allowed_x'`) and displayed (`'view_x'`) style items is specified.

5.4 conf/config_clientmap.py

The `config_clientmap.py` configuration file defines the mapping between web browsers and their supported options. See the comments in this file for further descriptions.

5.5 conf/config_controls.py

The `config_controls.py` configuration file defines access right independent parameters for BSCW operations. Generally it is not advised to make changes without consulting our support staff.

5.6 conf/config_convert.py

The conversion tool configuration is automatically performed by the "`bsadmin update_defaults`" script. This script will search the local system for archiver, encoder or converter commands and generate a `<bscw-runtime-path>/conf/config_convert.py` converter configuration file. To locate a converter command the script uses some internal heuristics and evaluates the users' environment search path variable (`PATH` (Unix) resp. `Path` (Windows)).

In the following paragraph the syntax of the converter configuration file is explained. The system commands for archiver, encoder or converter tools are given in the following three lists respectively:

1. The Archivers list contains triples `(type, create, extract)` with

`type`	the mime-type for the archiving tool
`create`	the shell command to create an archive
`extract`	the shell command to extract files from an archive

   Example:

   ```
   Archivers = [
     ('application/x-tar',
       '/usr/bin/7za a -ttar %(dest)s -r %(src)s [D_EXT=.tar]',
       '/usr/bin/7za x -ttar %(src)s',
     ),
     ('application/zip',
       '/usr/bin/7za a -tzip %(dest)s -r %(src)s [D_EXT=.zip]',
       '/usr/bin/7za x -tzip %(src)s',
     ),
     ('message/rfc822',
       'internal',
       'internal',
     ),
   ]
   ```

2. The Encoders list contains triples `(type, encoder, decoder)` with

`type`	the encoding-type for the encoding tool

`encoder` the shell command to encode a file

`decoder` the shell command to decode a files

Example:

```
Encoders = [
  ('compress',
    '/usr/bin/compress -f -c %(src)s > %(dest)s',
    '/usr/bin/uncompress -c < %(src)s > %(dest)s',
  ),
  ('gzip',
    '/usr/bin/gzip < %(src)s > %(dest)s',
    '/usr/bin/gzip -d < %(src)s > %(dest)s',
  ),
  ('x-bzip2',
    '/usr/bin/bzip2 < %(src)s > %(dest)s',
    '/usr/bin/bzip2 -d <  %(src)s > %(dest)s',
  ),
  ('x-uuencode',
    '/usr/bin/uuencode %(src)s dummy > %(dest)s',
    '/usr/bin/uudecode -p %(src)s > %(dest)s',
  ),
]
```

3. The Converters list contains 5-tuples `(src_type, dest_type, quality_factor, command, info)` with

`src_type` the mime-type from the source file

`dest_type` the mime-type from the destination file

`quality_factor` a number between 0 and 1 to estimate the quality of the conversion. If you have more than one tool for the same conversion, the one with the best quality is chosen.

`command` the shell command to convert a file

`info` information about what is lost during the conversion

Example:

```
Converters = [
  ('application/pdf', 'text/plain', '1.0',
    '/usr/bin/pdftotext -enc UTF-8 %(src)s %(dest)s',
    'layout/images',
  ),
  ('application/postscript', 'text/plain', '1.0',
    '/usr/bin/ps2ascii -sOutputFile=%(dest)s -q -dBATCH %(src)s',
    'layout/images',
  ),
  #...
]
```

The shell commands have to be specified with an absolute pathname and are normally executed in a temporary directory in BSCW Temp. In a shell command the following patterns can be used:

`%(src)s` the absolute path of the source file

`%(dest)s` the base name of destination file

In squared bracket some additional parameters can be set:

`[S_EXT=.xxx]` specifies the extension of the source file

`[D_EXT=.xxx]` specifies the extension of the destination file

`[D_NAME=%(dest)s.xxx]` or

`[D_NAME=%(src)s.xxx]` specifies the name of the destination file

`[E_DIR=xxx]` specifies a directory, where the tool should be executed

Multiple parameters can be separated in the squared bracket with a semicolon.

To regenerate the converter file, e.g. after you installed new converters or adopted your environment search path, run the script with the options `-s` (to skip a Python import check) and `-v` (to print some information about found converter commands):

```
$ bin/bsadmin update_defaults -h
usage:
 bsadmin update_defaults [-s] [-v|...|-vvv] [-w|...|-www]
 -s              skip import check
 -v ... -vvv     verbosity
 -w ... -www     warning level

$ bin/bsadmin update_defaults -s -v
...
Converter auto-configuration:
 Found Commands:
  '7z': '/usr/bin/7za'
  'antiword': '/usr/bin/antiword'
  'bzip2': '/usr/bin/bzip2'
  'cjpeg': '/usr/bin/cjpeg'
  'compress': '/usr/bin/compress'
  'djpeg': '/usr/bin/djpeg'
  'extract': '/usr/bin/extract++'
  'gif2tiff': '/usr/bin/gif2tiff'
  'gzip': '/usr/bin/gzip'
  'lynx': '/usr/bin/lynx'
  'pdftotext': '/usr/bin/pdftotext'
  'perl': '/usr/bin/perl'
  'ps2ascii': '/usr/bin/ps2ascii'
  'tar': '/bin/tar'
  'tiff2ps': '/usr/bin/tiff2ps'
  'uncompress': '/usr/bin/uncompress'
  'unzip': '/usr/bin/unzip'
  'uudecode': '/usr/bin/uudecode'
  'uuencode': '/usr/bin/uuencode'
  'w3m': '/usr/bin/w3m'
  'zip': '/usr/bin/zip'
conf/config_convert.py : updated...
```

5.7 conf/config_countries.py

The `config_countries.py` file defines country codes (based in ISO 3166 standard) for selections lists.

5.8 conf/config_ext.py

The `config_ext.py` file specifies Python modules that are be overloaded in the BSCW runtime directory. This mechanism allows to replace or extend Python code in the BSCW distribution library in each instance directory.

5.9 conf/config_help.py

The `config_help.py` file defines mappings from the BSCW context sensitive help to online help HTML pages.

5.10 conf/config_html_ui.py

The user profile definition determines which actions will appear at the user interface. All actions will subsequently be filtered for access control and feasibility at run time.

5.10.1 User Profiles

This configuration file allows a BSCW system administrator to define so-called user profiles. A user profile defines those actions which a user who has selected the respective profile sees in the BSCW interface. User profiles may be defined according to the expertise of particular user groups (e.g., beginners, advanced users or experts) or according to other classification schemes that may be appropriate for a particular BSCW server installation.

The default configuration file `config_html_ui.py` as delivered with the BSCW server software comes with three pre-defined profiles called `BEGINNER`, `ADVANCED` and `EXPERT` (see below). These profiles are upward compatible, i.e., the `BEGINNER` profile is a subset of the `ADVANCED` profile which in turn is a subset of the `EXPERT` profile. These profiles should only be considered examples: A BSCW administrator may modify the actions, buttons, etc., which shall appear in the interface of the respective profile to the requirements of his or her specific user community.

The profiles below apply unless the end user explicitly selects an action to appear at the user interface (action [Options > Preferences]). Profiles should have values 2**n to facilitate mask evaluations (see below). By default, the following bit masks for user profiles are provided:

```
ui_no           =   0   # action will not appear for any user
ui_beginner     =   1   # action will appear, if user has chosen beginner level
ui_advanced     =   2   # advanced level
ui_expert       =   4   # expert level

ui_waste        = 128   # action will be allowed in the waste
ui_yes          = 255   # action will always appear at ui
```

The variable `ui_profiles` defines the list of available u/i profiles that the user may choose from. The second value is used to translate the profile's name and must correspond to a variable in `<bscw-runtime-path>/messages/en/lg_msgconfig.py`.

```
ui_profiles = [
     (ui_beginner,  'ui_beginner'),
     (ui_advanced,  'ui_advanced'),
     (ui_expert,    'ui_expert'),
]
```

The default profile for new users (when registering) is defined as follows:

```
default_profile_new = ui_expert
```

The default profile for existing users (this applies when upgrading from versions lower than 3.3 of the BSCW server) is defined as follows:

```
default_profile = ui_expert
```

In the lists below, actions in the user interface may be set to one of the predefined action profiles, e.g. to ADVANCED. This value will be filtered by the user's individual user interface profile, e.g., ui_beginner.

Action profiles may be combined to allow for more flexibility, e.g., an action profile == ui_advanced | ui_expert means that both users with profile ui_advanced and users with profile ui_expert will have this action at their interface. Other combinations are possible.

```
BEGINNER = ui_beginner | ui_advanced | ui_expert  # ascending profiles
ADVANCED = ui_advanced | ui_expert                # ascending profiles
EXPERT   = ui_expert

WASTE    = ui_waste
ADMIN    = ui_yes
```

What follows in the configuration file is the list user_acts which specify the actions in the user interface according the profiles. The list MultiActions specify the actions which can be used on a selection of BSCW objects.

5.10.2 Columns

It is possible to configure the folder views for every folder type by altering the corresponding entries in config_html_ui.py. Additionally single columns can be hidden in specific users levels. To achieve this the following entries of the folder type list must be changed:

```
V_ANY = -1
V_ADM = V_ALL | V_DEF | V_MIN
V_AD  = V_ALL | V_DEF
V_A   = V_ALL

UI_VIEWS = {
  'Folder': {
    #...
    'columns': [
      (col_info,            V_ADM,          ui_yes, ),
      #...
      (col_moddate,         V_AD,           ui_yes, ),
      (col_events,          V_AD,           ui_yes, ),
    ]
  }
}
```

An entry has the following meaning (illustrated for he entry “Last Modified”):

```
      (col_moddate,         V_AD,           ui_yes, ),
```

The “Last Modified” column is displayed within the predefined views “All” (V_ALL) and “Default” (V_DEF) and available in all user levels (ui_yes). To display this column additionally within the view “Minimal” (V_MIN) the entry has to be changed as follows:

```
      (col_moddate,         V_ADM,          ui_yes, ),
```

Similar proceed for all other folder types (e.g. TaskList).

Columns are displayed in the order of the definition in the folder type list. Altering the order of the folder type list (in config_html_ui.py) leads to a redefinition of the order of the shown columns order at the user interface.

Note: Not every column order results in a usable display in all web browsers. Some columns (such as col_name) are mandatory. The columns col_info, col_check, col_icon, col_name and col_actions should not be altered.

5.11 conf/config_icons.py

The config_icons.py file maps BSCW objects to image file.

5.12 conf/config_meet.py

This is the configuration file for synchronous collaboration tools, on-line directories and messaging services. It contains the counter ID for a unique number and the lists Applications, OnlineDirs and MessagingServices.

An entry in the Applications list for a synchronous collaboration tool is of a 3-tuple

```
(<name>, <id>, <params>)
```

where <name> is the name of the tool, <id> a unique number and <params> a 2-tuple (<mime_type>, <call_cmd>). Here <mime_type> gives the Mime Type to start the application and <call_cmd> is a command string to call a user. In the command string following substitutes can be used: %(host)s IP address for the host of the participant, %(name)s the name of the participant, %(email)s the mail address of the participant.

An entry in the OnlineDirs list consists of a 5-tuple

```
(<name>, <home>, <view-address>, <icon>, <id>)
```

<name> is the name of the directory service. <home> and <view-address> are URLs with a link to the directory. <view-address> has to be directly connectible with an email address, i.e. it usually has to end with a '/'. <icon> gives a reference a icon in config_icon.py; <id> is a unique number.

An entry in the MessagingServices list consists of a 4-tuple

```
(<name>, <home>, <contact-link>, <id>)
```

<name>, <home> and <id> are defined as above, while <contact-link> may include %(uid)s which is replaced with the users' UID for the given service.

5.13 conf/config_menu.py

The config_menu.py file speicfies the BSCW menu configuration.

5.14 conf/config_metadata.py

The config_metadata.py file specifies the meta data for BSCW objects.

5.15 conf/config_mimegroups.py

The config_mimegroups.py file maps MIME types of different applications in groups, eg. Microsoft Office.

5.16 conf/config_mimeicons.py

This is (an excerpt from) the configuration file for icons. For further description see <bscw-runtime-path>/conf/config_icons.py.

```
access          =              ('msaccess.gif', (21, 21),    0)
aiff            =              ('audio.gif',    (21, 21),    0)
...
zip             =              ('zip.gif',      (21, 21),    0)
```

5.17 conf/config_mimemsg.py

While the translations in different languages of all built-in MIME-type messages are stored in `lg_msgconfig.py`, the file `config_mimemsg.py` keeps all user-defined MIME-type messages defined by an BSCW administrator.

5.18 conf/config_mime.py

This is the configuration file for the MIME-types. Default MIME-type information for BSCW details can be extended or modified directly for system-wide effect. To add MIME-types, add an entry to the list below. Also consider adding an entry to `bs_iconconfig.py` if the type should have its own icon (otherwise the 'Unknown' icon will be used) and adding an entry to `config_mimemsg.py` for the description of the MIME-type. The format for entries is:

```
name = (MIME-Type, suffix)
```

where

- `name` is the name of type's icon (in `config_icon.py`) and description (in `config_mimemsg.py`). The name must be unique and conform to Python naming conventions;
- `MIME-Type` consists of a type and a subtype divided by a slash (use lower case letters);
- `suffix` is used by document conversion assistant and to determine the document type by extracting a file archive (<= 3 characters recommended).

Note: Only a subset of the IANA (and common, non-standard) media-types are specified here. For more information on MIME-types see:

```
http://www.iana.org/assignments/media-types/
```

Examples of entries in the list are:

```
access          =  ('application/vnd.ms-access',      'mdb')
aiff            =  ('audio/x-aiff',                   'aif', 'aiff')
...
zip             =  ('application/zip',                'zip')
```

5.19 conf/config_mobile_ui.py

The `config_mobile_ui.py` file specifies the user interface setting for the BSCW mobile interface.

5.20 conf/config_mpick.py

The `config_mpick.py` file allows to overload non-existing BSCW database object classes with replacement classes at runtime. BSCW uses this mechanism to replace database objects for BSCW extension package classes which have been disabled.

5.21 msg/<lang>/lg_msgconfig.py

The `lg_msgconfig.py` file specifies numerous text strings which are used in the interface of the BSCW server. Since these text strings are obviously language dependent, they are stored in the respective language directories, i.e., there exists a file `lg_msgconfig.py` in `msg/en` as well as in `msg/de` and possibly in other language directories.

While the default language files are located in the distribution library directory `<bscw-path>/lib/bscw-5.0.9-3????-py26/bscw/msg/*` changes should be located in the corresponding BSCW instance runtime directory `<bscw-runtime-path>/bsext/msg/*`.

For example to change translations from `msg/en/lg_msgconfig.py` for your instance create a `<bscw-runtime-path>/bsext/msg/en` directory and copy the distribution `<bscw-path>/lib/bscw-5.0.9-3????-py26/bscw/msg/en/lg_msgconfig.py` file to this location. Now you can edit `<bscw-runtime-path>/bsext/msg/en/lg_msgconfig.py` and adopt it for your BCSW instance.

The file consists of a set of entries where each entry has the form

```
InternalName        = 'Interface text string'
```

where `InternalName` is the name of an entity in the BSCW server source code and `Interface text string` is the external representation of the entity in the user interface. Obviously, `InternalName` is the same character string for all languages whereas `Interface text string` is, in general, specific for each language. For example, the `lg_msgconfig.py` file for English contains the following two lines

```
Folder              = 'Folder'
ChangeEvent         = 'changed'
```

whereas the `lg_msgconfig.py` file for German contains

```
Folder              = 'Ordner'
ChangeEvent         = 'geändert'
```

Note the internal name and its external representation may be the same as for “Folder” (usually only in English) or it may be different as for “ChangeEvent” and “changed” (and, in general, for other languages than English).

A BSCW system administrator may modify the user interface of his or her BSCW server by replacing interface text strings, e.g., if the `ChangeEvent entry` is modified to

```
ChangeEvent         = 'modified'
```

the change event would appear with the name “modified” in the user interface.

Whereas `InternalName` must always be one word conforming to Python naming conventions, `Interface text string` may consists of several words and may include HTML mark-up and also parameters for variable parts and must therefore be enclosed in quotes (single `'`, double `"` or triple `"""`). For example:

```
ChAccessEvent       = 'access rights changed'
no_objects          = '<P><STRONG>No objects, currently.</STRONG></P>'
CreateEventMsg      = 'created by %(name)s, %(date)s'
```

The entries are not listed in detail here.

The `lg_msgconfig.py` files for English is the “default” file, i.e., if a `lg_msgconfig.py` file for a language different from English is lacking a translation, the entry is taken from the English file. In order to facilitate comparison of the `lg_msgconfig.py` files for different languages, the order of the entries in the files should not be modified, nor should entries be removed completely. Entries which should not or cannot be translated, should be commented out for performance reasons. Commenting out entries from a `lg_msgconfig.py` file for languages other than English may be sensible, e.g., if a translation is not desired which is normally the case for system messages.

6 BSCW Packages

This section contains instructions on how to configure the additional packages provided for the BSCW shared workspace system. Each package has to be enabled or disabled using the `bsadmin package` command, which creates the corresponding BSCW configuration directory (e.g. `<bscw-runtime-path>/conf/<package>/`) with the necessary package configuration files and changes the `PACKAGES` list in the `<bscw-runtime-path>/conf/config.py` file.

Generally all BSCW packages are maintained by the `bsadmin package` command line script for

1. management of *distributed* BSCW packages (as described in the sections below)
   - to enable a distributed BSCW package run
     ```
     bin/bsadmin package -e <pkg-name>
     bin/bsadmin package -e ldap
     ```
   - and to disable distributed a BSCW package run
     ```
     bin/bsadmin package -d <pkg-name>
     bin/bsadmin package -d ldap
     ```
2. management of *external* BSCW packages (e.g. customer developments). An external BSCW package is usually provided as a ZIP archive and enabled as follows
   - to extract an external BSCW package run
     ```
     bin/bsadmin package -i <pkg-name>
     bin/bsadmin package -i fhg_fit.zip
     ```
   - to enable an external BSCW package run
     ```
     bin/bsadmin package -e <pkg-name>
     bin/bsadmin package -e fhg_fit
     ```
   - and to disable an external BSCW package run
     ```
     bin/bsadmin package -d <pkg-name>
     bin/bsadmin package -d fhg_fit
     ```

Finally the command `bsadmin package -l` provides an overview about enabled/disabled BSCW packages

Depending on the particular BSCW package further configuration has to be done either in the BSCW instance configuration file (`<bscw-runtime-path>/conf/config.py`) or within the BSCW package configuration files (located in `<bscw-runtime-path>/conf/<package>/`). Please refer the following description for each BSCW package.

6.1 Content Search *PyLucIndex*

Preferably BSCW uses a full text search for BSCW meta data and document contents based on the Lucene Java indexing and search framework. The provided `PyLucIndex` package is the preferred way to enable search for Windows and Unix based BSCW instances.

The package `PyLucIndex` uses PyLucene, a “JCC” compiled python extension for Lucene Java. You need to download and install PyLucene before you activate his package.

PyLucene is maintained under the Apache Lucene project at the Apache Software Foundation. For more information on PyLucene, please visit `http://lucene.apache.org/pylucene/`.

A source distribution can be downloaded from

```
http://www.apache.org/dyn/closer.cgi/lucene/pylucene/
```

BSCW 5.0 supports PyLucene 3.0.1 to 3.6.2

Important:

- due to API incompatibility previously supported versions of PyLucene 2.3 to 2.9 are **not** supported any longer. When upgrading from a pre-BSCW 5 version the replacement/installation of PyLucene 3.0.1 to 3.6.1 is required.
- A PyLucene version for the Python version your BSCW instance is installed with is required
- Additionally PyLucene requires an installed Java Runtime Environment 1.6 (or 1.5)

We gratefully acknowledge the work of the Lucene group (especially Doug Cutting) and the PyLucene group (especially Andi Vajda) who did an excellent job in making Lucene available to the Python developers.

Note Windows platforms (32-bit): for your convenience we provide pre-built binaries of PyLucene (with Java 1.6) as mirror download on the BSCW website at

- Python 2.6
  `http://www.bscw.de/files/Download/Software/PyLucene/lucene-3.6.1-py2.6-win32.egg`
- Python `egg` archives are handled with Setuptools and EasyInstall, e.g.
  `easy_install lucene-3.6.1-py2.6-win32.egg`
  for details refer
  `http://pypi.python.org/pypi/setuptools#using-setuptools-and-easyinstall`
- Ensure Java JRE 1.6 is installed and the following directories are appended to your `Path` environment variable:

```
C:\Python26;C:\Python26\Scripts;C:\Program Files\Java\jre6\bin;
     C:\Program Files\Java\jre6\bin\client;
```

Configuration

The main configuration required is for content search, i.e. indexing document contents. You will need to define converters for all document types that should be indexed. BSCW already provides a framework for document conversion which is used by this indexing package.

Please install needed converter programs and adopt the paths of the converter programs to reflect your local installation. There are a number of open source and also commercial conversion tools available. For example, we suggest using the `xpdf` tools for PDF conversion (available for different platforms from http://www.foolabs.com/xpdf/). For MS-Word document conversion you may want to use `antiword` (available for different platforms from http://www.winfield.demon.nl/). The SWISH++ Unix-based file indexing and searching engine also provides a separate text-extraction utility (`extract`) which supports indexing of non-text files. It is essentially a more sophisticated version of the Unix `strings` command - see http://swishplusplus.sourceforge.net/.

After the installation of PyLucene enable the BSCW *PyLucIndex* package with

```
bin/bsadmin package -e PyLucIndex
```

If you installed additional converter programs update the configuration by using

```
bin/bsadmin update_defaults -s -v
```

(as described in section 5.6) to update the `<bscw-runtime-path>/conf/config_convert.py` converter file.

Furthermore the index configuration allows some fine tuning of the PyLucene indexer:

- `FILES_TXT`
  Directory to store text file representation

- `INDEX_DIR`
  Directory to store the index files

- `INDEX_LOG`
  Log file for indexing process (set None for no logging)

- `INDEX_USE_BSDDB`
  Optionally use Berkeley DB library (bsddb) for storage of lastmod

- `CREATE_INDEX_ARGS`
  Arguments for automatic restart of "`bin/bsadmin create_index`"

- `INDEX_VM_MAXHEAP`
  Max heap for Java VM (lucene only)

- `INDEX_QUERY_HELP`
  link to the query syntax documentation
  Note: this actually depends on the installed version of PyLucene! (see below for possible changes in BSCW: `INDEX_QUERY_OPERATOR_AND`)

- `INDEX_QUERY_OPERATOR_AND`
  default query operator: in PyLucene, the `OR` operator is the default conjunction operator. i.e. a search for "brown sugar" yields all documents that contain any of the words "brown" OR "sugar" - to use this query type set:
  ```
  INDEX_QUERY_OPERATOR_AND = False
  ```
  in BSCW we change the default query operator to AND: that way the "Search in Documents" behaves like a search in Google

- `INDEX_QUERY_LEADING_WILDCARD`
  allow leading wildcards (e.g. *ook)
  Note: in PyLucene leading wildcards are not supported by the QueryParser by default. However they can be enabled. Note that this can be an expensive operation: it requires scanning the list of tokens in the index in its entirety to look for those that match the pattern.

- `INDEX_OBJECT_MAXLOAD`
  number of objects to load from DB while indexing (chunk size)

- `INDEX_OBJECT_MAXBUF`
  size of internal object buffer (for incremental index update)

- `INDEX_QUERY_MAXHITS`
  number of hits to return in one query to indexer during search

The following directives allow fine tuning of Lucene indexer: (see `http://lucene.apache.org/` for details)

- `INDEX_RAM_BUFFER`
  Buffer Size in MB (default: 16 MB)
  For the added documents, flushing is now triggered either by RAM usage of the documents or the number of added documents. Lucene developers recommend for faster indexing performance to flush by RAM usage instead of document count and use as large a RAM buffer as you can.
  Note:
  - setting `INDEX_RAM_BUFFER` to a negative value will set `DISABLE_AUTO_FLUSH` which prevents triggering a flush due to RAM usage (and uses document count instead - see *MaxBufferedDocs* below)
  - if flushing by document count is also enabled (via *MaxBufferedDocs*), then the flush will be triggered by whichever comes first.

- `INDEX_MERGE_FACTOR`
  MergeFactor - must never be less than 2. The default value is 10.
  Determines how often segment indices are merged by `addDocument()`. With smaller values, less RAM is used while indexing, and searches on unoptimized indices are faster, but indexing speed is slower. With larger values, more RAM is used during indexing, and while searches on unoptimized indices are slower, indexing is faster. Thus larger values (> 10) are best for batch index creation,and smaller values (< 10) for indices that are interactively maintained.

- `INDEX_MAX_BUFFEREDDOCS`
  MaxBufferedDocs - must never be less than 2. The default value is 10.
  Determines the minimal number of documents required before the buffered in-memory documents are merged and a new Segment is created. Since Documents are merged in a *RAMDirectory*, large value gives faster indexing. At the same time, *mergeFactor* limits the number of files open in a *FSDirectory.*

- `INDEX_MAX_MERGEDOCS`
  MaxMergeDocs - default value is *Integer.MAX_VALUE*.
  Determines the largest number of documents ever merged by *addDocument()*. Small values (e.g., less than 10,000) are best for interactive indexing, as this limits the length of pauses while indexing to a few seconds. Larger values are best for batched indexing and speedier searches.

- `INDEX_MAX_FIELD_LENGTH`
  MaxFieldLength - limits number of terms to store per field
  By default Lucene stores first 10.000 terms ("words") this may restrict search results on document content (especially for longer documents)
  Note: `INDEX_MAX_FIELD_LENGTH = None` will allow unlimited number of terms per field

- `INDEX_MAX_CLAUSE_COUNT`
  MaxClauseCount - set the maximum number of clauses permitted per *BooleanQuery.*
  Default value is 1024.

- `INDEX_LANGUAGE_DEPENDANT_FIELDS`
  define a list of fields to be indexed with a special language dependent analyzer.
  Note: this is currently still experimental (and only supported for English and German)

If you want to alter one of this configuration directives append the directive to the end of the instance configuration file (`<bscw-runtime-path>/conf/config.py`).

There you may also change the directories to contain the text file representations and the Lucene index itself. You may want to adjust some of the index parameter (such as merge factors) - see `http://lucene.apache.org` for details on how this affects indexing.

Command line tools

You may run the indexer using the provided command line tool

```
$ bin/bsadmin create_index
```

You may query the indexer using the command line tool

```
$ bin/bsadmin search
```

1. "`bsadmin create_index`" - generates the PyLucene index
   First make sure that no other indexing process is running. On Unix systems you may kill the indexing process as follows:
   ```
   $ kill `cat <bscw-runtime-path>/var/data/IndexLck`
   ```
   To start the indexing process on Unix systems you may use for example
   ```
   $ nohup ./bin/bsadmin create_index -cqt >/dev/null 2>&1 </dev/null &
   ```
   The commandline usage is as follows:
   ```
   bsadmin create_index -c [-u] [-otU] [-{v|q}] [-r <min>] [-R <hour>]
   bsadmin create_index -{i|s}  [-otU] [-{v|q}] [-r <min>] [-R <hour>]

   -c         create new index (forced if no index exists)
   -cu        create new index & force update of document text representations
   -i         incremental index update
   -s         scan database continuously
   -o         suppress periodic optimization (optimize only on start)
   -t         display timer info at exit
   -U         unlock at first (dangerous)
   -v         verbose
   -q         quiet
   -r <min>   report interval (default 30 min, 0: no report)
   -R <hour>  automatic restart in '+<hour>' or at 0 < <hour> < 24
   ```
   Note: option "`-u`" is only possible in conjunction with option "`-c`" (i.e. all text files will be removed before new Index is created)

   The "`bsadmin create_index`" script will create / update the PyLucene index. If no index exists yet it will be newly created. By default the script will update an existing index when it is invoked (use option "`-c`" to force creation of a new index).

   option "`-i`" will perform an incremental index update (default), i.e. only documents that have been modified or added (since last index run) will be (re-)indexed. Outdated (i.e. deleted) documents will be removed from the index.

   The indexing process will automatically create/update text representations of documents during indexing. This requires configuration of according converters (to text/plain format - see above).

   A document conversion will be performed when necessary, i.e. documents that have been modified will be updated; text representation of outdated (i.e. deleted) documents will be removed (use option "`-u`" to force removal of all text representations initially).

2. "`bsadmin search`" - performs a query on the PyLucene index

```
bsadmin search [-a] [-i] [-c] [-v] [-l <lang>] <query>
-a : search all fields (default: content search)
-i : search by ID only
-l : set user language for search
-c : show hit count only
-v : verbose
```

This script passes a query to the PyLucene index and returns a list of results as BSCW object IDs. It may be used for testing. Here verbose mode delivers extra document info on the results.

Note:

- option "`-i`" allows to check if an object (BSCW ID) is contained in the index.
- option "`-a`" allows to search in multiple fields (e.g. name, description etc.).

You may use any valid Lucene query, e.g.

```
$ bin/bsadmin search -v "contents:bscw AND class:Document"
```

Note: the command line search does not check any access rights, i.e. you will receive all results that match the query. When using the search in the web front-end, of course access rights are checked and only filtered results show up.

Index creation and update

If the package is enabled (in `<bscw-runtime-path>/conf/config.py`) and an index is already created (and not locked) BSCW attempts to automatically start the indexer when the BSCW server process is started (via "`bsadmin start`" [Windows] or "`start_servers`" [Unix]).

The "`bsadmin create_index`" tool provides an option ("`-s`") to continuously scan the database and thereby update the index (while BSCW server is running). This option is used when BSCW starts the indexer itself (actually option "`-sqr60`" is used).

Thus recommended usage of the indexer is to initially create the index manually using for example

```
$ bin/bsadmin create_index -cqt
```

and then let BSCW update the index continuously.

Note: the indexer logs progress and errors to the configured log file (e.g. `<bscw-runtime-path>/var/log/index.log`).

If the indexer was not started upon BSCW start due to a failure (e.g. a missing `IndexPos` file) run

```
$ bin/bsadmin create_index -viU
```

manually to incrementally index all missing objects. After "`bsadmin create_index`" finished updating the index restart your BSCW server, eg.

```
$ bin/start_servers -k                    UNIX
$ bin/start_servers

> bin/bsadmin stop [-s]                   Windows
> bin/bsadmin start [-s]
```

Note: if (for some reason) you ever want to completely re-build the index there are two options:

a) stop BSCW; then remove `var/data/IndexPos`; restart BSCW then create new index from scratch e.g. via "`bsadmin create_index -cuqt`" (see "create_index" above)

b) stop BSCW; then remove `var/data/Index*` and `var/data/Text/*`; restart BSCW then create new index from scratch e.g. via "`bsadmin create_index -cqt`" (see "create_index" above)

Finally restart BSCW again (to let BSCW update the index continuously).

Both methods should result in a 'fresh' (and up-to-date) index and newly created text representation of all indexable documents (if option -u is given in method a):

- Option a) allows a quick rebuild without updating text representations (if option `-u` is omitted).
- Option b) is the ultimate "from-scratch" solution as all index-data is cleaned up before rebuilding the index (you may also want to '`rm var/log/index.log`').

6.2 LDAP

The Lightweight Directory Access Protocol (LDAP) is a protocol for accessing online directory services. It runs directly over TCP, and can be used to access a standalone LDAP directory service or to access a directory service that is back-ended by X.500. The BSCW system implements an interface to LDAP servers based on the Python-LDAP package. Python-LDAP wraps an underlying LDAP C library that provides an RFC 1823 API, such as OpenLDAP (http://www.openldap.org/).

Installation

To install the BSCW LDAP module

1. The BSCW LDAP module needs the *python-ldap* package. Python-LDAP 2.4.10 provides functionality for accessing LDAP servers from Python. It wraps an underlying LDAP C library that provides an RFC 1823 API, and requires at least OpenLDAP 2.4.11+ (http://www.openldap.org/) to build.

   Download Python-LDAP at http://www.python-ldap.org// and install the package:

   - unpack the source distribution of Python-LDAP, e.g.

     ```
     $ gunzip < python-ldap-2.4.10.tar.gz | tar xf -
     ```

   - build and install Python-LDAP (as described in the `README` file)

     ```
     $ cd python-ldap-2.4.10
     $ vi setup.cfg
     $ python setup.py build
     $ python setup.py install
     ```

     This installs the shared library `_ldap.so` (among other things) in your Python `site-packages` path.

2. Additionally the BSCW LDAP module may need the *py-smbpasswd* package to create SMB password hashes if the LDAP directory uses LANMAN or NT password hashes.

   Download the *py-smbpasswd* at http://barryp.org/software/py-smbpasswd/ and install the package.

   Note: If the *py-smbpasswd* package is not found and your LDAP schema uses LANMAN or NT password hashes the `lmpassword`, `sambalmpassword` and `ntpassword`, `sambantpassword` attributes are ignored and the BSCW change password operation will fail.

3. To enable the BSCW *ldap* package run

   ```
   bin/bsadmin package -e ldap
   ```

   This creates a default `<bscw-runtime-path>/conf/ldap/config_ldap.py` configuration file.

4. Adapt the configuration file `conf/config_ldap.py` to your needs, especially the “`hosts`” map

and the "`auto_registration`" list:

- `hosts` is a dictionary mapping distinguished names (DNs) to hostname[:portnumber]

  When an LDAP object is searched (referred by a DN), this table is looked up for a corresponding LDAP server address. The DNs should be in a 'canonical' form (lower case, no spaces before or after '`,`' and '=').
- `certificate_files` is an dictionary containing for each LDAPS URI `hostname[:portnumber]` value from the `hosts` dictionary a path name to a file containing the CA certificates needed to validate server certificates.
- `auto_registration` is a list (or tuple) of patterns for generation of DNs from user names. An user is automatically registered with BSCW during login if all of the following items apply:
  1. direct HTTP server LDAP authentication (recommended; see notes for `use_ldap_passwords` below) or `COOKIE_AUTHENTICATION` is enabled (see BSCW configuration),
  2. the user with the given login name is not already a registered BSCW user,
  3. one of the generated DNs can be found in the corresponding LDAP server,
  4. the resulting LDAP object has an `objectClass` as defined in the `person_classes` list (default: `person_classes = [ 'person', ] ,`
  5. and the user can "bind" (authenticate) to the resulting LDAP object with the given password.

  When a BSCW user is registered in this way, all BSCW relevant attributes (i.e. the user details) are set by corresponding LDAP object attributes (see also `update_list` in `bs_ldap.py` for attribute name translation). A special '`ldap_bind`' attribute of the BSCW user object refers to the users' DN.

  Furthermore, if the `auto_registration` list is not empty and direct HTTP server LDAP authentication or `COOKIE_AUTHENTICATION` is enabled, then all BSCW users that have an LDAP binding set are authenticated via the LDAP server. During login, the BSCW server checks the success of a bind operation to the user's DN with the given password (see notes for `use_ldap_passwords` below).
- `auto_registration_roles` defines initial roles for automatically registered users. The list consists of pairs `('attribute=value', role)`

  Note: the role must be assignable for user objects i.e. it must appear in the list `cl_action.user_roles`.

  Note: at the moment the `'attribute=value'` string is only looked up in the DN (`user.ldap_bind`) of the user. The LDAP attributes of the user are ignored. This might be changed in the future.
- `use_ldap_passwords` defines how BSCW handles users with LDAP binding and local BSCW users (without LDAP binding):
  - If `use_ldap_passwords` is `1` then **for all users** passwords are verified against the LDAP-server. Hence an existing user who is not found on an LDAP server cannot login to the system any more.
  - If `use_ldap_passwords` is `2` then the user password is verified against the LDAP-server only for users with a LDAP binding or users found on a LDAP server. Note the following implications:
    - a local BSCW user who is not found on a LDAP server and who does not have a LDAP binding can still login to the system.
    - a local BSCW user who is found on a LDAP server and provided the correct LDAP

credentials will take over the local user (by adding a LDAP binding).

- If `use_ldap_passwords` is `3` then the user password is verified against the LDAP-server only for users that have a LDAP binding.

Note:

- By default BSCW cookie authentication is enabled and passwords are checked immediately against the LDAP server at the start of each session. Alternatively it is possible to configure the HTTP server to do direct LDAP authentication (e.g. via the Apache HTTP server `mod_authnz_ldap` or `mod_ldap` module).
- If the Apache HTTP server LDAP authentication modules are used and BSCW should be able to change passwords at the LDAP server `use_ldap_passwords` must be set to `3`, otherwise the BSCW change password action interferes with the LDAP authentication modules internal password cache.
- When using BSCW authentication only *digest authentication* is recommended. *Basic authentication* is supported, but discouraged to use because every request will be immediately authenticated against the LDAP server.

- `ldap_searches` defines a list of member search options `(qid, pattern)` or `(qid, pattern, pattern_args)` or `(qid, pattern, rdnfilter)` for the workspace invite member action (`op_addmb`):
  - `qid` is an unique identifier for the search and must be translated in `packages/ldap/messages/*/lg_msgconfig.py`.
  - `pattern` is a LDAP query where `'%(query)s'` is replaced by the user input of the addmb search form
  - `pattern_args` (optional) defines additional query input fields, which substitute `'%s'` occurrences within the query pattern. Pattern arguments are 3- or 4-tuples:

```
('entry-name-0', 'entry-default-val-0', 'entry-query-0'),
('entry-name-1', 'entry-default-val-1', 'entry-query-1', [
                                          'entry-dropdown-0',
                                          'entry-dropdown-1', ...]),
```

  - `rdnfilter` (optional) defines an optional filter for a relative DN type, which allows to additionally remove query results which do not match the given RDN value list

  Examples of LDAP queries:

  - search subtree of defined base DN(s) for the given query:

```
('mb_search_ldap_uid', 'cn=*%(query)s*'),
('mb_search_ldap_uid', '(|(cn=*%(query)s*)(uid=*%(query)s*))'),
('mb_search_ldap_uid', '(sAMAccountName=*%(query)s*)'),
```

  - search subtree with 2 input fields `'mb_search_ldap_cn'` and `'mb_search_ldap_uid'`

```
('mb_search_ldap',
     '(|(cn=%s)(uid=%s))',
     ('cn', '', '*%(query)s*'),
     ('uid', '', '*%(query)s*'),
),
```

  - search `subtree` of defined base DN(s) for query '`ou=*%(query)s*`' and remove results which relative DN of type '`ou`' does not match the given list `['sales', 'ext']`:

```
('mb_search_ldap_org', '(ou=*%(query)s*)', ('ou', ['sales', 'ext'])),
```

LDAP Browser

After installation of the LDAP package, a small “organisational browser” is enabled. When opening a user info window (e.g. by clicking on a user name in the web interface) the users' LDAP binding (if defined) is shown. By selecting the link of the LDAP binding field the user information (as retrieved from the LDAP server) is displayed.

If the LDAP package is installed and activated, the `Goto`-Menu contains an entry `Organisation Info` that invokes the organisational browser. The browser connects to the LDAP servers in the hosts map and allows operation on LDAP objects. The operations search, view and attribute editing are supported.

Note: `ORG_INFO_URL` must **not be set** in `<bscw-runtime-path>/conf/config.py`, because this will override the handler invoked by the `Organisation Info` menu entry.

Note: You need basic knowledge of directory services in general and especially need to know some details about LDAP in order to configure BSCW for LDAP. Besides the more technical IETF RFCs and Drafts about LDAP – which can be found at http://www.ietf.org/ – we suggest to read the IBM Redbook: *Understanding LDAP* (SG-244986, June 1998), available at http://www.redbooks.ibm.com/.

6.3 Blog (Weblogs)

The package *blog* extends BSCW by blog functionality. You either can create personal blogs, group blogs or public blogs.

At creation of a blog or in the blog properties one can define some handling options and set up default access right, i.e. who should add blog entries and who should read the blog. By default everyone who could read blog entries also can make comments. The access rights may can be changed individually by editing the roles.

Also it is possible to specify the layout of a blog, either as default layout, as a layout with BSCW navigation or as a user defined layout with an own template and an own style sheet file.

This package is enabled by default in a new BSCW server installation. No additional software installation or configuration is required on server-side. If disabled, the package may be enabled again by running

```
        bin/bsadmin package -e blog
```

6.4 Case – File Synchronisation

The *case* package provides an optional feature for BSCW that allows users to synchronize documents stored in their shared workspaces with their local file system (i.e. Windows PC). You may want to enable this package if you want to offer this additional functionality to your end users.

After the case package is activated a new top-level object 'Case' is visible at the user interface (in 'Goto' menu/icons)

This package is enabled by default in a new BSCW server installation. No additional software installation or configuration is required on server-side. If disabled, the package may be enabled again by running

```
        bin/bsadmin package -e case
```

You may want to provide different global defaults for your users in the instance configuration file (`<bscw-runtime-path>/conf/config.py`). The possible configuration directives and their defaults are as follows:

- `CASE_LOCAL_PATH`
  defines default case path on local disk. Note: the `%s` is replaced by the user name
  ```
  CASE_LOCAL_PATH = 'C:\\Users\\%s\\BSCWCase'
  ```
- `CASE_MAX_VERSIONS`
  defines maximum number of versions to be stored in local case
  Note: user may choose whether versions of documents shall be downloaded
  ```
  CASE_MAX_VERSIONS = 3
  ```

Notes:

- This feature is only available for Windows (XP/Vista/7) Operating Systems (client-side).
- This feature is only available in the professional edition of BSCW.
- See BSCW Help for further details.

6.5 Export PDF

The *exportpdf* packages provides an optional feature for BSCW that allows users to export container views to PDF format. With PDF export enabled the listings of many container objects, i.e. objects that can contain other objects, may be exported in PDF format for printing. Examples are folders, blogs and contact lists. You may want to enable this package if you want to offer this additional functionality to your end users.

For installation and configuration of the package proceed as follows:

1. Make sure the required third-party software is available on your system (server). The package requires the following python extensions:
   - Python Imaging Library (PIL)
     http://www.pythonware.com/products/pil/
   - The ReportLab PDF Library (versions 2.4 - 2.7 are supported)
     http://www.reportlab.org/
2. To enable the BSCW *exportpdf* package run
   ```
   bin/bsadmin package -e exportpdf
   ```

Note: This feature is only available in the professional edition of BSCW.

6.6 Document Generator

The Document Generator facilitates the management of sets of documents which need to be continually revised and periodically published. Based on hierarchically structured templates, documents within such document sets may be automatically created by an evaluation process using the templates. The Document Generator can thus help to reduce redundancy and to achieve a consistent structure and layout of the documents. The Document Generator is particularly useful for - but not restricted to – shared website management.

To enable the BSCW *factory* package run

```
bin/bsadmin package -e factory
```

6.7 Flow-Folder

Flow folders allow you to manage work flows where documents follow a certain work process and are forwarded from one user to another for subsequent processing. Each flow folder has a number of tasks which are to be carried out by the users responsible in the order specified. Flow folders - like normal folders - may contain objects of all types, e.g. documents, other folders or discussion forums.

This package is enabled by default in a new BSCW server installation. No additional software installation or configuration is required on server-side. If disabled, the package may be enabled again by running

```
bin/bsadmin package -e FlowFolder
```

6.8 Mobile access

The "mobile" package provides an alternative mobile user interface to BSCW, especially designed for modern smart phones with HTML & JavaScript browsers.

On each login, users can decide which interface they want to use, while BSCW already proposes the interface that fits best to the requesting browser.

As the mobile interface is tightly coupled with the BSCW core, it doesn't offer any package-local settings.

The "mobile" package is enabled by default on new BSCW servers and supersedes the old "Mobile" package. Thus, it no longer requires an XSLT processor or other prerequisites. In turn, pure WAP-capable devices are no longer supported.

Note: The "mobile" package requires cookie authentication as authentication method (see section 5.2.4 on page 59 for details).

If disabled, the package may be enabled again by running

```
bin/bsadmin package -e mobile
```

6.9 MoinMoin Wiki integration

The '*moin*' package generates a configuration which connects a *MoinMoin* Wiki (`http://moinmo.in/`) with a BSCW workspace, i.e. it create a new instance of a *MoinMoin* Wiki and configure it in such a way, that only members of a given BSCW workspace can access it.

This package supports *MoinMoin* version 1.9, 1.8 or 1.5 in combination with the Apache HTTP server.

1. Install either
   - MoinMoin 1.9 (`http://hg.moinmo.in/moin/1.9/`)
   - MoinMoin 1.8 (`http://hg.moinmo.in/moin/1.8/`)
   - MoinMoin 1.5 (`http://hg.moinmo.in/moin/1.5/`)
2. To enable the BSCW *moin* package run
   ```
   bin/bsadmin package -e moin
   ```
3. Append the following configuration variables to `<bscw-runtime-path>/conf/config.py`
   - `MOIN_INSTALLATION` - path to *MoinMoin* Wiki program folder.
   - `MOIN_PYTHON` - path to python interpreter used by *MoinMoin* Wiki.
   - `MOIN_SHARE` - path to share folder of *MoinMoin* Wiki. It must contain the default directories for '`data`' and '`underlay`'.
   - `MOIN_INSTANCES` – name of a directory where sub-directories for the different *MoinMoin* Wiki instances should be created (defaults to `'moin'` which creates a directory: `<bscw-runtime-path>/moin`).
4. To enable the "moin" package on existing servers, configure a *MoinMoin* Wiki for a shared workspace as follows:
   ```
   bin/bsadmin conf_moin <moinid> <folderid> [<lang>]
   ```
   configures a *MoinMoin* Wiki for the members of the given folder. Only the members of the

folder can access the Wiki.

- `<moinid>`
  virtual path to access the *MoinMoin* Wiki under the URL `http://<server>/<moinid>/`
  Notes:
  - only alpha numeric characters and _ are allowed,
  - `<moinid>` **must not** `moin`, which conflicts with the *MoinMoin* internal path alias.
- `<folderid>`
  id or full path to a BSCW folder
- `<lang>`
  language for Link in folder and default *MoinMoin* Wiki language (default: en).

After creating a new instance add (if necessary) one of the following lines (depending on your Apache HTTP server version)

```
Include "<bscw-runtime-path>/conf/apache2/moin.conf"
Include "<bscw-runtime-path>/conf/apache24/moin.conf"
```

to your Apache site configuration file and restart your Apache HTTP server.

Hint: If you use MoinMoin 1.9 open after installation the location

```
http://<server>/<moinid>/LanguageSetup
```

to install the preferred help and system page packages.

Attention: The default location of the *MoinMoin* instances was moved from `<bscw-runtime-path>/var/data/moin` to `<bscw-runtime-path>/moin`. To preserve all existing *MoinMoin* Wikis the contents from `<bscw-runtime-path>/var/data/moin` must be moved to `<bscw-runtime-path>/moin`. Since the file layout has been changed `bin/bsadmin conf_moin -r` must be run to update the configuration. The newly generated configuration file `conf/apache{2,24}/moin.conf` must be included into the Apache HTTP server site-configuration (in place of the old `var/data/moin/apache.conf` file).

6.10 Poll

The *poll* package provides several types of opinion surveys in BSCW. These surveys can be left open to the public (*Poll*) or limited to a closed participant group (*Voting*).

Appointment Schedulings provide a convenient way to agree on meeting dates with a larger group of participants. While Polls are available in all editions of BSCW, Votings and Appointment Schedulings require a professional license.

The *poll* package is enabled by default on new BSCW servers and requires no external components. If disabled, the package may be enabled again by running

```
bin/bsadmin package -e poll
```

When the package is activated a new object 'Poll' is enabled at the user interface (in 'File/New' menu).

There is no special configuration required for this package. However you may change some defaults and system behaviour in the instance configuration file (`<bscw-runtime-path>/conf/config.py`) by appending configuration directives. The possible configuration directives and their defaults are as follows:

- `VOTING_TOKEN_EXP`
  Voting participants receive email notifications with links to access the Voting. Each link includes

an individual security token with temporary validity. After the token has expired, the access to the Voting is denied. The token's lifetime usually depends on the specified end date of the Voting to allow access (and voting) at least until the end of the Voting. If no Voting end is specified, the token's lifetime is calculated from the start date (or the current time, if no start date is specified).
`VOTING_TOKEN_EXP` allows to specify the lifetime of tokens in case no clear end date can be calculated.
Possible values are strings which may be specified in seconds or minutes (hours, days, weeks) by using an additional `'s'`, `'m'` (`'h'`, `'d'`, `'w'`) suffix.
Example:`VOTING_TOKEN_EXP = '1w'` would specify one week
Note: `VOTING_TOKEN_EXP` should be at least `'1d'` (it must be > 0).

- `SCHEDULE_SUGGESTIONS_ENABLED`
  Defines if the option 'New participants may suggest others for voting' should be available for Appointment Schedulings. (Otherwise, `SCHEDULE_SUGGESTIONS_DEFAULT` will apply)

- `SCHEDULE_SUGGESTIONS_DEFAULT`
  Defines the default value for the option 'New participants may suggest others for voting'.

- `SCHEDULE_CONFIRMATION_ENABLED`
  Defines if the option 'Suggested participants need to be confirmed by me'
  should be available at all. (Otherwise, `SCHEDULE_CONFIRMATION_DEFAULT` will apply)

- `SCHEDULE_CONFIRMATION_DEFAULT`
  Defines the default value for the option 'Suggested participants need to be confirmed by me'

- `SCHEDULE_CONDITIONALVOTE_ENABLED`
  Defines if the option 'Participants may vote with Maybe' should be available at all.
  (Otherwise, `SCHEDULE_CONDITIONALVOTE_DEFAULT` will apply)

- `SCHEDULE_CONDITIONALVOTE_DEFAULT`
  Defines the default value for the option 'Participants may vote with Maybe'

6.11 Portal

The *portal* package provides an optional feature for BSCW that allows users to configure a personal portal page - as well as to add a portal to a shared workspace. Within a portal various portlets may be added and configured. Portlets may display information stored within BSCW (folders, tasks etc.) as well as information stored in other sources like RSS feeds.

When the package is activated a new top-level object 'Portal' is enabled at the user interface (in 'Goto' menu/icons)

Note: This feature is only available in the professional edition of BSCW.

For installation and configuration of the package proceed as follows:

1. *Enable the Portal package*
   If disabled, the package may be enabled again by running
   ```
   bin/bsadmin package -e portal
   ```
2. *Optional step: change the Portal configuration*

You may change the portlet configuration in the instance configuration file `<bscw-runtime-path>/conf/config.py` by overriding the defaults. The following settings may be changed to your need:

- `PORTAL_SHOWATLOGIN`
  determines if the user portal page is immediately displayed after the user logs in. (By changing this setting to `False` the users' home folder is shown after login instead.)
- `PORTAL_WIDGETS_ITEMS`
  number of items to show per page used by (most) internal (BSCW) widgets user configurable.
- `PORTAL_PORTLETS`
  list of available/enabled portlets - i.e. portlets users may add using 'File/New/Portlet'. This may include predefined portlets as well as your own portlets from `<bscw-runtime-path>/conf/portal/config_portlets.py`.
- `PORTAL_AUTO_CONFIG_USR_PORTAL`
  a list of portlets that will be automatically (initially) added to a users' personal portal

Notes:

- all settings will take effect for all users on this BSCW server. Automatic initialization will only take effect for new portals/users.
- you may (as administrator) check the current settings by using the `bsadmin getconfig` and query for example the `PORTAL_PORTLETS`:

```
$ bsadmin getconfig PORTAL_PORTLETS
```

3. *Optional step: provide additional portlets*

You may also add further portlets to the list of available portlets and thereby make them available to your users in the BSCW portal(s) by editing the file

```
<bscw-runtime-path>/conf/portal/config_portlets.py
```

and finally defining a list of your own custom portlets in the central configuration file

```
<bscw-runtime-path>/conf/config.py
```

These additional portlets will become available for all users, e.g.

```
CUSTOM_PORTLETS = [
     'weather_portlet',
     'rss_tagesschau_portlet',
     'netvibes_dilbert_portlet',
     'vimeo_portlet',
]
```

Note each additional portlet listed in your `CUSTOM_PORTLETS` configuration must also be defined in the custom portlet config file

```
<bscw-runtime-path>/conf/portal/config_portlets.py
```

Initially a sample configuration file for additional portlets is created as default configuration in

```
<bscw-runtime-path>/conf/portal/config_portlets.py
```

The suggested procedure is to first adopt `config_portlets.py` to your needs, i.e. define your own portlets (see below), and then define the `CUSTOM_PORTLETS` list in `<bscw-runtime-path>/conf/config.py` to add the portlets you want to provide.

Note: You may want to provide translation for the new portlets you've added. You may define the portlet title in different languages as well as its description and settings. You need to add an according message files in

```
<bscw-runtime-path>/bsext/msg/en/portal/custom_portlets.py
<bscw-runtime-path>/bsext/msg/de/portal/custom_portlets.py
...
```

For example the title and description of the sample `weather_portlet` could be defined in a message file (like `custom_portlets.py`) as follows:

```
weather_portlet = 'Weather (Germany)'
weather_portlet_desc = 'watch German weather forecast'
```

Regarding definition of custom portlets, currently the following options exist

- *static content*: allows you to define a portlets with static HTML content this is the most simple extension and useful for showing messages. Note: the content will be **copied** once the portlet is added by the user (see `helloworld_portlet` for example)!
- *iframe integration*: allows you to integrate external sources - either from external websites or even from your (possibly internal) websites use `PORTLET_CLASS: "URLFramePortlet"` as basic setting (see `doodle_portlet` for example).
- *RSS integration*: allows you to define specific feeds to be integrated as a portlet - useful for showing news / announcements etc. (see `tagesschau_portlet` for example).
  Note: you may also use the Netvibes News Widget for new integration however the news provider must support Netvibes News then (see *netvibes_welt_portlet* for example).
- *Netvibes widget integration*: allows you to integrate widgets provided by Netvibes - you may find a list of available widgets at http://eco.netvibes.com/widgets The Netvibes Widget must support the Export feature (known as UWA Blog Export or UWA IFrame Export - see http://dev.netvibes.com/doc/uwa/howto/uwa_iframe). You will need the `moduleUrl` from the Netvibes widget (use `'share'`) (see `netvibes_ipcheck_portlet` for example).

Disclaimer: External Widgets

When using external widgets you may have to check the terms of use of the widget provider before inserting the widget into the portal! OrbiTeam is not reliable for any damages incurred by external widgets or any interruption of external services used by integrated widgets.

Integration of external widgets using GoogleMaps, Netvibes Widgets or IFrame technology has been done according to current web standards. External widgets should not be allowed to access any private information stored within the BSCW shared workspace system due to current web security standards (and their implementation in JavaScript and todays Web Browser technology). However there is still a risk of potential security vulnerability by external widgets. We therefore recommend to only integrate external content and/or services from trusted providers.

Google®, Netvibes®, Facebook®, Twitter® etc. are registered trademarks of the according companies in different countries of the world. OrbiTeam is not a partner, affiliate, or licensee of those companies. OrbiTeam acknowledges the terms of use of the provided services of those companies.

By using any of the integrated services provided via the BSCW portal you (as provider of the local BSCW) - and possibly your end users (as registered users of the external services) - have to accept the terms of services of the external provider of the integrated services. At any time you may disable any of the external services provided via the BSCW portal by changing the configuration file on your BSCW server.

OrbiTeam is not reliable for any consequences that occur from not accepting the terms of services of the external service providers or even from any abuse of the external services by end users.

6.12 Presence

The package '*presence*' indicates the "BSCW presence" of other members in your shared workspaces. In order to use this feature you may activate a presence toolbar which shows the presence status of each member. The toolbar is shown below the navigation bar of the shared workspace.

The "BSCW presence" is measured by the activity of a user in his/her web browser window where BSCW is used, i.e. if an user activates this window, moves the mouse over this window or types any keys on this browser window he becomes active in the sense of "BSCW presence". The different states of the user presence result on different time periods defined in the configuration file. If a user is active the presence toolbar is updated automatically.

To enable the BSCW *presence* package run

```
        bin/bsadmin package -e presence
```

You may need to adapt the presence configuration by editing the instance configuration file (`<bscw-runtime-path>/conf/config.py`) and appending configuration directives. Possible configuration directives and their defaults are as follows:

- `P_TOOLBAR`
  defines the default view of the toolbars for users which have not changed it by themselves: `0` - hide, `1` - show all, `2` - show online. Default `P_TOOLBAR = 0`
- `P_SHOW_ALL_LIMIT`
  If a folder has more members than the `P_SHOW_ALL_LIMIT` the view of the presence toolbar changed to the the 'show online' mode. Default: `P_SHOW_ALL_LIMIT = 100`
- `P_REFRESH`
  gives the refresh timeout of the presence toolbar. Default: `P_REFRESH = 60`
- `P_AVAILABLE_TIMEOUT`, `P_RECENTLY_TIMEOUT`, `P_INACTIVE_TIMEOUT`
  define time periods after how much seconds the user state will changed if the user is inactive. Defaults

```
  P_AVAILABLE_TIMEOUT =  300
  P_RECENTLY_TIMEOUT  = 3600
  P_INACTIVE_TIMEOUT  = 7200
```

6.13 Readers

The package '*readers*' gives an action to select and filter the BSCW events to answer questions like

- Who has read my documents?
- Which documents I have read?

In the selection box at the end of the form you can select the documents which has been created, modified or read by a selected member. If an indexer is installed the content of the selected documents can be filtered by an search pattern. Initially the action shows you all members which have read documents created by yourself.

The results of the selection or all documents of the folder can be visualized by a graph. This graph may answers the questions

- Who are the most active users?
- Which are the important documents?

For visualisation the HyperGraph applet is used (`http://hypergraph.sourceforge.net/`). It requires Java installed in your browser and JavaScript to control the HyperGraph applet.

No additional software installation or configuration is required on server-side. This package is not enabled by default. To enable the BSCW *readers* package run

```
        bin/bsadmin package -e readers
```

6.14 RSS

The package "*rss*" is implemented using the PyRSS2Gen.py library of Andrew Dalke and the Universal feed parser library by Mark Pilgrim. See file `PyRSS2Gen-LICENSE.txt` and `feedparser.py` for licensing and copyright conditions.

BSCW's RSS package supports:

1. import of external RSS feeds into BSCW via "RSS Feed" objects BSCW's "RSS Feed" objects allow subscription to arbitrary RSS feeds which will then be treated like BSCW folders.
2. export of BSCW event histories as RSS news feed BSCW events are rendered as a RSS 2.0 news feed which external RSS "aggregators" (or RSS "readers") may poll for RSS formatted news items.

For a (comprehensive) description of the RSS 2.0 protocol see

```
   http://blogs.law.harvard.edu/tech/rss
```

Numerous RSS feed readers can be found in the web. Many modern web browsers like Firefox, Opera, or Safari allow to directly import and read external RSS feeds. See your browser's documentation for more information.

6.14.1 Export of BSCW event histories as RSS news feed

Depending on your server configuration `http://<server>/bscw/bscw.cgi/?op=rss` will render an XML formatted RSS news channel of all events which the authenticated user has access to.

You may adapt the rss configuration by editing the instance configuration file (`<bscw-runtime-path>/conf/config.py`) and appending the following configuration directives

- `RSS_TIME_SPAN`
  is the age of the event from now in seconds. By default, `RSS_TIME_SPAN` is set to a `7` day period.
- `RSS_AUTHENTICATION`
  determines the used authentication method: if set to `1` always authentication credentials are required from the RSS news reader; if set to `0` *unauthenticated* reading of BSCW's RSS events using token authentication is allowed. This is the default (see 6.14.3 below)

No additional software installation or configuration is required on server-side. This package is enabled by default. If disabled, the package may be enabled again by running

```
        bin/bsadmin package -e rss
```

Note that the operation `op=rss` will only appear to the end users interface, if the user has set her user profile level to "EXPERT". The operation will appear under menu "*GoTo*", action "*Events*". It will result in the original RSS XML formatted output. The user may then copy the URL from her browser's URL field into whatever RSS news aggregator she uses.

6.14.2 RSS reference links for use by RSS-compatible browsers

The RSS package, if activated, also includes an RSS reference into any of BSCW's container HTML rendering. Thus

```
<link rel="alternate"
    type="application/rss+xml" title="BSCW RSS 2.0 NewsFeed"
    href="http://<server>/bscw/bscw.cgi/?op=rss" />
```

(depending on your server configuration) is included in the HTML head section. This allows RSS compatible browsers, like Firefox, Opera or Safari, to display an icon which indicates that an RSS news feed is supported: end-users click on that icon to create a *LiveBookmark* in Firefox or directly open the BSCW news feed in their browser (Opera and Safari). Firefox also offers a number of plugins which display news feeds more nicely. See `http://addons.mozilla.org/`extensions and search for "RSS".

Please note you will a least need the following web browser version zto support RSS: Firefox: version 1+, Opera: version 8+, Safari: version 2+, Internet Explorer 7+. Mozilla Thunderbird 2+ is also capable of displaying RSS 2.0 news feeds.

6.14.3 Authenticated and unauthenticated reading of RSS event feeds

Not all external RSS news aggregators are capable of requesting authentication credentials. Therefore, the RSS package can be configured to allow for unauthenticated access via public script, using, e.g.,

```
http://<server>/pub/bscw.cgi/?op=rss&token=123:aksddf34sd$tt"
```

where "token" consists of a pair (user id: her encrypted password). Btw. encrypted passwords can be obtained using the system administrator's shell command `bsadmin users -p user_name`.

At the user interface, menu command "*GoTo/Events*" will either render a secure script URL (`http://<server>/bscw/bscw.cgi/?op=rss`) or a public script URL (`http://<server>/pub/bscw.cgi/?op=rss&token=123:aksddtt`), depending on the value of `RSS_AUTHENICATION`.

This also holds true for the RSS reference link included in BSCW containers' HTML source code rendering, cf. 6.14.2 above.

6.15 Secure key management

The package '*Secure*' provides basic support for key management in BSCW.

End users may upload a public key to be stored in BSCW and made available to other BSCW users. Within a shared workspace users may define a public group key - by uploading an existing key or generating a new public key.

A key pair containing a public key and a private key can be generated using the BSCW helper application BSCW Desktop and its Key Manager component.

The key pair can also be generated using any one commercially or freely available key toolkits such as PGP (Pretty Good Privacy) or GnuPG (GNU Privacy Guard).

Requirements

The package '*Secure*' complements the functionality of the BSCW Desktop application.

The BSCW Desktop is a standalone Java application that requires a local Java runtime environment (JRE 1.6 or later). The application provides functions for easy and efficient document upload to the BSCW server as well as for key management and data encryption on the client.

The BSCW Desktop application allows users to encrypt and sign documents using member public keys or group public key stored in BSCW. End users must maintain the private keys for decryption on the client (in the Key Manager component).

Thus for effective usage of the Secure package's features the use of the BSCW Desktop application is required. Furthermore an installation of the GnuPG commanline key toolkit for generation of key pairs is required.

GnuPG binaries are available for both Windows and Linux:

```
http://www.gnupg.org/
```

Finally, the Python wrapper for GnuPG - python-gnupg - is required:

```
http://packages.python.org/python-gnupg/
```

Note: This feature is only available in the professional edition of BSCW.

Configuration

This package requires only minimal configuration by the administrator. To enable the BSCW *Secure* package run

```
bin/bsadmin package -e Secure
```

After installation of the GnuPG software you may need to configure the path to the `gpg` binary - in case it is not included in the default `PATH` (of the `bscw` user). In this case append to `<bscw-runtime-path>/conf/config.py`

- `PGP_COMMAND`
  defines the absolute path to your `gpg` binary:

```
# Windows sample config:
# PGP_COMMAND = "D:\\Programme\\GNU\\GnuPG\\gpg"

# Unix sample config(s)
# PGP_COMMAND = '/usr/bin/gpg'
```

Note: you may need to configure the GnuPG software to use a special user-id and provide a location for the local GnuPG data, e.g.

```
PGP_COMMAND = \
  '/usr/bin/gpg -u www-data --homedir <bscw-runtime-path>/data/Temp/.gnupg'
```

In this case the httpd runs as '`www-data`' and the `data/Temp` directory needs '`chown www-data`' with full access (`rwx`) to user '`www-data`' and set-group-id for group '`bscw`' (`rws`).

Note: depending on your OS/Hardware you may need to ensure that the system provides sufficient entropy (required for key generation by GPG). If the system runs out of entropy GPG might stop in the middle with a message like this:

```
++++++++++..+++++.++++++++++++++.++++++++++...++++++++++++++...++++++
Not enough random bytes available. Please do some other work to give
the OS a chance to collect more entropy! (Need 284 more bytes)
```

If your experience such problems you may install “rng-tools” on Linux, which will utilize a hardware random generator to provide entropy. For test purposes you can omit the hardware random device and use `/dev/urandom` (eg. using `/usr/sbin/rngd --rng-device=/dev/urandom` or setting `HRNGDEVICE=/dev/urandom` in `/etc/{default|sysconfig}/rng-tools`). Do **never** use this setting in a production environment, since this will introduce a weakness in generated keys!

6.16 SSO – Single Sign On

BSCW supports different mechanisms for integration with an existing Single Sign On (SSO) infrastructure. By using SSO a BSCW server may be integrated into an IT infrastructure where different applications share the same user base and provide a central login mechanism the end users (e.g. in a web portal).

BSCW now supports CAS (Central Authentication Server), an open source SSO server developed by Yale University (see `http://www.ja-sig.org/products/cas/`), Shibboleth, a standards-based, open source middleware software which provides SSO even across organizational boundaries (see `http://shibboleth.net/`) and OpenID (see `http://openid.net/`), .

6.16.1 CAS Authentication

CAS authentication allows users to authenticate at a central authentication server. In combination with a LDAP service first time CAS users are automatically registered at their first login at the BSCW server. To configure CAS

1. Edit the main server configuration file `<bscw-runtime-path>/conf/config.py` as follows:

- Define the URL of the CAS Single Sign On service, e.g.

```
CAS_URI = 'http://sso.domain.org:8080/cas'
```

- Define a Single Sign On prefix and enable cookie authentication for this prefix

```
SSO_PREFIX = '/cas/'
SSO_COOKIE = ('bscw_cas', None, 120)
```

- To define an alternate secure authentication path for CAS enter the tuple

```
(SSO_PREFIX, { 'mode': AUTH_MODE, 'cookie': SSO_COOKIE })
```

in `SCRIPTS_ALIASES`, e.g.

```
SCRIPTS_ALIASES = {
    '/bscw/': [
        (SSO_PREFIX,
            {'mode': AUTH_MODE, 'cookie': SSO_COOKIE }),
    ]
}
```

2. Create a new Apache HTTP server configuration with

```
$ ./bin/bsadmin conf_apache -n
Configure 'gzip' compression ...
Configure 'static' resources 'var/www/20120101-1200-25000'...
 (Long time future expire dates)
Configure secure prefix '/bscw/' ...
 (HTTP_AUTHORISATION passed to BSCW)
 (Cookie authentication enabled)
Configure secure prefix '/cas' ...
 (HTTP_AUTHORISATION passed to BSCW)
 (Cookie authentication enabled)
Configure public prefix '/pub' ...
 (No authentication)

Creating Apache HTTP server configuration files in
/opt/bscw/srv/bscw.domain.org/conf/apache{2,24}
   mod.conf ... module configuration file
  site.conf ... virtual host site configuration file
  bscw.conf ... BSCW configuration file
```

and restart your web server to reload its configuration, e.g.

```
/etc/init.d/apache2 restart
```

6.16.2 OpenID

In order to activate OpenID single-sign-on registration and authentication see `http://openid.net/`. Afterwards edit the main server configuration file `<bscw-runtime-path>/conf/config.py` and define

```
OPEN_ID_DEFAULT = ("openid.net", "http://openid.net/get-an-openid")
```

This will show a link to the "default provider" `openid.net` in the login page. This enables a user to get an OpenID URL if he does not have one. If you do not want to give a link to a default provider set

```
OPEN_ID_DEFAULT = ("", "")
```

Note: `COOKIE_AUTHENTICATION` must be set and location (see above) must be `None` when OpenIDs are used. Also the `python-openid` package must be installed (tested with `python-openid-2.2.1`, `http://openidenabled.com/python-openid/releases/`).

OpenID registration and authentication is disabled with

```
OPEN_ID_DEFAULT = None
```

6.16.3 Shibboleth Authentication

Shibboleth allows users to log in via Single Sign-On as well as normal users to log in via user name and password. First time Shibboleth users can be automatically registered and their profile can be updated on every Login, so that their user details always up-to-date.

The '*Shibboleth*' package was removed in BSCW 4.5 and its functionality was merged into the BSCW kernel (see section 5.2.4 on page 60). Please disable the '*Shibboleth*' package if it was not automatically disabled by the BSCW 4.5 installation.

Shibboleth Service Provider configuration

In order to use BSCW with Shibboleth a Shibboleth Service Provider (e.g. Apache `mod_shib`) has to be installed on the same host like BSCW. Please refer to the deployment guides of your federation or to the official Shibboleth Wiki `https://wiki.shibboleth.net/confluence/display/SHIB2/Home` on how to install and configure a Shibboleth Service Provider in your environment.

BSCW needs at least the following values for an authenticated Shibboleth user:

- Application ID (`Shib_Application_ID`)
- Identity Provider (`Shib_Identity_Provider`)
- Email address (mail)

The environment variables `Shib_Application_ID` and `Shib_Identity_Provider` shoud be automatically set by *mod_shib* (BSCW automatically switches back to `HTTP_SHIB_APPLICATION_ID` and `HTTP_SHIB_IDENTITY_PROVIDER` for old (not recommended) Shibboleth 1.3 installations, see below).

Please make sure that the mail attribute is available for all Shibboleth users once they are authenticated. Also ensure that the Shibboleth 2.x `attribute-map.xml` maps the above attributes to a web server environment variable with the name given between parentheses.

For example, to configure BSCW to use the Shibboleth attribute containing the email address of a user, you would change Shibboleth's `attribute-map.xml` configuration like this:

```
<Attribute
```

```
        name="urn:oid:0.9.2342.19200300.100.1.3"
        id="Shib-InetOrgPerson-mail"
        aliases="mail" />
```

BSCW configuration

You must add an entry for your federations at two places within the instance configuration file (`<bscw-runtime-path>/conf/config.py`). In the example we show it for the federation '`SnakeOilProviders`' and also as a a commented entry for '`BscwTest`':

```
FEDERATIONS = {
    'SnakeOilProviders': ('login_shib', '/snakeoil-login.gif'', (
        (r'[^@]*@snake-oil\.com', 1),
        (r'[^@]*@snake-oil\.de', 1),
    )),
    # Another federation
    #'BscwTest': ('login_shib', '/bscwtest-login.gif'', ()),
}

SCRIPTS = {
    ...
    '/pub/snakeoil/':
        ('SnakeOilProviders', '', CREATE_SCRIPTS, SECURE_SCRIPTS),
    # Another federation
    #'/pub/bscwtest/':
    #    ('BscwTest', '', CREATE_SCRIPTS, SECURE_SCRIPTS),
}
```

Notes:

- If you need more than one federation you must configure them with different Application Ids in the Shibboleth configuration. The Application Ids must be 'default' or match the name given in `FEDERATIONS` and `SCRIPTS`.
- If you make changes like this to the instance configuration file (`<bscw-runtime-path>/conf/config.py`) you have to regenerate the Apache configuration and index pages with "`bsadmin conf_apache`" and "`bsadmin index_page`" respectively. This also requires a restart of the Apache server.
- If Shibboleth is the only/primary authentication system for BSCW, we also recommend setting

  ```
  SERVER_LOGOUT = '/Shibboleth.sso/Logout?return=/pub/'
  ```

  (it depends on your Shibboleth configuration and we have not a good idea yet how to do it with more than one federation).

  This then destroys not only the BSCW but also the Shibboleth session and sends the user back to the BSCW start page. This should work even if a user does not have a Shibboleth session.

The following CGI environment variables are interpreted by BSCW:

Shibboleth 2.x	Shibboleth 1.3	BSCW key
`Shib_Application_ID`	`HTTP_SHIB_APPLICATION_ID`	`shib_app_id`

Shib_Identity_Provider	HTTP_SHIB_IDENTITY_PROVIDER	shib_idp
mail	HTTP_SHIB_INETORGPERSON_MAIL	email
givenName	HTTP_SHIB_INETORGPERSON_GIVENNAME	givenname
sn	HTTP_SHIB_PERSON_SURNAME	surname
org-dn	HTTP_SHIB_SWISSEP_HOMEORGANIZATION	org
telephoneNumber	HTTP_SHIB_PERSON_TELEPHONENUMBER	phone
homephone	HTTP_SHIB_INETORGPERSON_HOMEPHONE	homePhone
mobile	HTTP_SHIB_INETORGPERSON_MOBILE	mobile
language	HTTP_SHIB_INETORGPERSON_PREFERREDLANGUAGE	preferredLanguage

BSCW needs only values for `shib_app_id`, `shib_idp`, and `email`. The others are optional. If your Shibboleth installation sets other CGI environmet variables, e.g. `Shib-IDP` instead of `Shib_Identity_Provider` and `Mail` instead of `mail` (i.e. you don't want to use an Attribute alias) then you can redefine the environment keys in the instance configuration file (`<bscw-runtime-path>/conf/config.py`) by adding:

```
HTTP_SHIB_ENVIRONMENT = [
    #(bscw_key, evironment_key)
    ('shib_idp', 'Shib-IDP'),
    ('email', 'Mail'),
]
```

6.17 SMS

The package '*SMS*' provides a simple interface for the editing and delivery of short messages in BSCW. SMS messages are sent to mobile phones using the GSM short message standard and can be read with any GSM phone.

The SMS package requires 3rd party software for the actual transmission of SMS messages to the SMSC (SMS center) of the mobile network provider. It does only provide a framework for the editing and delivery of messages in BSCW.

The current implementation works with the 'Alamin' or the 'Yaps' SMS gateway which must be installed on the local or a remote host.

- For installation details of 'Alamin' see http://www.alamin.org/.
- For 'Yaps' version 0.96 and the KSMS server mode patch are required.

After installation of the SMS gateway must append the `SMS_SERVER` configuration variable in your instance configuration file `<bscw-runtime-path>/conf/config.py`:

- `SMS_SERVER`

to define used SMS gateway service.

```
# Configuration for Yaps:
SMS_SERVER = ('Yaps', ('your.yaps.host', 909))
# Configuration for Alamin:
SMS_SERVER = ('Alamin', ('your.alamin.host', 11201))
# The Alamin server also supports user Authentication:
SMS_SERVER = ('Alamin', ('your.alamin.host', 11201), 'user', 'passwd')
```

You may define additional configuration details and thereby change the default settings by adding one of the following variables (listed below with their default setting) to the instance configuration file `<bscw-runtime-path>/conf/config.py`:

- `MAX_SMS_SIZE`
  maximum number of allowed characters. Default is `160`

- `MAY_SEND_SMS`
  limit group of users who are allowed to send SMS. Default is `None`

  ```
  MAY_SEND_SMS = None -> all users
  MAY_SEND_SMS = [ 'koch', 'hinrichs', 'paulsen' ]
  ```

- `SMS_CHARSET`
  SMS character set conversion from UTF-8 to ISO-8859-1, if `SMS_CHARSET = 'ISO-8859-1'`. Default is `''`

- `MOBILE_TRANS`
  mobile phone number translation

  ```
  MOBILE_TRANS = [
       ('^00', '+'),
       #   ('^0', '+49'),
  ]
  ```

This package is not enabled by default. To enable the BSCW *SMS* package run

```
bin/bsadmin package -e SMS
```

6.18 Sync - MS Outlook Synchronization

The package '*sync*' enables PIM synchronisation for MS-Outlook. End users may synchronise their BSCW contacts and calendars with their MS-Outlook client.

The synchronisation feature is implemented as Java applet. Thus Java is required (Java Plugin, JRE 1.6 or later).

This feature is only available for Windows (XP/Vista/Win7) Operating Systems (client-side).

Note: the synchronisation feature uses the BSCW XML-RPC API (X-API). for data exchange between the Java applet and the BSCW-server. This package therefore requires activation of the BSCW XML-RPC API. By default standard webservice calls are already allowed for registered users - unless this setting is changed in your instance configuration file (`<bscw-runtime-path>/conf/config.py`). Otherwise make sure that the configuration includes

```
ACCEPT_WEBSERVICES = 1
```

This package is enabled by default in a new BSCW server installation. No additional software installation or configuration is required on server-side. If disabled, the package may be enabled again by running

```
bin/bsadmin package -e sync
```

Note: This feature is only available in the professional edition of BSCW.

6.19 Tasks

This packages provides an optional feature for BSCW that allows users to create tasks that may be combined to ad-hoc (mini-)workflows.

The *Tasks* package is enabled by default on new BSCW servers and requires no external components. If disabled, the package may be enabled again by running

```
        bin/bsadmin package -e Tasks
```

After activation a new top-level object '*Tasklist*' is enabled at the user interface (in *'Goto'* menu/icons).

Note: This feature is only available in the professional edition of BSCW.

See BSCW Help for further details.

6.20 Timeline

This packages provides an optional feature for BSCW that allows users to view time aware object in a *Timeline* view. You may want to enable this package if you want to offer this additional functionality to your end users.

When the package is activated (in `<bscw-runtime-path>/conf/config.py`) a new menu item (and optional toolbar item) `[Goto > Timeline]` is enabled at the user interface. By selecting the `[Timeline]` action a new window opens that will show all time aware objects in the given context (i.e. the current folder, including all sub-folders) on a *Timeline*. The user may select what kind of time aware objects to show. Time aware objects especially include: *Project*, *Phase*, *Appointment*, *Task*, and *Opinion Polls* (including *Voting*, *Poll* and *Appointment Scheduling*).

See BSCW Help for further details.

This package is not enabled by default. To enable the *Timeline* package run

```
        bin/bsadmin package -e Timeline
```

6.21 WebFolder

The *WebFolder* package provides an optional feature for BSCW that allows users to create so-called Website Folders, special folders containing a website, rather similar to a Wiki system.

The "WebFolder" package is enabled by default on new BSCW servers and requires no external components. If disabled, the package may be enabled again by running

```
        bin/bsadmin package -e WebFolder
```

There is no required configuration, the configuration defaults should work on all systems. You may define additional configuration details by apending one of the following variables to the instance configuration file `<bscw-runtime-path>/conf/config.py`:

- `WF_DEFAULT_SAMPLE`
  Number (beginning with 0) of default WebFolder sample content, which is offered in "New Website Folder". A usual BSCW server comes with four sample contents: "basic" (0), "project" (1), "faq" (2) and "demo" (3). It is also possible to extend the offered sample contents. Please contact the BSCW support for detailed information.

- `WF_DEFAULT_DESIGN`
  Number (beginning with 0) of the default WebFolder design, which is selectable in "New Website Folder". An off-shelve BSCW server has four designs built-in: Tree navigation (0),

Query navigation (1), Tree in orange color (2) and Query in orange color (3). If you wish to add more designs, please contact the BSCW support.

- `WF_MAX_VERSIONS`
  Specifies the predefined setting for auto-versioning in Website Folders. Possible values:

`1`	New documents are not set under version control.
`0`	New documents are automatically set under version control and all revised versions will be stored.
`-1`	Use global (server-wide) MAX_VERSIONS setting.
`>1`	New documents are automatically set under version control, but only the given number of latest versions will be kept. Saving a new version will remove the oldest version if the limit has been reached. The default setting is to keep 10 versions.

- `WF_DEFAULT_TEMPLATE_DOC`
  Name of the default layout page, as offered in "New Layout Page". The layout pages 'treetemplate' and 'querytemplate' are part of any standard BSCW server and implement different navigation types.

- `WF_DEFAULT_TEMPLATE_DOC_NAME`
  Default name for new layout pages inside of BSCW. Note that the page itself might contain information about a different name, which is used at higher priority.

- `WF_DEFAULT_STYLE_DOC`
  Name of the default style definition, as offered in "New Style Definition". Pre-defined style definitions are 'defaultstyle' and 'orangestyle'.

- `WF_DEFAULT_STYLE_DOC_NAME`
  Default name of new style definitions inside of BSCW.

- `WF_DEFAULT_TEMPLATE_FOLDER_NAME`
  Default name of the template folder inside of Website Folders. Template folders are optional, but useful to hold templates for empty pages or other often-usedpage types.

For end-user help concerning Website Folders, see the BSCW Help

6.22 Wsmap

The package '*wsmap*' provides an optional feature for BSCW that allows users to visualize their shared workspaces in a graph view using the "*Constellation Roamer*" by Asterisq (`http://asterisq.com`).

You may want to enable this package if you want to offer this additional functionality to your end users. To enable the BSCW *wsmap* package run

```
        bin/bsadmin package -e wsmap
```

Once the package is activated a new menu item (and optional toolbar item) [`Workspace Map`] is enabled at the user interface in the`[Goto]` menu. By selecting the `[Workspace Map]` action a new window opens that will show graph of shared workspaces and members. The graph shows all shared workspaces of the user, highlighting the current folder. For each shared workspace the members are shown in the graph (with links going from folders to the respective members).

This feature is available on all platforms and in all editions of BSCW and contains a free version of the "*Constellation Roamer*" graph visualisation toolkit (with generous permission by Asterisq).

As the "*Constellation Roamer*" graph toolkit is Flash based, a recent version of the Adobe(tm) Flash player is required on client side. If you have the need of graph visualisation you are encouraged to purchase a license of the various "*Constellation Roamer"* toolkit editions provided by Asterisq, see

```
        http://asterisq.com/products/constellation/roamer/license
```

for details.

7 Administration of BSCW Servers

There are three methods to administer the BSCW server:

- through a HTML interface available to those users who have been registered as server administrators in the variable `SERVER_ADMINS` of the BSCW server instance configuration file `<bscw-runtime-path>/conf/config.py` (see section 5.2),
- by direct editing the configuration files described in section 5 with a text editor of your choice,
- through the `bsadmin` scripts which are available in the instance directory of the BSCW server (the `bsadmin` script may only be invoked by the user who installed the BSCW instance, e.g. the BSCW administrator user ID).

It may depend on the particular task which methods can or has to be used. For instance, the initial set-up of the BSCW server requires editing the BSCW instance configuration file `<bscw-runtime-path>/conf/config.py` with a text editor. If the server is running, further modifications of the configuration file may either be done by direct editing or through the HTML interface. Administration tasks such as removing or adding users require a running server and can only be done through the HTML interface or with the `bsadmin` scripts. Starting or stopping the server can only be done with the `bsadmin` script.

In general, it is recommended to use the HTML interface after the BSCW server has been installed successfully and started with the `bsadmin` script since it provides all the functionality which is needed for a system administrator.

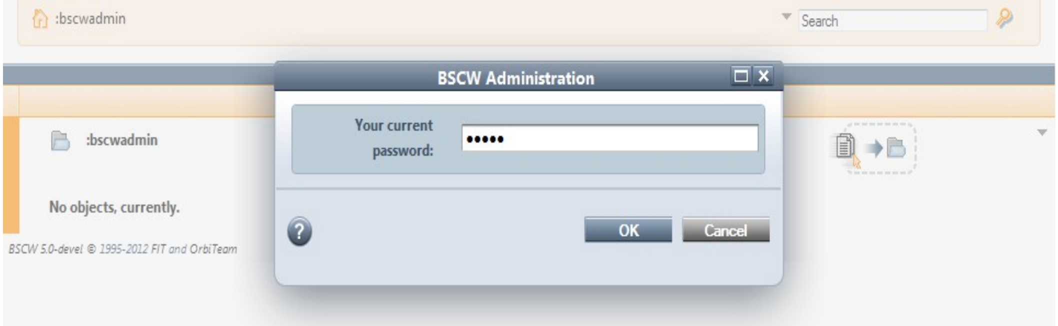
Please note that **a server administrator needs to understand what s/he is doing**. Any actions carried out by the server administrator may destroy data or may even damage the BSCW server instance.

As a server administrator you are also responsible for other measures against loss of data. Please remind to set up the BSCW **daily garbage collection**. It is urgently recommended to **install a regular back-up procedure** for the data of the BSCW server, e.g., to recover in case of hardware or software crashes. In particular, it is highly recommended to make a back-up of the system, including the configuration files, if you want to make modifications to the system through the administrator tools described in the following sections.

7.1 Administration using the Web Interface

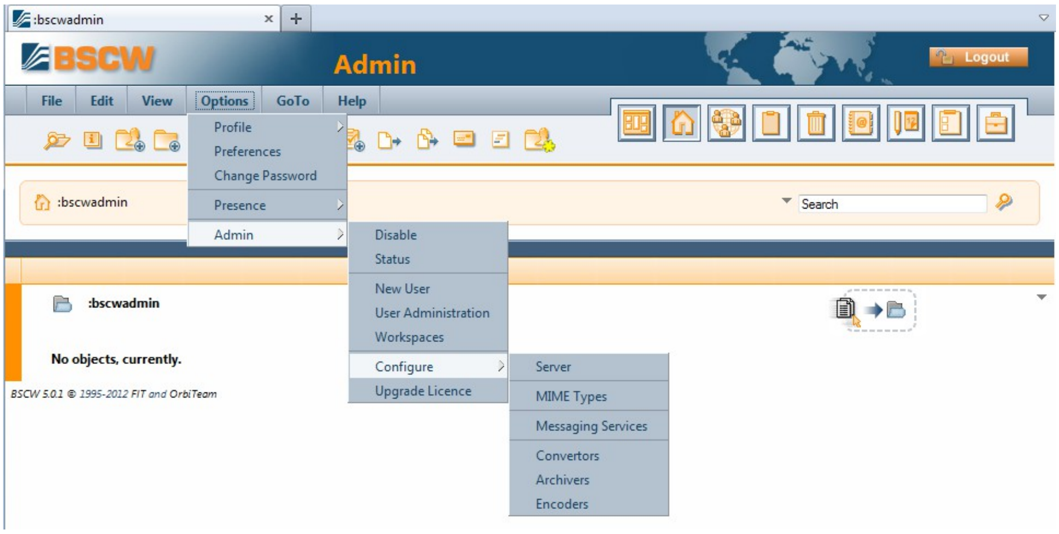
A running server can be administered using the web interface. (Note most administration tasks do not need to shut down the server; some even require a running server.) If you are registered as a BSCW administrator in the variable `SERVER_ADMINS` of the BSCW instance configuration file `<bscw-runtime-path>/conf/config.py`, you will find the action `[Admin]` in the `[Options]` menu.

Administrator users explicitly need to log in a second time with their password at `[ Options > Admin ]` to gain BSCW administrator rights. Without this additional administrator authentication no



Your current password: •••••

administrative rights are applied to their account. After successful login an additional [Admin] menu is available at [Options > Admin] and the administrator status is indicated by a "**Admin**" label at top of the BSCW user interface.

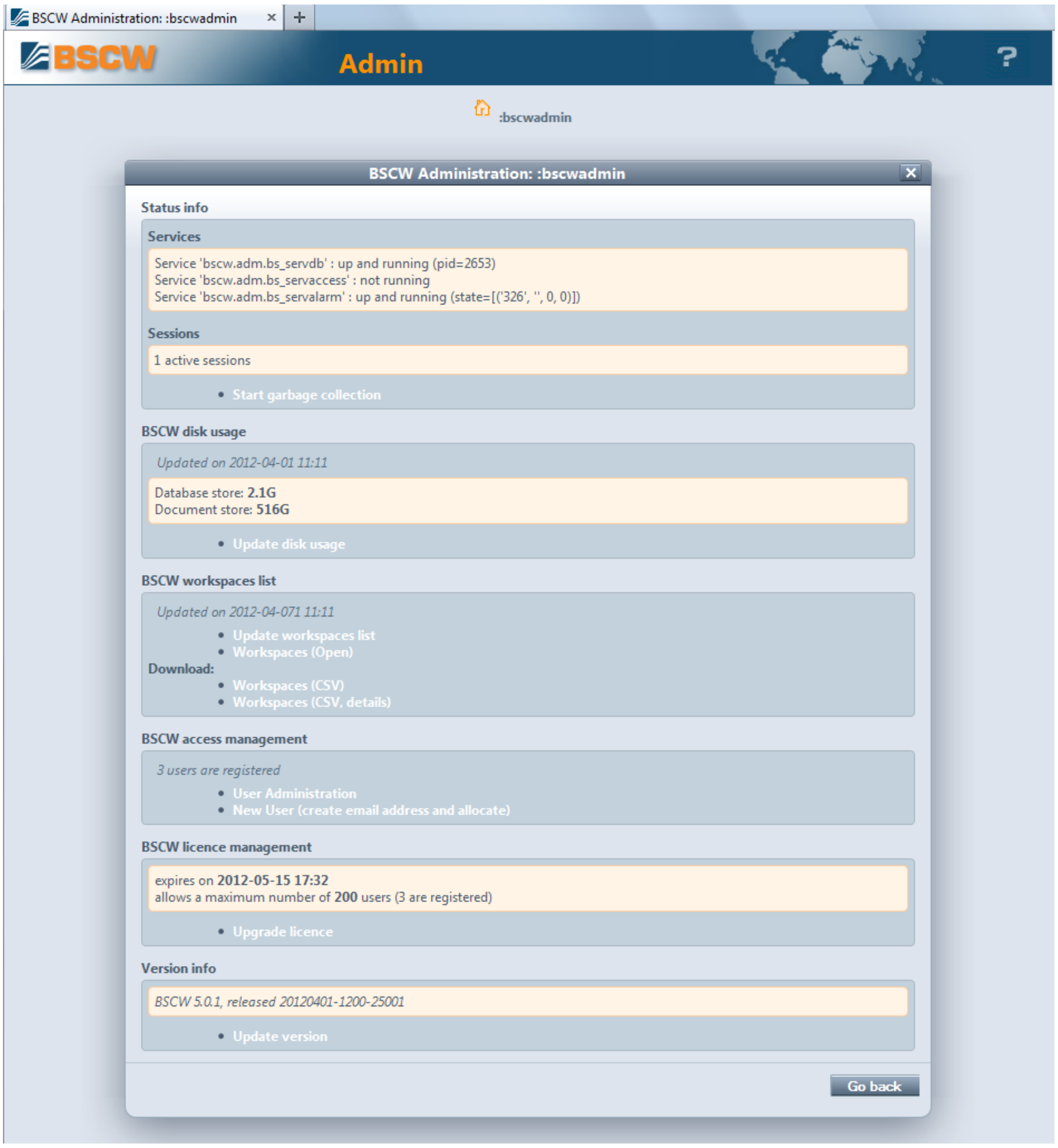


Using the administrative menu allows to perform different administrative tasks. The [Admin] menu contains the following entries

- the [Disable] entry disables the administrative rights of the current user again,
- the [Status] entry displays the BSCW status page,
- the [User administration] and the [New User] entries provide BSCW user access management functionality to search, modify, create or delete new users,
- the [Workspaces] entry display a table of all existing shared workspaces
- the [Configure] menu entry allows BSCW server configuration via the web interface,
- the [Upgrade license] entry summarizes the BSCW license management and provides functionality to apply for a new license by contacting the OrbiTeam license service.

7.1.1 BSCW status page

The BSCW status pages provides an overview about the BSCW management functions and lets you perform all major administrative tasks.



BSCW Administration: :bscwadmin

Status info

Services

Service 'bscw.adm.bs_servdb' : up and running (pid=2653)
Service 'bscw.adm.bs_servaccess' : not running
Service 'bscw.adm.bs_servalarm' : up and running (state=[('326', '', 0, 0)])

Sessions

1 active sessions

- Start garbage collection

BSCW disk usage

Updated on 2012-04-01 11:11

Database store: **2.1G**
Document store: **516G**

- Update disk usage

BSCW workspaces list

Updated on 2012-04-071 11:11

- Update workspaces list
- Workspaces (Open)

Download:

- Workspaces (CSV)
- Workspaces (CSV, details)

BSCW access management

3 users are registered

- User Administration
- New User (create email address and allocate)

BSCW licence management

expires on **2012-05-15 17:32**
allows a maximum number of **200** users (3 are registered)

- Upgrade licence

Version info

BSCW 5.0.1, released 20120401-1200-25001

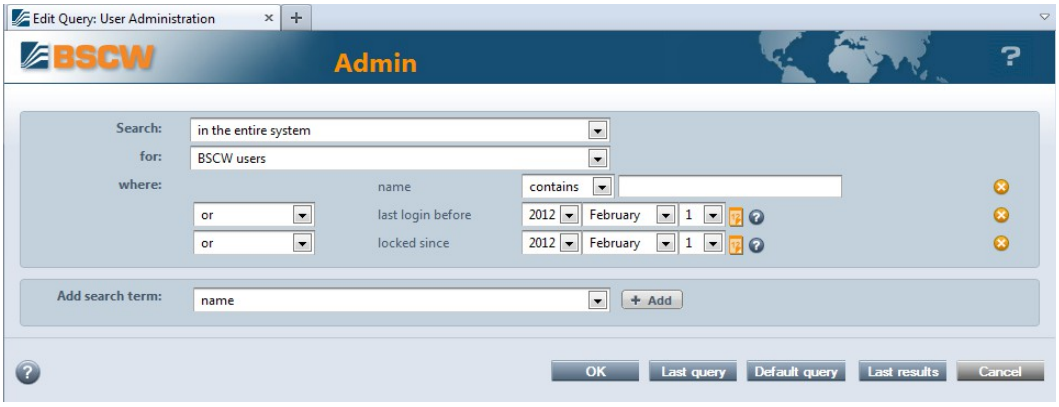
- Update version

Go back

If you click on `[Start garbage collection]` the garbage collector is started, which will become necessary if you have downloaded a new license and want to install it. `[Update disk usage]` calculates the used disk resources of the BSCW database server. `[Update workspace list]` allows to update an overview of all existing workspaces. You can immediately browse this workspace list by clicking on `[Workspaces (Open)]` or download it as comma separated list file by choosing `[Workspaces (CSV)]` resp. `[Workspaces (CSV, details)]`, `[User administration]` or `[New User]` allows to search, modify, create or delete user accounts. Using `[Upgrade license]` form allows to perform license upgrades. Finally `[Update version]` will open a link to the BSCW download web site.

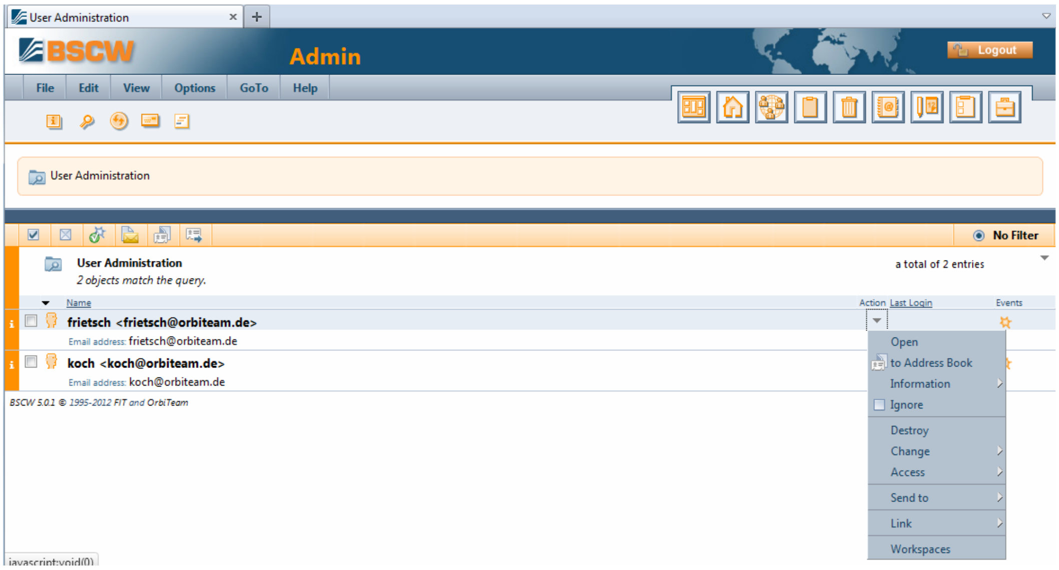
7.1.2 BSCW Access Management

The `[User administration]` and the `[New User]` entries provide BSCW user access management functionality to search, modify, create or delete users. Clicking on `[User administration]` shows the following form to search for registered users of the system:



Search: in the entire system
for: BSCW users
where: name contains
or last login before 2012 February 1
or locked since 2012 February 1
Add search term: name

After specifying a query and submitting it, the system will present a list of one or more users (in case the query matched registered users). Note the search can be restricted to particular attributes, e.g. as shown above to user *name*, *last login before* or *locked since* dates. The result of a query may look as follows:



User Administration

User Administration
2 objects match the query.
a total of 2 entries

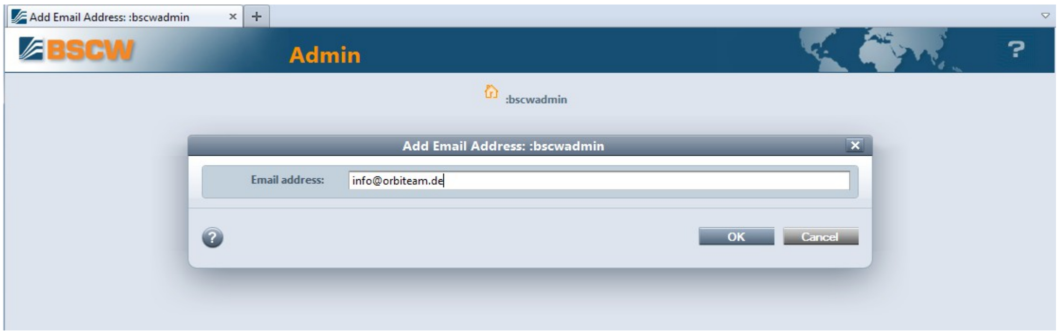
frietsch <frietsch@orbiteam.de>
Email address: frietsch@orbiteam.de

koch <koch@orbiteam.de>
Email address: koch@orbiteam.de

BSCW 5.0.1 © 1995-2012 FIT and OrbiTeam

Using the action menu provided on each result entry, the BSCW administrator may remove users from the system, rename users, edit user properties such as language preferences, change the users' password or access rights.

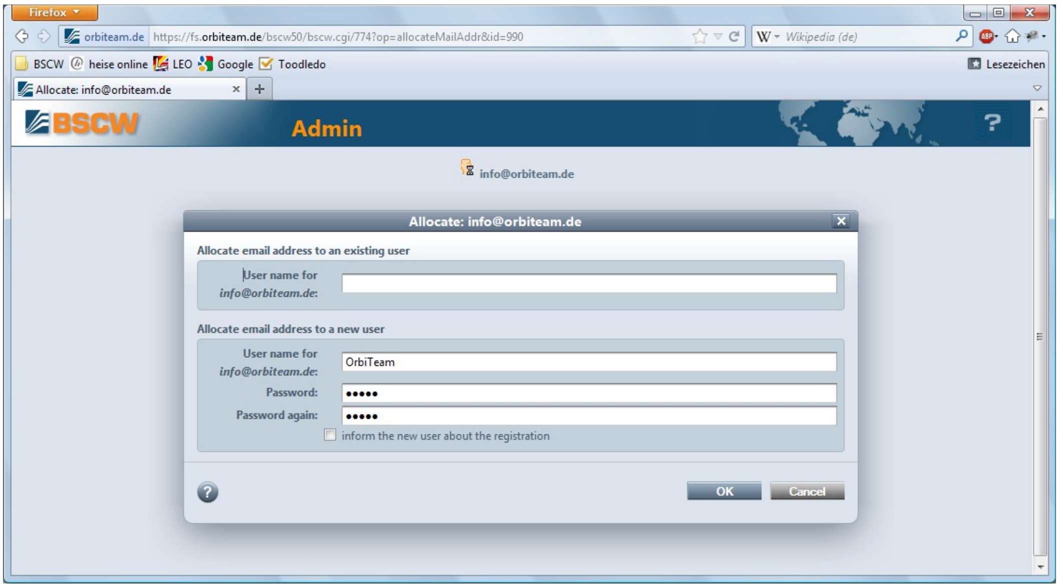
The creation of a new user is performed in two steps. In a first step the BSCW administrator creates a new email address which is afterward allocated to a (new) user (see also section 7.3). Clicking on `[New User]` shows the following form, which is used to add a new email address to the system



Add Email Address: :bscwadmin

Email address: info@orbiteam.de

After entering an email address and clicking on `[OK]` the new email address must be allocated to a (new) user, which is done within the next form:



Allocate: info@orbiteam.de

Allocate email address to an existing user

User name for *info@orbiteam.de*:

Allocate email address to a new user

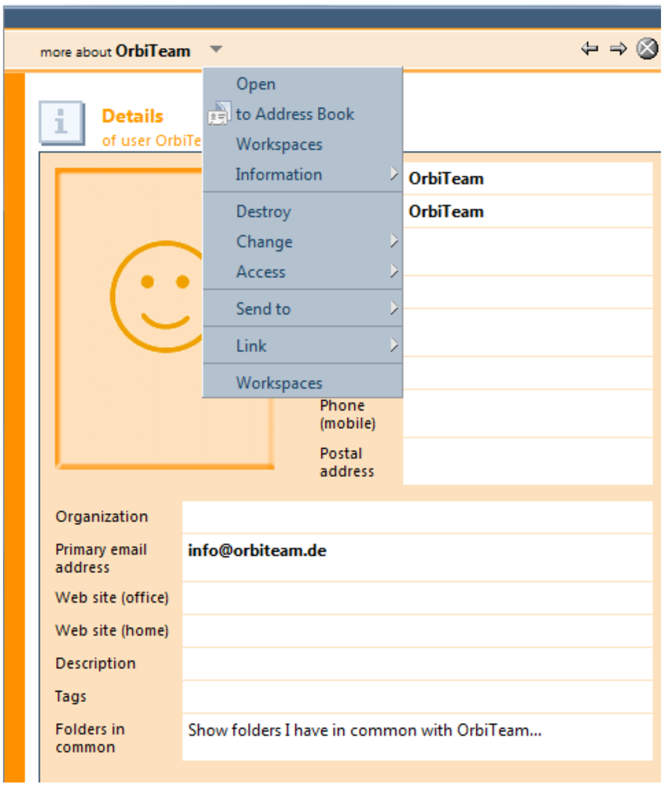
User name for *info@orbiteam.de*: OrbiTeam

Password: •••••

Password again: •••••

□ inform the new user about the registration

Finally the info page of the entered user is shown:



more about OrbiTeam

Details of user OrbiTe

OrbiTeam

OrbiTeam

Phone (mobile)

Postal address

Organization

Primary email address: info@orbiteam.de

Web site (office)

Web site (home)

Description

Tags

Folders in common: Show folders I have in common with OrbiTeam...

The administrator may repeat the allocation of the email address to another registered user, or set the email address to “bounced” status (see section 7.3).

7.1.3 Configuration menu

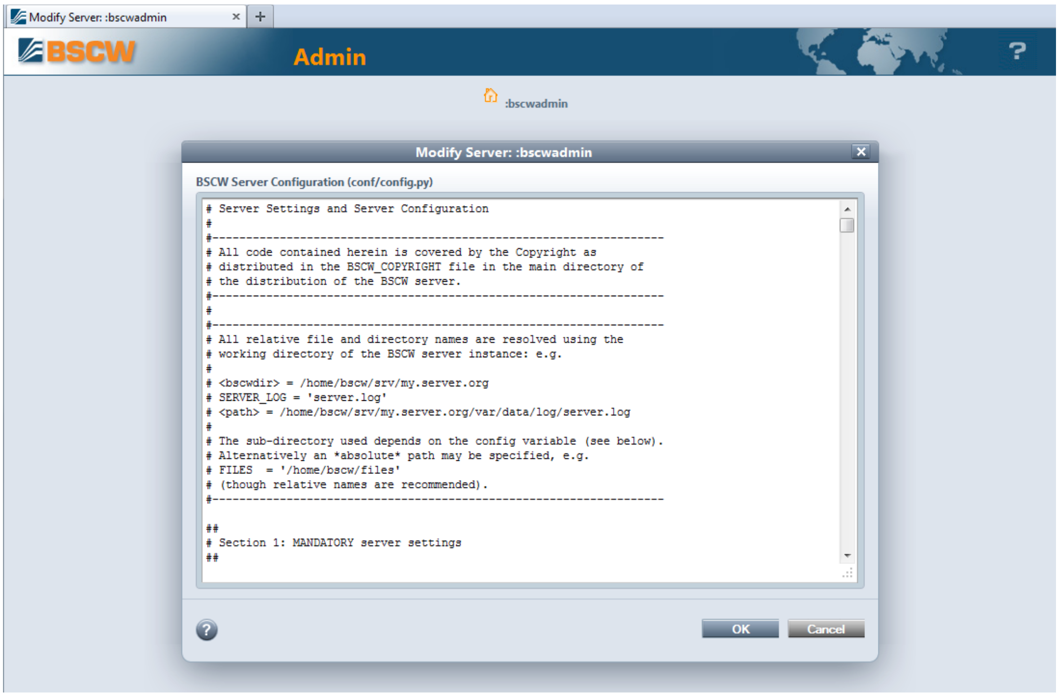
The configuration menu allows the BSCW configuration via the web interface. The entries

- Server
- Converter, Archivers, Encoders

- MIME Types
- Synchronous Tools
- Messaging Services, On-line Directories

are related to the configuration files described in section 5 *Configuration of the BSCW Server*. Opening an entry will display a form to modify the corresponding configuration file.

Clicking on `[Server]` will open the following form which allows to edit the BSCW instance configuration file `<bscw-runtime-path>/conf/config.py`. Changes to the configuration are submitted to the BSCW server by pressing the `[OK]` button.



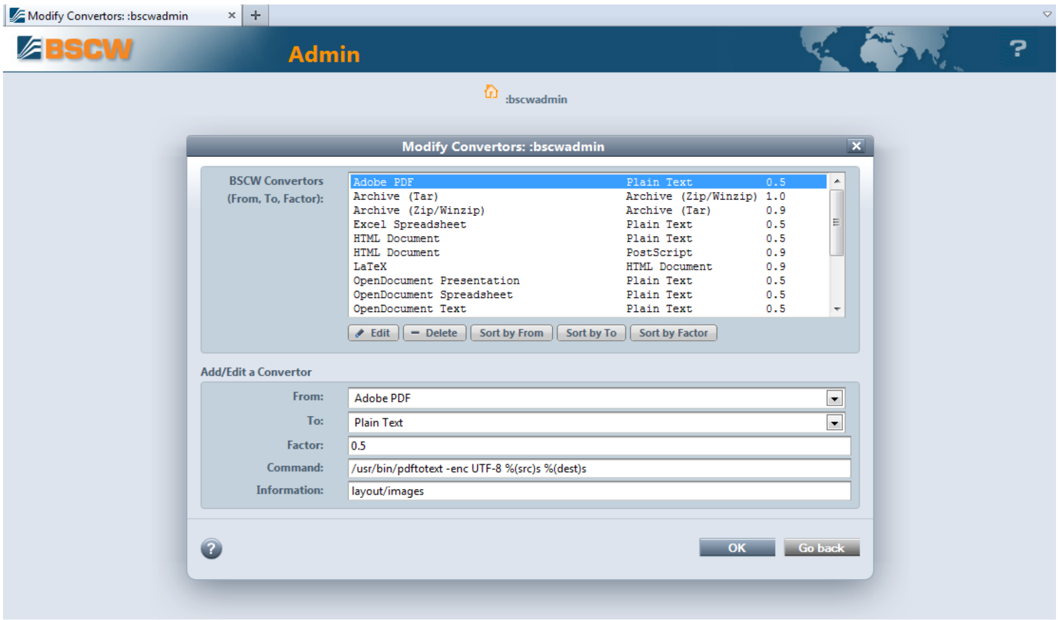
BSCW Server Configuration (conf/config.py)

```
# Server Settings and Server Configuration
#
#-------------------------------------------------------------------
# All code contained herein is covered by the Copyright as
# distributed in the BSCW_COPYRIGHT file in the main directory of
# the distribution of the BSCW server.
#-------------------------------------------------------------------
#
#-------------------------------------------------------------------
# All relative file and directory names are resolved using the
# working directory of the BSCW server instance: e.g.
#
# <bscwdir> = /home/bscw/srv/my.server.org
# SERVER_LOG = 'server.log'
# <path> = /home/bscw/srv/my.server.org/var/data/log/server.log
#
# The sub-directory used depends on the config variable (see below).
# Alternatively an *absolute* path may be specified, e.g.
# FILES  = '/home/bscw/files'
# (though relative names are recommended).
#-------------------------------------------------------------------

##
# Section 1: MANDATORY server settings
##
```

Note: Be careful when editing the BSCW instance configuration. Configuration errors may lead immediately to an dysfunctional BSCW instance.

Clicking on `[Converters]`, `[Archivers]` or `[Encoders]` will show the a form to configure existing converters and archiver or encoder commands:



Modify Convertors: :bscwadmin

BSCW Convertors (From, To, Factor):

```
Adobe PDF                          Plain Text            0.5
Archive (Tar)                      Archive (Zip/Winzip)  1.0
Archive (Zip/Winzip)               Archive (Tar)         0.9
Excel Spreadsheet                  Plain Text            0.5
HTML Document                      Plain Text            0.5
HTML Document                      PostScript            0.9
LaTeX                              HTML Document         0.9
OpenDocument Presentation          Plain Text            0.5
OpenDocument Spreadsheet           Plain Text            0.5
OpenDocument Text                  Plain Text            0.5
```

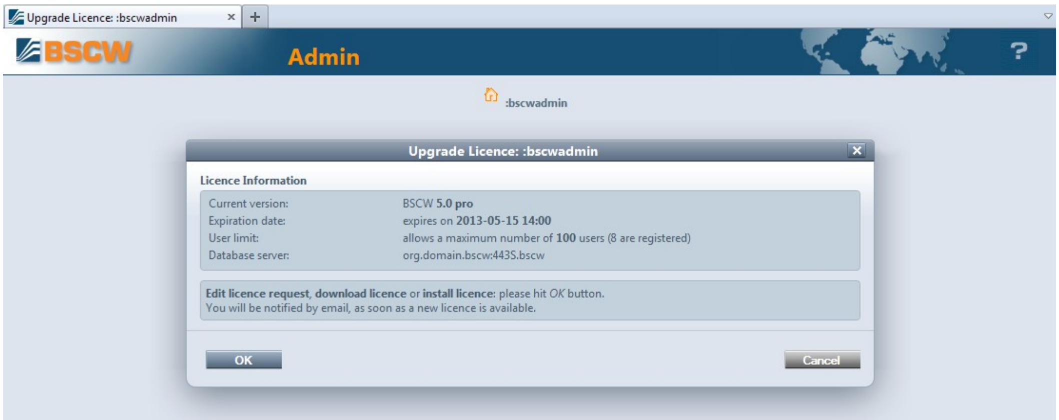
Add/Edit a Convertor

From:	Adobe PDF
To:	Plain Text
Factor:	0.5
Command:	/usr/bin/pdftotext -enc UTF-8 %(src)s %(dest)s
Information:	layout/images

In the shown case the form modifies the configuration file `<bscw-runtime-path>/conf/config_convert.py`, in particular to add new conversion tools to the BSCW server (see section 5.6, which provides a method to automatically locate required conversion commands on the BSCW server system).

7.1.4 BSCW license management

The BSCW license management allows to apply for a license resp. to prolong an expired license and to install a granted license To apply for a license open the *Upgrade License* form by clicking the `[Upgrade License]` button:



Upgrade Licence: :bscwadmin

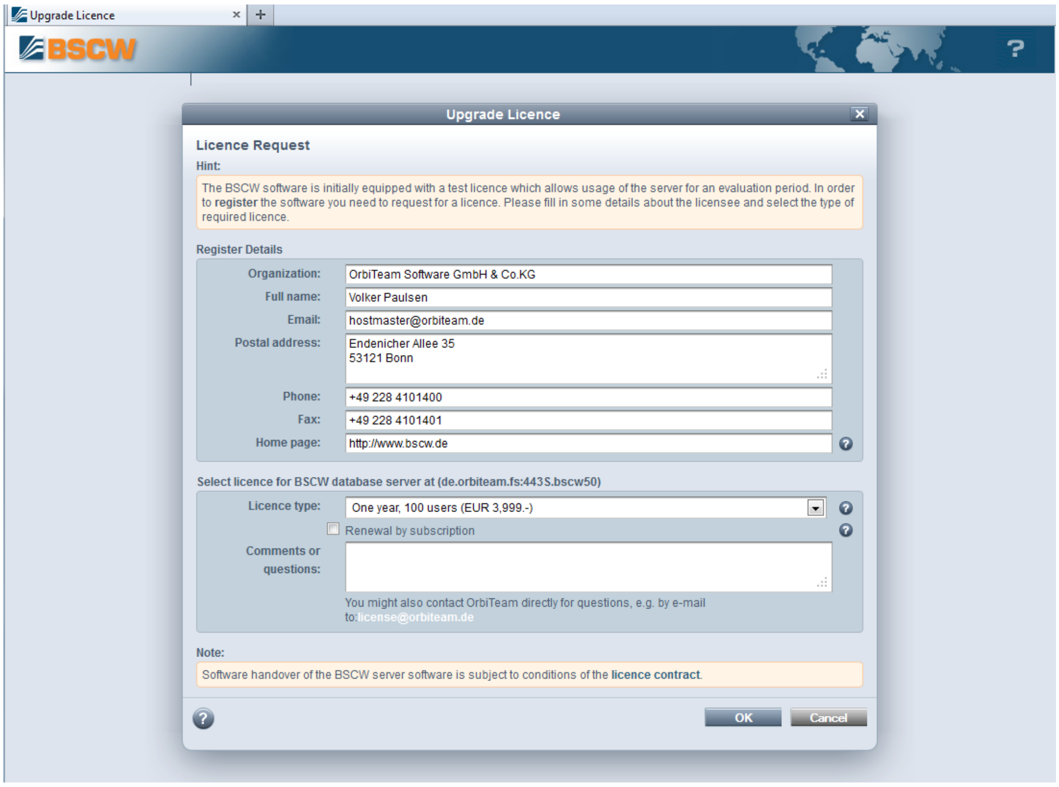
Licence Information

Current version:	BSCW **5.0 pro**
Expiration date:	expires on **2013-05-15 14:00**
User limit:	allows a maximum number of **100** users (8 are registered)
Database server:	org.domain.bscw:443S.bscw

Edit licence request, **download licence** or **install licence**: please hit *OK* button.
You will be notified by email, as soon as a new licence is available.

Next press `[OK]` which allows to edit a license request, to download or to install a license:

- When applying for a license please fill in the license request form and press `[OK]` to submit the license request to OrbiTeam. Further details about the license acquisition process are given in section 9 .



Upgrade Licence

Licence Request

Hint:

The BSCW software is initially equipped with a test licence which allows usage of the server for an evaluation period. In order to **register** the software you need to request for a licence. Please fill in some details about the licensee and select the type of required licence.

Register Details

Organization: OrbiTeam Software GmbH & Co.KG
Full name: Volker Paulsen
Email: hostmaster@orbiteam.de
Postal address: Endenicher Allee 35
53121 Bonn
Phone: +49 228 4101400
Fax: +49 228 4101401
Home page: http://www.bscw.de

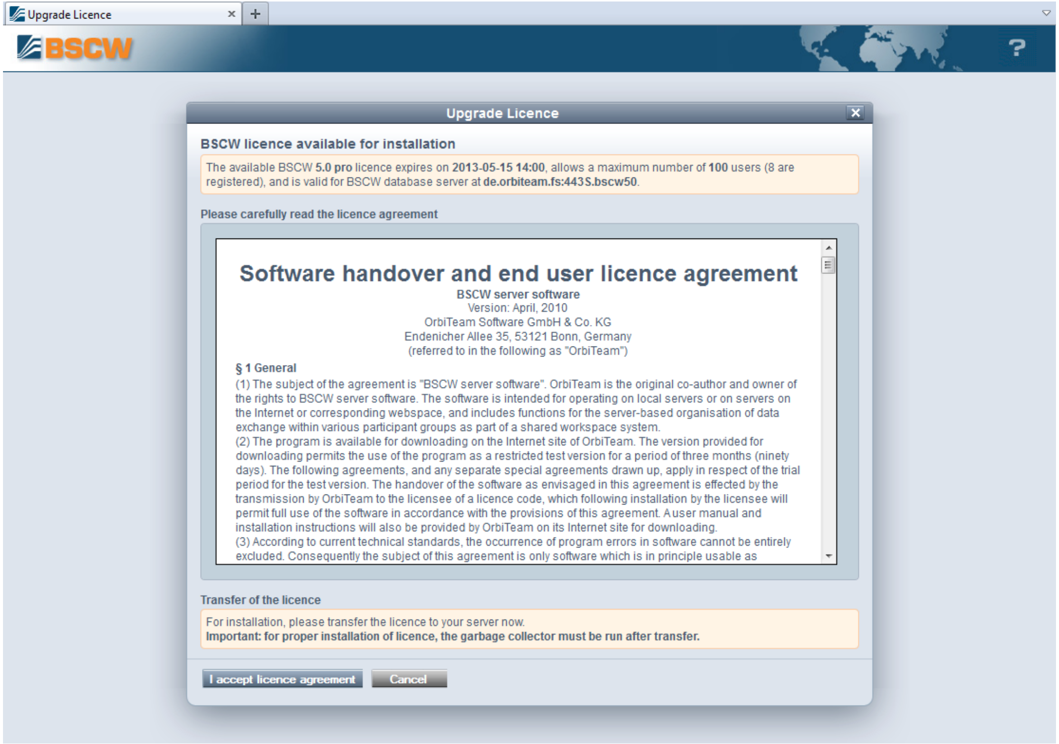
Select licence for BSCW database server at (de.orbiteam.fs:443S.bscw50)

Licence type: One year, 100 users (EUR 3,999.-)
□ Renewal by subscription
Comments or questions:
You might also contact OrbiTeam directly for questions, e.g. by e-mail to: license@orbiteam.de

Note:

Software handover of the BSCW server software is subject to conditions of the **licence contract**.

- When downloading a granted license you have to accept the license agreement



Upgrade Licence

BSCW licence available for installation

The available BSCW **5.0 pro** licence expires on **2013-05-15 14:00**, allows a maximum number of **100** users (8 are registered), and is valid for BSCW database server at **de.orbiteam.fs:443S.bscw50**.

Please carefully read the licence agreement

Software handover and end user licence agreement

BSCW server software
Version: April, 2010
OrbiTeam Software GmbH & Co. KG
Endenicher Allee 35, 53121 Bonn, Germany
(referred to in the following as "OrbiTeam")

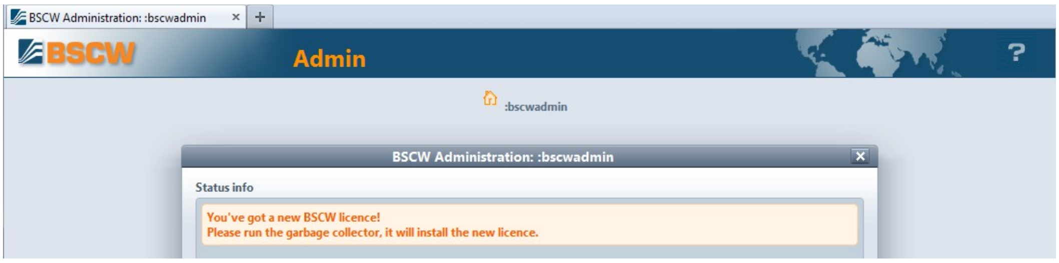
§ 1 General

(1) The subject of the agreement is "BSCW server software". OrbiTeam is the original co-author and owner of the rights to BSCW server software. The software is intended for operating on local servers or on servers on the Internet or corresponding webspace, and includes functions for the server-based organisation of data exchange within various participant groups as part of a shared workspace system.
(2) The program is available for downloading on the Internet site of OrbiTeam. The version provided for downloading permits the use of the program as a restricted test version for a period of three months (ninety days). The following agreements, and any separate special agreements drawn up, apply in respect of the trial period for the test version. The handover of the software as envisaged in this agreement is effected by the transmission by OrbiTeam to the licensee of a licence code, which following installation by the licensee will permit full use of the software in accordance with the provisions of this agreement. A user manual and installation instructions will also be provided by OrbiTeam on its Internet site for downloading.
(3) According to current technical standards, the occurrence of program errors in software cannot be entirely excluded. Consequently the subject of this agreement is only software which is in principle usable as

Transfer of the licence

For installation, please transfer the licence to your server now.
Important: for proper installation of licence, the garbage collector must be run after transfer.

and run a garbage collection by pressing `[Start garbage collection]` on the BSCW status page:


BSCW Administration: :bscwadmin
BSCW
Admin
:bscwadmin
BSCW Administration: :bscwadmin
Status info

You've got a new BSCW licence!
Please run the garbage collector, it will install the new licence.

7.2 Administration using the `bsadmin` script

The `bsadmin` script constitutes the central access point to the BSCW instance from the command line. Starting with BSCW 5 it is located in the bin directory of each BSCW instance: `<bscw-runtime-path>/bin/bsadmin`.

The primary purpose of the `bsadmin` script is starting and stopping the BSCW server, starting the garbage collector and executing the workspace report function. The garbage collector is normally triggered on a regular basis, e.g., by a cron job on Unix systems or by the task scheduler on Windows 7, 8/Server 2012, 2008, 2003 systems. Therefore the normal usage of the `bsadmin` script is only as:

```
bin/sadmin start
bin/sadmin stop
bin/sadmin garbage
```

In addition, the script can be used for a number of administration functions.

For historical reasons the `bsadmin` script contains also a number of functions which can (and should!) be carried out through the HTML interface. Furthermore, it provides features which are used during software development of the BSCW server software, e.g., for debugging purposes. Since these functions are only useful for the BSCW software developers but not for the normal BSCW server administrators, they are not explained below.

The commands marked with (I) are normally used during installation only and are invoked automatically. The commands marked with (D) provide debugging information (do not use without advice from support@orbiteam.de, otherwise you may damage your database).

When using the `bsadmin` command without any arguments, it displays the list of possible arguments as follows:

`bsadmin archive`	archive an artifact via command line	
`bsadmin chkconfig`	check configuration make directories and cgi scripts	(I)
`bsadmin chkfiles`	check for missing document files	
`bsadmin chpwd`	change user password and lock/unlock user	
`bsadmin clean_anon`	remove unowned objects in “anonymous” top level folders	
`bsadmin conf_apache`	BSCW Apache web server configuration	
`bsadmin conf_iis`	BSCW IIS configuration	(1)
`bsadmin conf_tzdata`	configure timezone data	
`bsadmin create_index`	generate search index	(2)
`bsadmin dbcheck`	database check/repair	
`bsadmin dbscan`	scan database file and print offset, class name and id	(D)
`bsadmin dbsummary`	print a summary of all classes in the database	
`bsadmin du`	show/update BSCW database disk usage	
`bsadmin find`	find (recursively) documents, e.g. `:user/workspace/folder/.../doc`	
`bsadmin fsck`	check file tree for obsolete files and directories	

`bsadmin fix_keys`	remap mail address and user keys	
`bsadmin garbage`	BSCW garbage collector	
`bsadmin getconfig`	get configuration info from `config.py`	
`bsadmin index_page`	generates an index page for the script directories	
`bsadmin ldapbind`	change user LDAP binding(s)	
`bsadmin ldapupdate`	synchronize BSCW users with LDAP	
`bsadmin ldif`	export users to LDIF format	
`bsadmin license`	request a new license, check license details or warn about license expiry	
`bsadmin listmeta`	export metadata as CSV list	
`bsadmin listws`	list (shared) workspaces, update workspace list	
`bsadmin ls`	list documents given by file path	
`bsadmin mailaccess`	list all folders w/ enabled mail access	
`bsadmin manage_servers`	manage BSCW servers machine-wide	
`bsadmin members`	add or remove users from workspaces	
`bsadmin mkfolder`	creates folders	
`bsadmin oauth`	list oauth consumers	
`bsadmin openid`	list openids	
`bsadmin package`	(un)install a BSCW package	
`bsadmin prtactions`	print all defined actions	
`bsadmin quota`	user disk quotas commands	
`bsadmin register`	registration of email addresses and new users	
`bsadmin rename`	rename an user	
`bsadmin report`	modify report configuration	
`bsadmin rmevents`	remove (dequeue) all events older than n days	
`bsadmin rmobj`	remove BSCW folders/documents given by ID or filepath	
`bsadmin rmuser`	remove an user	
`bsadmin rmwaste`	remove objects from waste baskets (resp. clipboards)	
`bsadmin roles`	add, edit or assign roles	
`bsadmin search`	query PyLucene index	(2)
`bsadmin sendmail`	BSCW mailer	(D)
`bsadmin servuno`	BSCW user notification service debugging	(D)
`bsadmin service`	manage Windows NT platform BSCW service	(2)
`bsadmin start`	start BSCW instance servers	
`bsadmin stop`	stop BSCW instance servers	
`bsadmin syncf`	synchronizes BSCW folder with file system directories	
`bsadmin themes`	generate the CSS files needed for the BSCW themes	
`bsadmin update_defaults`	update configuration files with new defaults	(I)
`bsadmin update_helper`	update resource files for desktop widgets and uploader	(I)
`bsadmin users`	list users and mail addresses	
`bsadmin versions`	list/remove versions from document version stores	
`bsadmin wstat`	print workspace statistic	
`bsadmin xml_tmpl`	generate cached XHTML files for all XML templates	

(1) only on Windows 7, 8/Server 2012, 2008, 2003

(2) only if a content search package `PyLucIndex` is installed

(I) required during installation

(D) for debugging only

7.3 User administration

The BSCW server can be configured to allow

- self-registration by users
- registration of new users only by the system administrator and possibly other authorized persons.

The variable `MAY_REGISTER` in the BSCW instance configuration file (`<bscw-runtime-path>/conf/config.py`, see section 5.2) specifies which of the registration modes shall apply. When self-registration is enabled, the name space of legal email addresses may be restricted by using the variable `RESTRICT_MAIL`.

Note that there are two forms of self-registration:

- an user may enter his own email address to become a newly registered user of a BSCW server.
- an already registered user may invite another person by using his or her email address.

In principle, a registered BSCW user is identified by his email address, i.e., a particular email address specifies exactly one BSCW user. Therefore, the "creation" of a new BSCW user starts with the specification of an email address, either through self-registration or by the system administrator through the administrator interface described in the preceding sections.

The specification of an email address for a user is the first step of the registration process. The second step is the allocation of a user name and password to this email address. After the first step and before the execution of the second step an email address is called *pending*.

For self-registration the BSCW server sends an email message with a "token" (the registration URL) to the specified email address that allows the execution of the second step (or the resetting of a password, see below). If the email message cannot be delivered (e.g., because the email address was wrong), the intended recipient will never receive this email and therefore cannot carry out the second step of the registration process, i.e., the email address remains pending forever. This two-step procedure ensures that email addresses of registered users are always correct, unless a user looses his or her email account later without providing a new email address. In this case the system administrator may correct wrong email addresses through the administration interface.

If an email address remains pending because email messages cannot be delivered to the given address (this may be annoying for the system administrator since he has to take care of the bounced emails) the system administrator can set such an email address to *bounced*. This has two effects: firstly, the respective address(es) will not produce any bounced emails any more since the BSCW server filters all outgoing email messages against the bounced addresses list. Secondly, the BSCW server does not allow the first step of the registration process for this email address any more.

This second effect can also be used to exclude particular persons from using a BSCW server: If the system administrator sets a particular email address to bounced, the user associated with this email address cannot re-register with the server any more using this bounced email address.

In the case of user registration through system administrators (see section. 7.1 or 7.2), they should be careful when entering the email address of new users since the verification process for the email addresses as in the case of self-registration is not carried out. Erroneous addresses would only be detected when the BSCW server sends an email message to such an address, assuming that someone takes care of bounced email messages. In the case that bounced email addresses are deleted automatically (some mail servers are configured that way), such erroneous addresses may not be detected at all. If, in the case of self-registration, a user enters an email message that is already associated with a registered user, the BSCW server acts as follows:

- If the user wants to register as a new user from the registration page, the system assumes that the respective user has forgotten his or her password. It therefore sends an email message to the email address, which allows the selection of a new password.
- If the user wants to invite another user to this server, the system assumes that the user was not aware that the other user was already registered. The system therefore replaces the invited user's email address by the invited user's login name.

7.3.1 User status with `bsadmin users`

To create reports about existing users the `bsadmin users` script provides the following options:

```
$ ./bin/bsadmin
 bsadmin users -n [-T|-E|-I] [{-o|-O} <ndays>] [-L<f>] [<u1> ... <un>]
 bsadmin users -m            [{-o|-O} <ndays>] [-L<f>] [<u1> ... <un>]
 bsadmin users -a            [{-o|-O} <ndays>] [-L<f>] [<u1> ... <un>]
 bsadmin users -p            [{-o|-O} <ndays>] [-L<f>] [<u1> ... <un>]
where:
 -n    print username(s)
 -m    print username(s) and primary mailaddress
 -a    print username(s) and all mailaddresses
 -p    print username(s) and passwords (htpasswd format)
of all or given users <u1> ... <un>; additional options:
 -T    append creation, last-access time stamps
 -E    append account-expiry, passwd-expiry time stamps
       ('=' marks user individual account/password expiry date)
 -I    append last ip address
 -L    consider locked users with lockflags <f> ::= {'a'|'l'|'s'|'e'}
       (locked by 'a' - admin, 'l' - LDAP, 's' - system, 'e' - expired)
 -o    consider users with last access before <ndays> days
 -O    consider users with creation before <ndays> days
```

7.3.2 User registration with `bsadmin register`

For the administration of users and their email addresses the `bsadmin register` script offers the following functionality:

```
$ ./bin/bsadmin
 bsadmin register <addr>                  info about email mail address
 bsadmin register -a [-o<n>]              print email addresses
 bsadmin register -b [-f<f>][-<n> [-d]]   print bounced email addresses
 bsadmin register -b <addr>               set bounced
 bsadmin register -c <addr> [<lang>]      create pending email address
 bsadmin register -d <addr>               delete email address
 bsadmin register -i <addr>               print user / <unknown> / <pending>
 bsadmin register -n <addr> <newaddr>     rename (change address)
 bsadmin register -p [-f<f>] [-o<n> [-d]] print pending email addresses
 bsadmin register -p <addr> [<lang>]      set pending
 bsadmin register -r <addr> <user> [<pw>] register new user
 bsadmin register -u [-o<n>]              print allocated email addresses
 bsadmin register -u <addr> <user>        allocate secondary email address
 bsadmin register -U <addr> <user>        allocate primary email address

-f<f>       consider email addresses with flags <f> ::= {n|f|a}+
              n - address  w/o invitation (default)
              f - address  w/   invitation to a folder (workspace)
              a - address  w/   invitation to an appointment
            (if option '-f<f>' is omitted, the default '-fn' is assumed)
-o<n>       list email addresses with modification date before <n> days
-o<n> -d    delete email addresses with modification date before <n> days
```

For instance to create a new user use the following command

```
$ ./bin/bsadmin  register -r name@domain.org name passwd
```

7.3.3 User management with `bsadmin (rename|chpwd|rmuser)`

The administration scripts `bsadmin rename`, `bsadmin chpwd` and `bsadmin rmuser` are allow to manage BSCW user accounts via command line. The `bsadmin rename` script renames user account names an provides the following options:

```
$ ./bin/bsadmin rename
usage:
 bsadmin rename [-n] <oldname> <newname>
where:
 -n           do not send an email notification
```

The `bsadmin chpwd` script allows beside user account password changes to lock and to unlock user accounts or to maintain user account and password expiry settings as follows:

```
$./bin/bsadmin chpwd
usage:
 bsadmin chpwd [-v]                                 <user> [<pwd>]
 bsadmin chpwd [-v] {-l|-u|-e} {-n<email>|--notify} <user>
 bsadmin chpwd [-v] {-E<date>|-p}                   <user>
 bsadmin chpwd [-v] [-r] {-e|-p}                    <user>

<user> [<pwd>]  set new password for user
 -l             lock user
 -u             unlock user
 -e             expire user (lock account if user is expired)
 -n <email>     send email notification BCC to email address
 --notify       send email notification to locked user only
 -E <date>      expire user at <date> timestamp (<yyyy-mm-dd hh:mm> or <3d>)
 -p             expire password (force password renewal at next login)
 -r             reset password (-p) / account (-e) expiry
 -v             verbose
```

The `bsadmin rmuser` script removes user accounts from the BSCW system. When removing user accounts only private (non-shared) user data will be removed permanently, while data in shared workspaces is preserved. If the owner of a shared workspace is removed BSCW will determine a new owner among the remaining workspace members. In particular `bsadmin rmuser` provides the following options:

```
$ ./bin/bsadmin rmuser
usage:
 bsadmin rmuser [-n|b|v] -a [-e <charset>] [-o <nowner>] [--dry-run] <user>
 bsadmin rmuser [-n|b|v] -m <nowner>                     [--dry-run] <user>
 bsadmin rmuser [-n|b|v] [-f]                                        <user>

 -n            do not send an email notification
 -b            set user email address(es) invalid ("bounced")
 -m <nowner>   merge workspaces to <nowner>
 -a            archive users' artifacts in "var/data/rmuserarc" (zip)
  -o <nowner>  set owner of owned workspaces to <nowner> when archiving
  -e <charset> encode pathnames as <charset> (default: UTF-8)
  --dry-run    verbose output and no changes are committed
 -f            force destruction of all owned workspaces
 -v ... -vv    verbose output
```

7.3.4 Additional anonymous users

Additionally to user '`anonymous`', more anonymous users can be registered. Access to these anonymous accounts are also not controlled by authentication, but may be restricted by means of HTTP server configuration, just as in the case of user 'anonymous'. This way different levels of access control can be implemented, from unrestricted public access to anonymous intranet or even anonymous subnet access. Adding an additional anonymous user requires (in this order):

1. configuration of an access control file (e.g. 'u_intranet.txt'),
2. specification of an associated CGI path (i.e. '/intra/' in <bscw-runtime-path>/conf/config.py SCRIPTS)
3. configuration of the Web server (via bin/bsadmin conf_apache (and restart of the Web server))
4. access to the newly created server CGI path (to create the new anonymous user)

Note: Additional anonymous users may be removed in contrast to the system user 'anonymous'. The required steps in detail are as follows:

1. First you have to configure your Web server to handle restricted access to the anonymous prefix. For the Apache Web server, you would have to add:

```
Alias   /intra  .../var/www
<Location "/intra">
    # use CGI scripts
    Options ExecCGI
    AddHandler cgi-script .cgi
    # set index file
    DirectoryIndex index.html default.htm
    # client access (Apache 2.2)
    Order deny,allow
    Deny from all
    #  some dedicated IP numbers may access this directory
    Allow from 10.23.45.67
    Allow from 10.23.45.68
    # client access (Apache 2.4)
    #Require all denied
    ##some dedicated hosts (fqdn) or IP addresses may access this directory
    #Require host bscw.server.org
    #Require ip 10.23.45.67
    # ...
</Location>
```

   To automatically generate this configuration within your <bscw-runtime-path>/conf/apache{2,24}/bscw.conf file you have to create a <bscw-runtime-path>/conf/apache{2,24}/u_<username>.txt file which contains the Allow from (Apache 2.2) resp. the Require directives (Apache 2.4) for the allowed IP address ranges which may access the additional anonymous user prefix. Following the above example, create the file <bscw-runtime-path>/conf/apache2/u_intranet.txt and enter the following Allow directives (Apache 2.2)

```
Allow from 121.23.45.67
Allow from 121.23.45.89
```

   or create the file <bscw-runtime-path>/conf/apache24/u_intranet.txt and enter the following Require directives (Apache 2.4)

```
Require host bscw.server.org
Require ip 121.23.45.89
```

2. Next the creation of a new anonymous user must be accompanied by a adding a new entry in the `SCRITPS` dictionary in the central configuration specification (`<bscw-runtime-path>/conf/config.py`). Select as key a new prefix for a directory mapping in the Web server and specify a tuple of the username, the directory (not used anymore, set to `''`), the standard scripts and further scripts. For example:

```
SCRIPTS = {
     '/bscw/':   (None,         '', CREATE_SCRIPTS, SECURE_SCRIPTS),
     '/pub/':    ('anonymous', '', CREATE_SCRIPTS, SECURE_SCRIPTS),
     '/intra/':  ('intranet',  '', CREATE_SCRIPTS, SECURE_SCRIPTS),
}
```

   Note: when the newly entered path is accessed for the first time via the Web server URI `/intra/bscw.cgi` the specified username (e.g. `intranet`) is automatically created as anonymous user. If the username already exists and is a (non-anonymous) standard user a "`Bad script name`" error will be raised.

3. Now execute the `bsadmin conf_apache` command to generate a new `<bscw-runtime-path>/conf/apache{2,24}/bscw.conf` file and restart your Apache HTTP server.
4. After these steps have been carried out, create the workspaces of the newly created anonymous user by accessing the the URL:

```
http://<server>/intra/bscw.cgi
```

7.4 Asynchronous Services

7.4.1 User Notification Services (UNO)

The user notification services (UNO) perform the following tasks (depending on the configuration settings in the BSCW instance configuration file `<bscw-runtime-path>/conf/config.py`):

- sending periodical workspace activity reports via email to give the users an overview about recent activities in a specific time period (e.g. daily)
- sending direct email notifications to inform the users about recent events

Using the user notification services a BSCW user does not need to contact its BSCW server(s) so often to check for new events. If the user notification services are activated, the users' event preference page provides a column for subscription of the "*Daily Report*" or the "*Direct Email*" notification (depending on the UNO service configuration).

To activate the user notification services the BSCW administrator has to start the additional UNO server (`bscw.adm.bs_servuno`) in the `SERVERS` list in `<bscw-runtime-path>/conf/config.py`:

```
SERVERS = [
    ('UnoSocket', 'bscw.adm.bs_servuno'),
]
```

The UNO server sends HTTP requests to the BSCW server using a (virtual) web server on `localhost:HTTP_LOCAL_PORT` (default `localhost:80`). If your web server is not listening to localhost you may need to define an additional (virtual) web server running on `localhost:HTTP_LOCAL_PORT` (for Apache HTTP server configuration hints see section 3.3.1 (Unix) or section 4.4.2 (Windows)).

Note: This setting will start and stop the UNO server automatically with the BSCW database server.

The following variables have to be set for the configuration of the user notification services:

- `SERV_UNO_STATE` defines a file name for saving the state of the UNO service. The file is written, when the UNO is stopped and read when the server is started again.
- `SERV_UNO_TIMES` contains a dictionary of fine tuning parameters for the UNO service; for details see the server instance configuration file `config.py`.
- `WSREPORT = 1` enables the daily workspace report
- `WSREPORT_DIRECT = 1` enables the direct email notification
- the default subscription for all users is defined by the `AUTOSUBSCRIBE_REPORT` flag. By default no report is sent to new users, each user may decide to subscribe to the workspace report by her/himself. The server administrator can change this behavior by setting the flag

  ```
  AUTOSUBSCRIBE_REPORT = 1
  ```

  If this is enabled new users will automatically subscribed to the report service (each user may then unsubscribe from the service).
- `DEFAULT_EVENTMASK` defines the event type subscription mask, with the values

  ```
  read = 1; create = 2; move = 4; change = 8
  ```

  By default all event types are subscribed (`read + create + move + change = 15`).
- `REPORTLOG` points to a file in which a protocol about the reports is logged, for example:

  ```
  REPORTLOG = 'report.log'
  ```

BSCW provides a real-time awareness service (AWS), which basically allows users to receive notifications about BSCW events in real-time, i.e. when they occur in the system. Moreover it allows users to see who (of their collaborators) is currently *on-line*. The AWS is provided through the BSCW Monitor Applet (*JMonitor*) which can be launched from context menu on users' BSCW home folder. To enable the AWS set the flag

```
AW_MONITOR_ENABLED = 1
```

The Monitor Applet requires the BSCW webservices API (XML-RPC). Setting `AW_MONITOR_ENABLED` to `1` **mandatory** requires to enable the BSCW webservices by setting `ACCEPT_WEBSERVICES = 1`

Note: See the `User Notification Services (UNO)` section in your instance configuration file for more details.

7.4.2 User account expiry

To configure an inactivity time interval after which users are expired, the administrator has to define in the BSCW instance main configuration files `<bscw-runtime-path>/conf/config.py` the `EXPACCT` directive. E.g configure the following value:

```
EXPACCT = '90d'
```

In this case user accounts are expired, after 90 days without login.

To automatically check user account expires, the "`expire.sh`" shell script must be periodically (e.g. daily or weekly) executed:

- copy the script from the BSCW distribution directory `<bscw-dir>/lib/bscw-5.0.9-3????-py26/etc/bin/expire.sh` to the desired destination (e.g. the `<bscw-runtime-path>/bin` directory), e.g.

  ```
  $ cd /opt/bscw
  $ cp ./lib/bscw-5.0.9-3????-py26/etc/bin/expire.sh./srv/bscw.domain.org/bin
  ```
- edit the script `<bscw-runtime-path>/bin/expire.sh` and set the variable `BSCW_WRK_DIR` to the path of the BSCW instance ("`<bscw-runtime-path>`").

Create a "crontab" entry as follows:

```
        $ crontab -e
        10 1 * * * <bscw-runtime-path>/bin/expire.sh \
                                    >> <bscw-runtime-path>/var/log/expired.log
```

Notes:

- If no automatic expiry check (and user account locking) via `expire.sh` is run periodically, user accounts will only expire (and locked) if the user explicitly logs in after the in `EXPACCT` defined period. This may result into an incomplete listing of inactive (and locked) accounts in the user management.

7.4.3 Automatic disk usage update

To provide an automated update of the storage usage of the BSCW database and uploaded documents displayed on the administrator BSCW status page (see section 7.1.1) the '`bsadmin du`' script (disk usage) is available:

```
./bin/bsadmin du -h
usage:
    bsadmin du                 show current BSCW database disk usage
    bsadmin du [-u]            update (re-calculate) BSCW database disk usage
```

To periodically re-calculate the storage usage configure the following "crontab" entry:

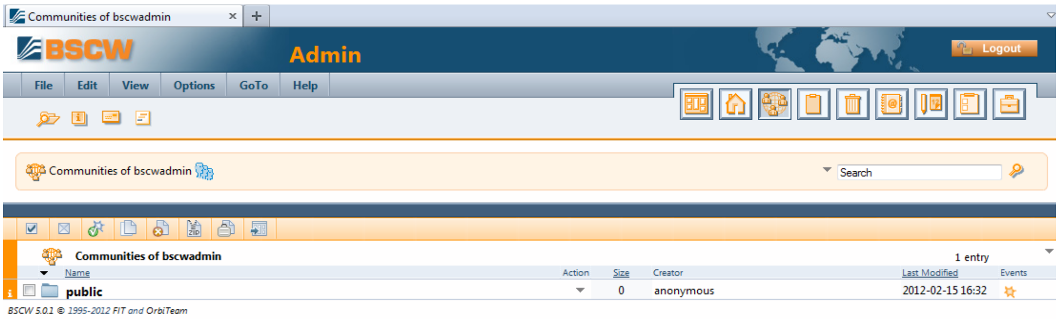
```
$ crontab -e
 15 2 * * * <bscw-runtime-path>/bin/bsadmin du -u
```

7.5 Public space deactivation

By default BSCW allows users in the “manager” role to publish the contents of a folder in a “public space”, which can be accessed by everyone over the World-Wide Web without being a registered user of the server.

To disable the public space for all users a BSCW administrator may delete the “public” folder contained in the “Communities” folder as follows:

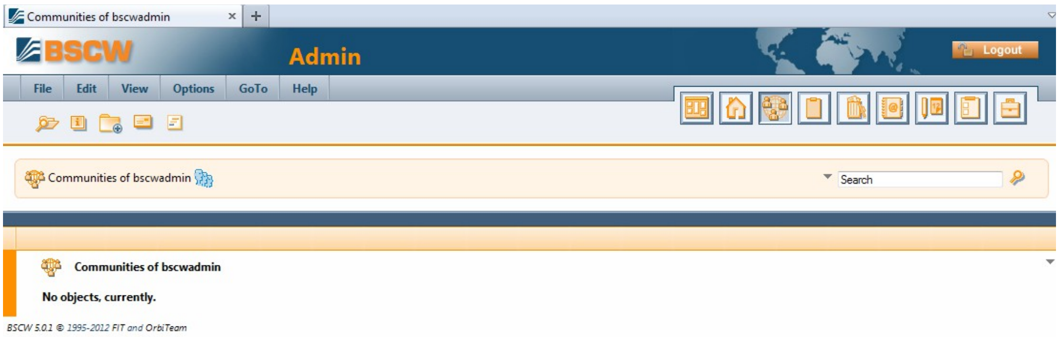
1. Log in a second time with password at `[ Options > Admin ]` to gain BSCW administrator rights. After successful login to the `[Admin]` page press `[OK]` to keep the administrator rights for your current session. The administrator status is indicated by a “Admin” label at top of the BSCW user interface.
2. Enter your “Communities” folder by clicking in the instant access bar
3. Delete the “public” folder as shown:



Communities of bscwadmin — 1 entry

Name	Action	Size	Creator	Last Modified	Events
public		0	anonymous	2012-02-15 16:32	

BSCW 5.0.1 © 1995-2012 FIT and OrbiTeam



Communities of bscwadmin

No objects, currently.

BSCW 5.0.1 © 1995-2012 FIT and OrbiTeam

Note: using the [delete] action will move the “public” folder to your trash.

To disable the “public space” it is sufficient to keep the “public” folder in your trash. If you [destroy] the “public” folder within your trash all “public spaces” are removed by uninviting the anonymous (pseudo-) user.

To enable the “public space” again, you can restore the “public spaces” by [undelete] the “public” folder from your trash or by creating a new folder with the name “public” with [File > New > Folder] in your “Communities” folder. Be sure to run from the command line

```
$ cd <bscw-runtime-path>
$ ./bin/bsadmin fix_anonymous
```

after you enabled the “public space” again.

7.6 WebDAV

WebDAV is an acronym for “Web-based Distributed Authoring and Versioning”. It is a set of extensions to the HTTP protocol (IETF RFC 2538) which allows users to collaboratively edit and manage files on remote Web servers, i.e., some of the BSCW features (e.g., document upload to a BSCW server or renaming of a document on a BSCW server) are also supported by the WebDAV protocol.

BSCW supports (a subset of) the WebDAV protocol. This means that some of the BSCW operations which are available via a Web browser and through the HTML interface of BSCW, are now also available via WebDAV clients (http://www.webdav.org/) for software supporting the WebDAV standard).

Note: There are many WebDAV clients available and we could only test a small subset of them with BSCW WebDAV support. From our tests we assume that not all WebDAV clients fully conform with the WebDAV specifications, i.e., you may have problems when using a particular WebDAV client with BSCW.

The BSCW WebDAV interface mandatory requires the Apache HTTP server version 2.2 or 2.4. Install Apache HTTP server. After the installation of the Apache HTTP server run bsadmin conf_apache (with BSCW user ID)

```
$ cd <bscw-runtime-path>
$ ./bin/bsadmin conf_apache
```

This creates new <bscw-runtime-path>/conf/apache{2,24}/ files.(cf. section 3.3.1 (Unix) / 4.4.2 (Windows)).

Microsoft Support for WebDAV

More recent Microsoft Windows and MS Office versions (Office 2003, 2007, 2010) provide WebDAV support. This allows the following applications:

1. Opening of BSCW workspaces within Windows Explorer. Proceed as follows:
   - Select “My Network Places” (either from your desktop or within Windows Explorer).
   - Select “Add Network Place”
   - Enter the URL of your BSCW home folder (or a sub-folder). This has the form
     https://bscw.domain.org/bscw/bscw.cgi/9620
     https://bscw.domain.org/bscw/bscw.cgi/home
     You will then be prompted for your BSCW user name and password.
   - Click “Finish”.

You may then browse with Windows Explorer through your BSCW workspaces in the same way in which you browse through the file system on your local computer. You may also "drag and drop" files from your local file system to BSCW folders or vice versa. When clicking the right mouse button, you may carry out several actions such as deleting or renaming an object.

2. Editing of MS Office files in BSCW workspaces. Proceed as follows:
   - From within Windows Explorer (see previous example) select, e.g., a MS Word document within a BSCW folder.
   - Open the document by double-clicking it.
   - Edit the Word document as usual.
   - After editing “Save” the document. The document will be saved back into the BSCW folder.

Known Problems

The Microsoft implementation of WebDAV is not fully compliant with the WebDAV RFC which may cause some problems. The following problems are already known:

- The RFC requests a special encoding of spaces and non-alphanumeric characters. MS Internet Explorer and Windows Explorer do not process such characters correctly. (Recommendation: Use only alphanumeric names (without spaces) for BSCW objects when MS Internet Explorer or Windows Explorer shall be used as WebDAV clients).
- Dialog boxes and error messages are sometimes misleading.
- Drag and Drop within the same directory results in a copy operation.
- Starting with Windows-Vista Microsoft requires a SSL encrypted connections via HTTPS (`https://...`) to allow WebDAV access.

See also the FAQ section 10.1.8 “How do I connect to BSCW using WebDAV?”. Please inform us if you observe additional problems.

7.7 Quota - Disk Usage Limitation

BSCW quota individually allows to restrict the amount of disk usage for users. In order to enable the BSCW quota system, the administrator has to define in a first step *limit classes*. Afterwards quota can be turned on for all or individual users by assigning a limit class to this users.

The BSCW disk space allocated to each user (quota) is computed as follows:

- When a user creates an object, the disk space used by the object is added to the quota of the owner of the folder wherein the object is created.
- In particular, when user A creates an object in a folder that is owned by user B, the quota of user B is affected, not the quota of user A.
- If the owner of a folder is removed from its members' list (either by others or by himself or herself), the ownership of the folder and of the objects therein is transferred to another person who still has access to this folder.
- This new procedure for computing the quota of users has the effect that users can always access all objects that contribute to their quota.

Note: By default quota is now enabled for the *anonymous* user to avoid the assignment of any resources to the *anonymous* user. To explicitly disable quota limitation for the *anonymous* user run the command `bsadmin quota off anonymous`. Alternatively you may assign a new limit class to the *anonymous* user with the command `bsadmin quota on -c <classname> anonymous`.

Quota is accessed by the BSCW administrator via the `bsadmin quota` command line interface. In general the `bsadmin quota` command supports the following four options

`bsadmin quota limit`	defines and lists all limit classes;
`bsadmin quota { on | off }`	enables/disables quotas for all or individual users;
`bsadmin quota { check |fix }`	checks or fixes disk and object usage for all users;
`bsadmin quota { report | class }`	report quota for users or limit classes

The `bsadmin quota` command executed without any argument displays the usage information:

```
$ ./bin/bsadmin quota
usage:
 bsadmin quota
 bsadmin quota { check | fix }            [<u1> ... <un>]
 bsadmin quota class                      [<c1> ... <cn>]
 bsadmin quota report [-b][-t][-L]  [-v]  [<u1> ... <un>]
 bsadmin quota on             [-v]        [<u1> ... <un>]
 bsadmin quota on      -c <c> [-v]        [<u1> ... <un>]
 bsadmin quota on      -R     [-v]        [<u1> ... <un>]
 bsadmin quota off            [-v]        [<u1> ... <un>]
 bsadmin quota limit  [-v]
 bsadmin quota limit      <c> { disk | objects } <soft> <hard> <time>
 bsadmin quota limit   -d <c> [ disk | objects ]

{ check | fix }   checks/fixes current disk and objects usage for users
class            report users for all specified classes
report           report quota for all or specified users
                 option -L: only quotas exceeding limits are shown
                 option -t: only quotas exceeding soft limits are shown
on               enable quota for users w/ default limit class
on    -c <c>     enable quota for users w/ limit class <c>
on    -R         reset quota timer for users
off              disable quota for users
limit            report quota limit classes
limit -d <c>     delete quota class <c>
limit -d <c> <l> delete quota limit <l> ::= {disk | objects} for class <c>
limit    <c> <l> add/replace quota limit class <c> for limit <l>
-v               verbose output
```

with the following option parameters:

```
<u1> ... <un>  string         list of registered BSCW user names
<c>             string        limit class name

<soft>          integer [char]limit value in bytes or in kilo (mega,
<hard>                        giga, tera) bytes with unit token 'K'
                              ('M', 'G', 'T').

<time>          integer [char]limit value in seconds or in minutes
                               (hours, days, weeks) with time token 'm'
                              ('h', 'd', 'w').
```

7.7.1 Limit Classes

A limit class specifies the amount of disk resources an user may use. Limit classes are manipulated with the `bsadmin quota limit` option, which allows the following parameters:

```
bsadmin quota limit [-v]
bsadmin quota limit     <c> { disk | objects } <soft> <hard> <time>
bsadmin quota limit  -d <c> [ disk | file ]
```

Listing of Limit Classes

The command `bsadmin quota limit` prints a list of all defined limit classes.

```
$ ./bin/bsadmin quota limit
                         Disk                      Objects
                  soft    hard    time      soft    hard    time
default              0       0      0s         0       0      0s
develop            40M     80M      2w       400     800      2w
```

The 'soft' value actually defines the amount of resource usage a user may allocate. The 'hard' value defines the maximum amount of resource usage at a time. The 'time' value defines the maximum time period a user may exceed the soft limit.

If an user exceeds her/his hard limit or does not reduce her/his resource usage below the soft limit after expiration of the 'time' limit, the user account gets **locked**. If an user account is locked only delete actions may be performed. The account automatically becomes unlocked if the user lowers her/his resource usage below the soft limit.

Definition of Limit Classes

A limit class is defined by the amount of disk space (disk limit) and the number of objects (object limit). In order to activate the BSCW quota system the administrator has to define at least one limit class and assign limits to this class.

- *Disk/Objects Limits*
  In the following example the '`develop`' class disk limit is set to 40 Mbyte soft and 80 Mbyte hard limitation with a time period of 2 weeks. In the second step the objects limit is set to a value of 400 objects soft and 800 objects hard limit and with a time period of 2 weeks:

  ```
  $ ./bin/bsadmin quota limit develop disk 40M 80M 2w
  $ ./bin/bsadmin quota limit develop objects 400 800 2w
  ```

- *Limit Class '`default`'*
  To enable quota immediately for new registered users the BSCW quota system supports a special limit class '`default`'. If a disk or a objects resource limit is defined for this class, quota is automatically enabled for all new users. In this case new users are assigned to this '`default`' limit class. Example:

  ```
  $ ./bin/bsadmin quota limit default disk 10M 15M 1w
  $ ./bin/bsadmin quota limit default objects 200 300 1w
  ```

  To disable this feature the '`default`' limit class must be removed with the command:

  ```
  $ ./bin/bsadmin quota limit -d default
  ```

7.7.2 Quota Activation

The administrator may enable (disable) quota for users with the `bsadmin quota on` (`bsadmin quota off`) command.

Enable Quota

If no limit class is specified with the `-c <c>` switch, the `bsadmin quota on` command enables quota for the specified user(s) and assigns them to the '`default`' limit class. Examples:

Enable quota for all users with assigned default limit class:

```
$ ./bin/bsadmin quota on
```

Enable quota for the individual users 'ruland' and 'koch' and assign them to the '`develop`' limit class:

```
$ ./bin/bsadmin quota on -c develop ruland koch
```

Change quota limit class for user 'paulsen' to class '`default`':

```
$ ./bin/bsadmin quota on -c default paulsen
```

Disable Quota

Quota may be disabled for all or individual user(s) with the `bsadmin quota off` command. Examples:

Disable quota for user 'hinrichs'

```
$ ./bin/bsadmin quota off hinrichs
```

Disable quota for all users:

```
$ ./bin/bsadmin quota off
```

Note: to disable automatic quota activation for new users the '`default`' limit class has to be removed (see above).

Note: To reset the quota limit timer for soft quotas use:

```
$ ./bin/bsadmin quota off -R hinrichs
```

7.7.3 Calculation of current disk usage

If quota is enabled for an existing user, the users' usage counters should be fixed to take the users current resource usage into account. For this purpose the BSCW quota system provides the commands

```
$ ./bin/bsadmin quota check
$ ./bin/bsadmin quota fix
```

The `check` command only proofs if the users' usage counters match the current (real) resource usage, while the `fix` command sets the users' usage counters to the current (real) resource usage.

Caution:

- To determine the current resource usage of an user, the `bsadmin quota fix` command has to examine all stored documents of the BSCW server. Depending on the number of stored documents this may take a *long* time.
- Do **never** run `bsadmin quota fix` while garbage collection is executed.

7.7.4 Report disk usage

The `bsadmin quota report` command prints a summary of the disk usage and quotas for all users:

```
$ ./bin/bsadmin quota report
                           Disk                            Objects
User            usage     soft     hard time     usage     soft     hard time
koch       --     12M      40M      80M             94      400      800
paulsen    +-   11.1M      10M      15M 3.3d       150      200      300
```

```
ruland      --   39.9M     40M     80M           345     400     800
```

For each user (with quota enabled) the current amount of disk space and number of objects is printed, along with any quotas of the users limit class.

If you additionally specify user names(s), a report is only generated for the given user(s):

```
$ ./bin/bsadmin quota report koch paulsen
                          Disk                          Objects
User            usage    soft    hard time     usage     soft     hard time
koch        --    12M     40M     80M             94      400      800
paulsen     +-  11.1M     10M     15M 3.3d       150      200      300
```

The additional switches `-t` or `-L` restrict the output of the quota report command to these users who are exceeding their soft limits (`-t`) or their limits (`-L`)

7.8 Definition of Roles

In the following section first a brief introduction in the BSCW role concept is presented. Then the definition of BSCW system defined roles is explained in detail and finally a simple mechanism to configure site-specific roles is given.

7.8.1 The BSCW role concept

In BSCW access rights are determined by the role or roles that a user holds. Roles are sets of actions that are allowed for the holder of a role. Users can be assigned one or more roles for an object at the same time. When a user holds a role, she may execute an action on the object if and only if the role includes that action. If a user holds multiple roles for an object, she is granted permission to the union of actions of all roles.

The scope of a role is the object for which a user holds that role and everything inside the object, unless and until the user is re-assigned another role. The role is thus valid for the object's scope: the object itself and its contents recursively. Roles are said to be *inherited* from a container object to its contents. Though this is also true for special containers like user's *Home* or *Clipboard*, the user's role in those special containers are not inherited to shared folders which are contained therein.

> Example:
>
> A user is by default the *Manager* of her *Home* space and of all objects and all sub-folders she perceives therein the default role *Manager* is inherited to the *Home* folder's scope.
>
> Assume that the user is now invited to a shared folder called *Project Documentation*, the inviting user assigns a role to her, say guest. The new member then holds the *guest* role for the entire *Project Documentation* and its contents. On the other hand, the shared folder *Project Documentation* appears top-level in the Home space of the new member. What roles will she play in the *Project Documentation* folder? If the role *Manager*, which she holds in her *Home* space, were inherited to *Project Documentation*, the user would hold *Manager* rights on the shared folder as well as *guest* rights which were assigned to her. To prevent this, special containers like *Home*, *Clipboard*, *Waste* do not inherited their roles to shared folders below. Instead, for shared folders inherit role assignments only from other shared folders.

In general roles in BSCW are either predefined by the system or defined by end-user (action "add role"). In the former case, the role can be applied to all BSCW objects. In the latter case, the role can only be assignment within the object's scope.

All roles (normal roles and special roles, see below) can be re-defined ("edit role") for any object, thereby changing the set of actions which are allowed for an object. In this case the changed role definition is valid for that object and its content recursively, but not for any other object. This means that there can be more than one role with the same name which have different scopes and different access rights definitions. There are different types of roles in BSCW:

Normal Roles

Normal roles in BSCW are roles which may be assigned to users without restrictions. Internally, these roles are prefixed by "`R2`" for predefined roles and by "`r`" for user-defined roles. End-users can only define ("Add role") normal roles.

Examples: `R2member`, `R2user`, `R2manager`, user-defined roles in workspaces like "Teacher" or "Student".

Special Roles

Special roles are roles which are restricted in the way in which they can be assigned to users or special in the way in which they are inherited. Their internal prefix is either "`R0`" or "`R1`". Only system administrators can define special roles; this is done in an extra “`local_roles`” package (see section 7.8.3 Site-specific Roles).

End-users cannot define (via “Add role”) special roles, but they may re-define (“Edit role”) `R0` or `R1` roles. As with normal roles, the changes which an end-user applies to a special role are limited to the object's scope.

System-defined roles: `R0` roles

System-defined roles are special roles which the system needs and which only the system can assign to users. In particular, users cannot be invited to workspaces in `R0` roles. By default, there are 2 system-defined `R0` roles in BSCW: `R0creator` and `R0owner`:

- `R0creator` is assigned to the creator of an object and is never re-assigned to another user.
- `R0owner` is by default assigned to the creator of an object, if created top-level (e.g., in the user's home or clipboard). Ownership is inherited to the object's scope: this means that the special role `R0owner` is assigned to all objects within the object's scope recursively.

Restricted roles: `R1` roles

Restricted `R1` roles behave differently from normal roles when the role holder is invited to a workspace. If a user holds a `R1` role and is invited to a workspace in another role, the invited role is simply ignored by the system. Instead, for that workspace the system assigns the special restricted role "`R1anonymous`" to the user.

The reason for this seemingly strange behavior lies in the past: recent BSCW systems allowed to invite the special user "anonymous" to workspaces, but restricted the anonymous user in its access rights. Younger BSCW systems must ensure the restricted access of anonymous users also for older BSCW databases. If, for instance, a group of users which contains the anonymous user is invited to a workspace holding the role `R2manager`, the anonymous user would automatically inherit the enhanced access rights of `R2manager`. This would be in contradiction to older BSCW systems and might grant the anonymous user access rights which were not intended in older BSCW databases.

Examples: `R1anonymous` (defined in all BSCW systems), `R1observer`

Assignment of roles

Normal roles and restricted roles are assigned in two ways:

- when inviting users to the members group of a workspace or other object
- explicitly for a user using the action "Assign role"

The former case allows to assign roles not only to users, but also to groups of users. This may lead to multiple roles a user holds: invite two groups of users which both contain a certain user.

The latter case is only possible for individual users, not for groups of users. It may be used to re-assign a role to a particular user who was invited as member of another group (the group being invited in another role).

When inviting users to a members group, any role which is defined globally or in that object's scope may be assigned to individual users or to groups of users. This includes restricted roles (`R1` roles), but not system-defined roles (`R0` roles).

Special roles can either not be assigned at all (`R0` roles) or they behave differently when being invited (`R1` roles). Cf. above for details.

What are user roles?

User roles are roles which are not assigned to a user in the scope of an object, but which are mapped to a user herself. User roles are valid for that user throughout the system and determine access rights to private data spaces of a user.

Only system administrators can assign a user role to a user (with “Assign Role” to an user object). The system administrator keeps a list of user roles available in `user_roles`. User roles can either be normal roles (`R2` roles) or restricted roles (`R1` roles).

The user role in which a user is registered or which a system administrator assigns to her determines the access rights to her private data spaces: her home space, clipboard, etc. By default, all private objects inside the private data spaces are subject to the user role which a user holds. Only when a user is invited to shared spaces, different roles are assigned to her and overrule her user role.

If a user is registered holding a restricted (`R1`) user role, she is restricted to the special "`R1anonymous`" role in all workspaces to which she is invited. This is regardless of the definition of her actual R1 user role. Therefore, user roles should in general be normal roles.

By default, BSCW user roles are set to “Manager” (`R2manager`, see `default_user_role` below). You may define your own role (e.g. `R2user`) and redefine the default role for registered users in your local `config_action.py`.

Extended access rights for the BSCW administrator

BSCW administrators may always execute the actions "Change role", "Assign role" and "Owner" on all folders, independent of their membership. Besides they may execute the action "More information" for all artefacts, and have the right to open all folders.

Because of the extensive rights that a BSCW administrator has (and must have), the administrator is not a role in the sense of the BSCW role concept for security reasons. It must be avoided under all circumstances that the property of being a BSCW administrator can be manipulated from the user interface.

7.8.2 Role definition and default roles

In general roles are defined as a union of action *views*. Action views are sets of actions specified for easier action handling. Action views are bit encoded, i.e. are defined as powers of 2. Currently there are twelve different views (‘`ext`’ stands for extended, all views have language dependent long names defined in `<bscw-pkg-dir>/bscw-5.0.9-3????-py26/bscw/msg/<lang>/lg_msgconfig.py`)

action view	*value*	*description*
`view('get')`	1	Actions that involve only 'read' access to an object, e.g. the `get` operation itself, `copy`, or `convert`
`view('get_ext')`	2	Actions that involve 'read' access to an object's meta data (description, info page), e.g. `info`.
`view('change')`	4	Actions that involve a 'write' access to an object, e.g. actions that create an object and actions that modify an object; this is the most common view.
`view('change_ext')`	8	Actions that move an object, i.e. change both the source and the target container; currently only the `cut` action
`view('owner')`	16	Actions that are exclusively for the owner of an object, i.e the `destroy` action.
`view('share')`	32	Actions that affect the access rights of an artefact (excluding role management), e.g. adding and removing a member.
`view('share_ext')`	64	Actions for role management, e.g. changing or assigning a role.
`view('edit')`	128	Actions for editing note board articles, attachments or appoint-ments.
`view('user')`	256	Actions that concern user information and communication, e.g. editing user details, sending e-mail or inviting other persons to become a BSCW user.
`view('waste')`	512	Actions that are possible in the waste, e.g. `destroy` and `undelete`.
`view('attend')`	1024	Actions allowed for attendees of a calendar appointment.
`view('creator')`	2048	Actions for the creator of an artefact, eg. edit and cut actions.
`view('responsible')`	4096	Actions for the responsible of a task.

A view comprises all actions that have this view assigned. The definition of a new view is done with the `view()` function.

Now we come to the definition of roles. The names of the predefined standard roles have the form `Ri<name>`, where `i` is a digit indicating the role type: `0`, `1` or `2` standing for 'system-defined', 'restricted' and 'normal', respectively. All standard roles are defined in the dictionary `default_roles` as follows

```
standard_views = (view_get | view_get_ext | view_change
                | view_change_ext | view_waste)
default_roles = {
  'R0creator': view_edit,
  'R0other': 0,
  'R0owner': view_owner,
  'R1anonymous': view_get,
  'R1restricted': view_get | view_get_ext,
  'R2associate': standard_views,
  'R2attendee': 0,
  'R2manager': (standard_views | view_user | view_share | view_edit |
      view_share_ext),
  'R2member': standard_views | view_user | view_share,
  'R2responsible': (standard_views | view_user | view_share | view_edit |
      view_share_ext | view_responsible),
}
```

The names of these predefined roles at the user interface are those that we introduced above. Internally, the standard predefined roles also have aliases that are used in BSCW kernel code.

```
other_role = 'R0other'              # special role "is a registered user"
owner_role = 'R0owner'              # special owner role
creator_role = 'R0creator'          # special creator role
anonymous_role = 'R1anonymous'      # default role for anonymous users
default_user_role = 'R2manager'     # default role for registered users
attendee_role = 'R2attendee'        # default role for attendees (appoint.)
default_role = 'R2member'           # default role for invitations
responsible_role = 'R2responsible' # responsibles for tasks etc.
```

BSCW defines the following roles:

1. "Restricted" Roles (BSCW core)
   - **Anonymous member**
     - "read-only" role for the pseudo member "anonymous" (to publish sub folders) without info-right
   - **Restricted member** (*)
     - "read-only" role for the usual members with info-right (descriptions are shown)
2. "Normal" Roles (BSCW core)
   - **Member** (*)
     - default role when inviting members with all read/write rights.
       Note: the default "Member" role allows members to invite/uninvite other member ("share view"), which is probably not desired.
   - **Associate member** (*)
     - default role for Communities (like Member role without "share view")
   - **Manager** (*)
     - workspace manager with the right to alter and assign roles (plus member rights)
     - default role for user top level objects: home, clipboard, waste.
   - **Participant**
     - special calendar role for the participants of an appointment
3. Additional "normal" roles (BSCW packages):
   - **Requestor**
     - role the requestor of a task holds (when started) (task package)
   - **Contractor**
     - role the contractor of a task holds (when started) (task package)
   - **Responsible**
     - role of the member who holds currently a flow folder (FlowFolder package)
4. System roles:
   - **Creator**
     - assigned by the system whenever a user creates a new object; normally allow cut/delete (If you withdraw the cut/delete right from the "Member" role, the "Creator" role will still allow creators of an object to delete it).
   - **Registered User**
     - this are all users of the system, like the Unix 'other' rights
   - **Owner**
     - The Owner role is assigned to each users top-level objects (home, clipboard, waste etc.) and inherited along the folder Hierarchy. It is used to assign resource usage for

the quota limitation.

(*) only these roles are shown in the "Invite Member" resp. "Assign Role" form.

All other roles are assigned automatically by the system and should therefore not assigned manually.

7.8.3 Site-specific Roles

In order to customize the default BSCW system roles settings it is possible to redefine roles using the BSCW instance action configuration files (`config_action.py`). By default the instance action configuration file is located in the main instance configuration directory `<bscw-runtime-path>/conf/` and allows to adopt BSCW kernel ("core") actions and roles:

```
<bscw-runtime-path>/conf/config_actions.py             # "core"
```

BSCW package actions and roles may be redefined in a package action configuration file `<bscw-runtime-path>/conf/<package-name>/config_actions.py` for each available package.

If you want to adopt the BSCW system roles setting use the

```
$ ./bin/bsadmin prtactions --print-conf
```

command line script to create a template action configuration file for each package. For convenience, each action configuration file template contains the action names for a package. Running `bsadmin prtactions --print-conf` will create/update the following files

```
<bscw-runtime-path>/conf/config_actions.py             # "core"

<bscw-runtime-path>/conf/blog/config_actions.py        # "blog"
<bscw-runtime-path>/conf/case/config_actions.py        # "case"
<bscw-runtime-path>/conf/factory/config_actions.py     # "factory"
<bscw-runtime-path>/conf/FlowFolder/config_actions.py  # "FlowFolder"
<bscw-runtime-path>/conf/poll/config_actions.py        # "poll"
<bscw-runtime-path>/conf/portal/config_actions.py      # "portal"
<bscw-runtime-path>/conf/presence/config_actions.py    # "presence"
<bscw-runtime-path>/conf/readers/config_actions.py     # "readers"
<bscw-runtime-path>/conf/rss/config_actions.py         # "rss"
<bscw-runtime-path>/conf/Secure/config_actions.py      # "Secure"
<bscw-runtime-path>/conf/SMS/config_actions.py         # "SMS"
<bscw-runtime-path>/conf/sync/config_actions.py        # "sync"
<bscw-runtime-path>/conf/Tasks/config_actions.py       # "Tasks"
<bscw-runtime-path>/conf/WebFolder/config_actions.py   # "WebFolder"
<bscw-runtime-path>/conf/wsmap/config_actions.py       # "wsmap"
```

In the following, we give an example for extending BSCW system defined roles (as described above) by adopting the action configuration file

```
<bscw-runtime-path>/conf/config_actions.py
```

We will define five new roles, "*Learner*", "*Author*", "*Domain manager*", "*Field manager*", and "*Educational advisor*".

To define language dependent translations for the roles name we create the following language dependent messages files for our BSCW instance (see also section "*5.21 msg/<lang>/lg_msgconfig.py*")

```
<bscw-runtime-path>/bsext/msg/de/lg_msgconfig.py
<bscw-runtime-path>/bsext/msg/en/lg_msgconfig.py
```

Here are the the file contents:

```
############################################################################
# File
#     <bscw-runtime-path>/conf/config_action.py
# The actions that are initially allowed for the new roles are
# given by 'or'-ing some bit_masks ("views")

# Note: The names of standard system defined roles must start with
# 'R2'. User friendly translations are defined in
# <bscw-runtime-path>/bsext/msg/*/lg_msgconfig.py
default_roles['R2learner'] = (
    view_get | view_get_ext | view_change | view_share)

# We might also use the default action set of other roles that are already
# defined (e.g. 'R2member'):

default_roles['R2author'] =  \
    default_roles['R2dommanager'] = \
    default_roles['R2fldmanager'] = \
    default_roles['R2eduadvisor'] = \
    default_roles['R2member']

# Note: Obviously it makes some sense to define different Roles that have
#       the same actions allowed *initially*

############################################################################
# File
#     <bscw-runtime-path>/bsext/msg/en/lg_msgconfig.py
# User friendly names for new roles defined in
#     <bscw-runtime-path>/conf/config_action.py
R2learner = 'Learner'
R2author = 'Author'
R2dommanager = 'Domain manager'
R2fldmanager = 'Field manager'
R2eduadvisor = 'Educational advisor'


############################################################################
# File
#     <bscw-runtime-path>/bsext/msg/de/lg_msgconfig.py
# User friendly names for new roles defined in
#     <bscw-runtime-path>/conf/config_action.py
R2learner = 'Lerner'
R2author = 'Autor'
R2dommanager = 'Domänenmanager'
R2fldmanager = 'Branchenmanager'
R2eduadvisor = 'Aus- und Weiterbildungsberater'
```

7.9 Site-specific banner

To customize the BSCW look you may specify a custom welcome message and insert a logo of your organisation into the BSCW index page. Additionally you can add a banner of your organisation at the top of each BSCW page. To insert a site-specific welcome message and logo into the index page or a banner at the top of each BSCW page follow these steps:

1. Create a BSCW instance specific resources directory `<bscw-runtime-path>/bsext/resources/icons` where to store your (customized) resource icons:

   ```
   $ cd <bscw-runtime-path>
   $ mkdir -p ./bsext/resources/icons
   $ chmod 755 ./bsext ./bsext/resources ./bsext/resources/icons
   ```

   Next link the resources directory from `./var/www/local` to `./bsext/resources`

   ```
   $ cd ./var/www
   $ ln -s ../../bsext/resources ./local
   ```

2. Copy a index logo or banner logo GIF/JPEG image of your organisation called `logo_index.gif` resp. `logo_banner.jpg` into the bsext resources icon directory (`<bscw-runtime-path>/bsext/resources/icons`):

   ```
   $ cd <bscw-runtime-path>
   $ cp logo_index.jpg logo_banner.jpg ./bsext/resources/icons
   $ chmod 644 ./bsext/resources/icons/logo_index.jpg
   $ chmod 644 ./bsext/resources/icons/logo_banner.jp
   ```

3. Run `bsadmin conf_apache` to make the local resources directory available to your Apache HTTP server configuration.
4. Add an entry `index_logo` resp. `server_logo` to the file `<bscw-runtime-path>/conf/config_icons.py`. Additionally you need to specify the image size `(width, height)`:

   ```
   index_logo = ('logo_index.jpg', (766, 132))
   server_logo = ('logo_banner.jpg', (220, 48))
   ```

   or

   ```
   index_logo = ('logo_index.gif', 766, 132))
   server_logo = ('logo_banner.gif', 220, 48))
   ```

   Note: The height of your banner image (`server_logo`) may not exceed `48` pixels.
5. To specify a custom welcome message for the BSCW index page define the `INDEX_MSG` configuration directive in the BSCW instance configuration file `<bscw-runtime-path>/conf/config.py` (see section 5.2.5 on page 70).
6. Run `bsadmin index_page` to generate new BSCW index pages

7.10 Server-wide template folders

By using the action `[File > New > from Template]` BSCW users may create new objects by copying it from an existing template. Any BSCW object can serve as a template, e.g. documents of any kind, polls, flow folders and tasks or even folders including their entire contents.

All templates to be copied either come from special template folders which are placed into the BSCW repository or from files which are placed in a special directory within `<bscw-runtime-path>` (see item 4 below). Template folders are special folders which you create by invoking `[File > New > Template Folder]` and which you then fill with template objects. They are indicated by a special "template folder" icon.

The templates that are being offered for copying in the Template selection list come from template folders in the following places:

1. The user's personal clipboard
2. Template folders in the current folder or template folders in folders on the current path
   - BSCW will also consider template folders which are contained in folders on the path upwards from the current folder to the top level folder.
   - Only template folders directly contained in folders on the current path are considered.
3. System-wide template folders/documents

   These are template folders which are accessible to all registered users. System-wide template folders are created and managed by the BSCW administrator only and are located within the anonymous user's clipboard.

   Additionally the BSCW administrator can provide document templates in a dedicated directory. The default document template directory is located at

   ```
   <bscw-runtime-path>/etc/doc_templates/
   ```

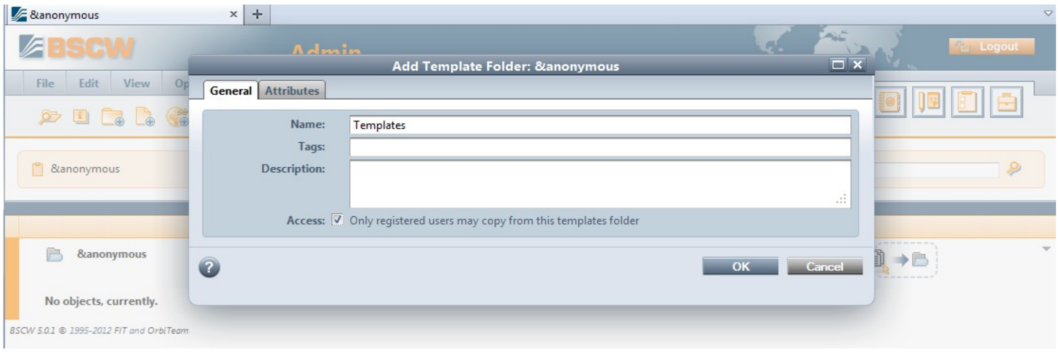
   The administrator might copy default documents into this directory. See also in the BSCW distribution directory `<bscw-path>/lib/bscw-5.0.9-3????-py26/etc/doc_templates/` for some basic template documents

To create a system-wide template folder:

1. Access the administrator's HTML interface, see section 7.1
2. Enter the URL

   ```
   http://bscw.domain.org/bscw/bscw.cgi/&anonymous?op=addtempls
   ```

   and fill out the following form:



Add Template Folder: &anonymous

General | Attributes

Name: Templates

Tags:

Description:

Access: ☑ Only registered users may copy from this templates folder

OK | Cancel

If the option „*Only registered users may copy from this template folder*" is activated, only registered users will be made available the template objects in their `[File > New > from Template]` form. Otherwise, even anonymous users can access the templates in this template folder.

Place whatever BSCW objects you want into the template folder to appear as templates for all (registered) users.

7.11 Web Service API

BSCW offers a range of services via different web service protocols: XML-RPC, JSON, SOAP

Basically most of the actions available on the user interface (like "add folder") are accessible via a web service API. Of course access to API is restricted via access control as in the regular user interface (i.e. authentication and BSCW internal roles and rights are respected).

In order to use the web service API it must be enabled by setting the `WEBSERVICE` variable to `1` (default) in the BSCW instance configuration file `<bscw-runtime-path>/conf/config.py`. If `WEBSERVICE` set to `0`, web service protocol requests will be rejected by BSCW with the HTTP error code `501: content_unsupported`.

A detailed description of the web service API can be found on our website at

```
http://www.bscw.de/english/documentation.html
```

or in the BSCW distribution at

```
<bscw-path>/lib/bscw-5.0.9-3????-py26/doc/devel/BSCW50-API-doc.zip
```

7.12 Some useful hints

There exist a number of naming conventions for user objects which may be useful to know for system administrators. These conventions can be used to address the respective objects directly by entering a corresponding URL into the address field of the browser. The URL patterns for these URLs are

`https://bscw.domain.org/bscw/bscw.cgi/<shortname><username>` or
`https://bscw.domain.org/bscw/bscw.cgi/<shortname><emailaddress>`

where `<shortname>` is a single character of the following list

```
<shortname> ::= {
     @ #addrBook |
     _ #waste |
     ` #case |
     $ #lockbag |
     & #bag |
     + #calendar |
     * #bookmarks |
     : #home |
     = #portal |
     ~ #tasklist |
     u #user |
     m #email address
}
```

and `<username>` is the name of a registered user and `<emailaddress>` is an email address for which a registration process has been initiated. For example, for a user with BSCW user name "`Smith`" and the email address "`smith@company.com`" the URL

- `https://bscw.domain.org/bscw/bscw.cgi/mSmith@company.com`
  will return the info page of the email address, in particular status information about the email address (pending, allocated, bounced) and a link to the BSCW user if allocated;

- `https://bscw.domain.org/bscw/bscw.cgi/uSmith`
  returns the info page for user Smith with additional information available only to system administrators such as icons leading to the user's home page, bag, waste basket and the list of locks that the user has currently set on documents;

The following short names may be used to immediately access the users' personal objects

- `https://bscw.domain.org/bscw/bscw.cgi/@Smith`
  shows the users' address book;

- `https://bscw.domain.org/bscw/bscw.cgi/_Smith`
  shows the users' waste basket;
- `https://bscw.domain.org/bscw/bscw.cgi/`Smith`
  shows the users' briefcase;
- `https://bscw.domain.org/bscw/bscw.cgi/$Smith`
  shows the locks that the user currently holds on documents;
- `https://bscw.domain.org/bscw/bscw.cgi/&Smith`
  shows the users' clipboard;
- `https://bscw.domain.org/bscw/bscw.cgi/+Smith`
  shows the users' calendar;
- `https://bscw.domain.org/bscw/bscw.cgi/*Smith`
  shows the users' bookmarks;
- `https://bscw.domain.org/bscw/bscw.cgi/:Smith`
  shows the users' home page;
- `https://bscw.domain.org/bscw/bscw.cgi/=Smith`
  shows the users' portal;
- `https://bscw.domain.org/bscw/bscw.cgi/~Smith`
  shows the users' task list.

8 BSCW Help

The BSCW help is available for from your BSCW server instance:

- http://<server>/pub/static/help/english/
- http://<server>/pub/static/help/german/

Alternatively you may access the help on our web page:

- http://www.bscw.de/help/ (English version)
- http://www.bscw.de/hilfe/ (German version)

The BSCW help files are provided as HTML pages for on-line browsing as well as PDF files for printing.

Note: to view PDF files you need the Acrobat Reader. You can download the Acrobat Reader for different platforms directly from the Adobe Web site at http://www.adobe.com/ free of charge.

8.1 Languages

8.1.1 Existing translations

English, French, German and Spanish interface languages are included in the standard distribution of BSCW. A number of people have already prepared translations into additional languages and made them publicly available. Please check the BSCW homepage at http://www.bscw.de/english/extensions.html for available languages.

Note: to select a specific language version you've got to instruct your browser to set your default language to the respective language. Alternatively you may define your language in your BSCW personal preferences settings [Options > Preferences][General][Basic Preferences]: select your language and disable the "Use automatic language detection from browser, if available" check box.

8.1.2 Translation instructions

You can add support for new languages by creating a sub directory in your BSCW instance <bscw-runtime-path>/bsext/msg folder with the ISO language code of the language, these codes are the lower-case two-letter codes as defined by ISO-639 (you can find a full list of these codes at a number of sites, such as: http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt).

Beside your instance specific modifications in <bscw-runtime-path>/bsext/msg/* the distributed translations are located in the <bscw-path>/lib/bscw-5.0.9-3????-py26/bscw/msg/* directories. The distributed directories bscw/msg/en/* and bscw/msg/de/* contain all relevant language dependent strings and more files for the English version (default). Relevant for translation are *.py, *_help.html, *.txt, *.mail, *.mail.txt and *.mail.html files:

- *.py: Python source code, containing variables which in turn contain natural language strings. Each *.py, except lg_msgconfig.py, corresponds to a *.xhtml file stored in bscw/templatea which contains content and layout information, but is language independent. At runtime both files are merged to produce a language dependent HTML output file.
- *_help.html: Help files for context sensitive help
- *.txt: Text templates, usually containing system messages
- *.mail: Mail templates

- `*.mail.txt`: Mail templates, containing mail messages, text only
- `*.mail.html`: Mail templates, containing mail messages, HTML formatted

Other files *need not*, *cannot* and *must not* be translated!

Please contact OrbiTeam <support@orbiteam.de> if you want to translate BSCW to a certain language or if you update an existing language. We can provide you with an Excel data sheet where you can enter your translations.

Alternatively you can provide a new translation BSCW as follows:

First create a new directory `<bscw-runtime-path>/bsext/msg/<your-language-two-letter-code>` and copy each relevant file to `<bscw-runtime-path>/msg/<your-language-two-letter-code>` translate the English strings, but make sure to leave HTML/Python syntax intact. Files which do not contain language dependent strings must not be copied.

Special attention should be paid to the central language dependent file "`lg_msgconfig.py`". Please read the instructions in the file; it contains a large set of Python variables used all over the code. Make sure to leave the Python syntax structure intact.

It makes upgrading to later versions a lot easier, if for each line in `msg/en/lg_msgconfig.py` there is a corresponding line in `msg/<your-language>/lg_msgconfig.py`, even if it is commented out. Also, the variables should appear in *exactly the same order* in all languages. It is recommended that you start your translation with `lg_msgconfig.py`.

Next translate the additional BSCW packages are stored under `msg/<package-name>`. Follow the translation procedure outlined above

Please send us an email (support@orbiteam.de) and include either the translation or a link to it. Also, please send us the names and institutions of the people who should be credited with the translation. We would like to include them in our hall of translators. Thank you very much for your work!

Notes:

- Some strings should not be translated at all, e.g., server error messages determined for system administrators - this is up to your discretion. A variable in `lg_msgconfig.py` that is not translated into `<your-message>` should be commented out, but left in that file to preserve the order of variables.
- Make sure that you do not add white space to HTML templates – just replace the English strings. Also make sure that you do not remove quotes from Python variables. This will result in syntactically incorrect Python code. Use simple quotes (`'` or `"`) for single-line strings, and triple quotes (`"""`) for multiple-line strings.
- Please do not translate the mail headers (`To:`, `From:`, `Subject:`, etc.) in `*.mail*` template files.
- Certain resources (AIR Widgets, Java Applets) are not included in the above mentioned files – contact OrbiTeam <support@orbiteam.de> to translate this resources.

8.2 BSCW Updates

New BSCW versions will be announced on the BSCW mailing lists. The versions can be found on the download page of http://www.bscw.de/english/download.html. Before upgrading to a new version please see section 2.3.

9 BSCW license

Initially the server software is equipped with a test license, which allows usage of the server for a period of 90 days. The maximum number of users who may register with the server is limited to 200 (see also file `BSCW_COPYRIGHT`).

> **Note:**
> Since parts of your BSCW server URL (scheme, server name and partial path) are included in the license code it is **not** possible to change the BSCW server URL (as specified in the `SERVER_ROOT` variable setting in the BSCW instance configuration file `<bscw-runtime-path>/conf/config.py`) **without** changing the license via the license upgrade process or reinstalling the test license.

A BSCW administrator may commence a license upgrade process by clicking the “Upgrade license” link, which is provided in the administrator interface of the BSCW server. The “new license” option in the “Upgrade license” action will connect to the license server configured in the variable `BSCW_LICENSE` (see `<bscw-runtime-path>/conf/config.py`). Here one of the following alternatives applies:

Application for a royalty free license:

After the request for a royalty free license, a license agreement is displayed. The licensee has to print, sign, and send this license agreement to licenser. After reception of the signed license agreement, licenser will decide if licensee qualifies for a royalty free license. As a rule, licenser will grant such a license to schools and universities for educational purposes but reserves the right to deny such a license without further notice.

Application for a commercial license:

After the request for a commercial license, licensee will receive (by fax, if licensee has provided a fax number or otherwise by postal mail, normally within less than three days) a license agreement and an invoice for the requested license. After payment the license is granted; payment implies acceptance of the license agreement.

When the license is granted, licensee is notified by email. A BSCW administrator is now able to upload the license to his server by means of the “get license” option in the “Upgrade License” action. Running the garbage collector and a BSCW server restart installs a new BSCW license.

Generally a license (as shown in the “Upgrade License” action) has the following format

```
<reversed hostname>:<port><scheme>.<path>
```

`<reversed hostname>`	reversed FQDN components of the hostname
`<port>`	port of the HTTP server
`<scheme>`	`H` for HTTP or `S` for HTTPS
`<path>`	local path to the bscw.cgi script

For example a license for a BSCW server on host `bscw.domain.org` with the script path `/bscw/bscw.cgi` using `HTTPS` looks like:

```
org.domain.bscw:443S.bscw
```

10 Frequently Asked Questions (FAQ)

10.1 BSCW Server Usage

10.1.1 What do I need to use BSCW?

- You need access to the Internet.
- You need a personal email address to register.
- You need need a Web browser to access shared workspaces and to download documents to your local computer. Most Web browsers (e.g. Firefox, Chrome, Safari and Microsoft Internet Explorer) are compatible. We recommend using latest versions of Firefox.

On most computers everything is already available. You need no special software installation to start with BSCW.

Keywords: Prerequisites, usage of BSCW

10.1.2 Do I need a special application for uploading documents?

- No, you only need your browser application. Today's browsers include support for uploading based on a standard protocol and BSCW users may upload documents using these browsers with no problems. If your browser doesn't support file upload you should update the browser version because with old browsers you are not able to use all the features of the Internet.
- In any case, there is a special helper application for file upload available which are called BSCW Desktop. These programs offers special features for uploading complete directories or packing files while you upload.

Keywords: helper program, additional programs, upload documents

10.1.3 How is BSCW intended to be used

BSCW is a groupware application. Users share workspace folders which contain objects. These objects can be files, discussions, notes, calenders etc.

- To start with BSCW a user has to **create a workspace**. Then he invites users to this workspace. These users can be already registered or unregistered. Invitation is possible with the users user name if he already has one or with the new users email address.
- After invitation the shared workspace folder appears in the home folder of the invited user. The invited user may now access the shared workspace.
- If a user leaves a group he may simply be removed from the list of workspace members.

It's also possible to define special access rights for invited members by using the built-in role based access control system. Please read the documentation for more details.

Keywords: Usage of BSCW

10.1.4 Why do I get a save dialog box when I select add document?

You have got activated the *BSCW Desktop Uploader* and did not install *BSCW Desktop* on your local system. Deactivate the *BSCW Desktop* with

```
[Options > Preferences]
[General]
[File Handling]
```

```
        [ ] Use BSCW Desktop client for file upload
        [OK]
```

Alternatively you may install and configure the *BSCW Desktop Uploader* (see online help).

Keywords: BSCW desktop, upload, preferences

10.1.5 I cannot log in. The server rejects me - what shall I do?

Please mind that the BSCW server distinguishes between uppercase and lowercase characters in username and password.

If you forget your password, you can't change your password in the normal way. For this emergency case, BSCW provides a specific procedure to assign a new password without having to provide the old one:

- Open the URL
  `https://<your-server>/pub/bscw.cgi?op=chpwd`
  (e.g. `https://public.bscw.de/pub/bscw.cgi?op=chpwd` on the public BSCW server)
- Fill in the form with your primary email address
- An email with further instructions to reset your password will be sent to you - follow instructions in the email.

Keywords: login failed, forgotten password, forgotten user name

10.1.6 How do I change my password?

You may change your password using the menu item:

```
        [Options] [Change Password]
```

Keywords: password, change password

10.1.7 How do I configure my web browser?

In general it should not be required to perform special configuration in your web browser when using BSCW with an up to date browser. However if you encounter problems while working with BSCW you should check the configuration of your web browser.

Most problems are related to caching. The web browser should always contact the BSCW server before using pages from the cache. Please ensure the following settings:

Firefox: View all configuration settings by entering '`about:config`' in the location field. Set the value ("double click") of `browser.cache.check_for_frequency` to `1`.

Internet Explorer: choose in

"Tools" > "Internet Options" > "General" >
"Temporary Internet files" > "Settings" >
[x] "Every visit to the page"

You should disable the feature "Tools" > "Internet Options" > "Advanced" > "Browsing" > "Show friendly HTTP error messages" (otherwise IE displays meaningless error messages).

Mozilla: choose in preferences

"Edit" > "Preferences" > "Advanced" > "Cache"   [x] Every time I view the page

Opera: choose in preferences

"Tools" > "Preferences"
[Advanced]
[History]
Check documents: [Always]

Please mind, depending on your web browsers version the above mentioned configuration procedures may differ!

Hint: Make sure that the clock on your computer is set correctly. Otherwise synchronisation between the BSCW server and pages in your local cache may not work correctly.

Keywords: browser, cache

10.1.8 How do I connect to BSCW using WebDAV?

WebDAV (*Web-based Distributed Authoring and Versioning*, see www.webdav.org) is a standard protocol which allows users to access files on remote web servers. BSCW implements WebDAV so that it is possible to browse, upload and download files on a BSCW server using a WebDAV compliant client tool.

WebDAV protocol support is integrated in most operating systems such as MacOS, Windows and Linux (using KDE, for example). Alternatively, special WebDAV client applications may be used which are available for different platforms and at different licensing models (for example: cadaver, DAV Explorer or Novell Netdrive Client).

Connecting to a WebDAV enabled server typically only requires provision of the network URL and user credentials (user name and password), however, the process may vary depending on the WebDAV application used. In the following we shortly describe how to connect to a WebDAV enabled BSCW server using Windows (Windows XP with SP3, Vista with SP2 or Windows 7 is assumed).

Note: Not every BSCW server is WebDAV enabled. This depends on the BSCW version and the server configuration. If in doubt ask your BSCW administrator for help.

How to connect to BSCW using WebDAV on Windows XP

In order to connect to the BSCW server using WebDAV, it is recommended to use "Network Places" in Windows and then to choose "add network place", where you have to enter the URL of your BSCW server and confirm with [Next]. You will then be prompted for your BSCW user name and password. If everything works fine you will finally see a new item in your "Network Places" in Windows where you get access to the BSCW server. Your home folder (':username') is accessible via the alias folder 'home'. You may now browse your workspaces using the Windows File Explorer, and upload or download files (using copy & paste or drag & drop).

How to connect to BSCW using WebDAV on Windows Vista/7

In order to connect to the BSCW server using WebDAV, it is recommended to open the "Computer" (icon on your desktop) then right-click (in an open space of the Window) and to select "Add a Network Location" from the context menu. In the "Add Network Location Wizard" click [Next]., choose a "custom network location" and enter the URL of the BSCW server when prompted for the "location of the website". Enter the full BSCW server URL (including `/bscw/bscw.cgi`) in the "Internet or network address field".

For example, enter "`http://mybscw.server.com/bscw/bscw.cgi`" and click [Next]. You will then be prompted for your BSCW user name and password. If everything works fine you will finally be prompted for a name for this location - enter a label of your choosing (e.g. "My BSCW Server"). A new item in your "Computer" with that name should then appear " (note that this operation may take some time for the first time). The item provides access to the BSCW server: your home folder (':username') is accessible via the alias folder 'home'. You may now browse your workspaces using the Windows File Explorer, and upload or download files (using copy & paste or drag & drop).

Note: In order to reuse WebDAV resources stored on Windows 7 you have to enable the "WebClient service" by setting the service Startup type to "Automatic", see

- English: `http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/f89838b2-e8ef-41ea-99b6-6c829ffbcacd.mspx`
- German: `http://technet.microsoft.com/de-de/library/cc781730%28WS.10%29.aspx`

Troubleshooting

In case the above described method does not work (e.g. password dialog keeps popping up) the following tips have proven to help in most cases on Windows:

1. Try to connect using **`https`** i.e. enter the full URL
   `https://bscw.server.de/bscw/bscw.cgi/`
2. If your server doesn't support HTTPS, ask your BSCW administrator to enable HTTPS on the server. If that is not possible follow the hints given by Microsoft on how to enable basic authentication for WebDAV on the client computer (see below).
3. Try to add the number sign # to the HTTP address when you enter the URL into the field of the "My Network Places" assistant, e.g. Enter
   `http://bscw.server.de/bscw/bscw.cgi/#`

   (This will force **Windows XP** to use the "Microsoft Data Access Internet Publishing Provider DAV 1.1" mechanism instead of "Microsoft-WebDAV-MiniRedir/5.1.2600")
4. If the password dialog pops up again and contains a hostname in front of your username (e.g. "server\smith"), correct the username (i.e. remove "server\"), enter your password and click [OK] (this step may need to be performed several times when connecting for the first time).
5. If establishing a network connection to your BSCW server is still not possible, try to add the network location and enter the 'share' URL: `\\bscw.server.de\bscw\bscw.cgi`
6. Make sure you've installed all recent updates and service packs.
7. Make sure you're BSCW server is running the most recent version of the BSCW software. If in doubt ask your BSCW administrator for help

Hints for Windows Vista

On Windows Vista you may not connect to your BSCW server as a network drive using WebDAV if the server does not support SSL. You may want to ask your BSCW administrator to enable HTTPS on the server. If that is not possible, you may want to follow the hints given by Microsoft on how to enable basic authentication for WebDAV on the client computer:

1. Click Start , type **regedit** in the Start Search box, and then click `regedit.exe` in the Programs list.
2. Locate and then click the following registry key:
   `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WebClient\Parameters`
3. On the Edit menu, point to New, and then click DWORD Value.
4. Type `BasicAuthLevel`, and then press Enter.
5. On the Edit menu, click Modify. In the Value data box, type 2, and then click OK.
6. Close the registry editor. Finally, you should restart you computer.

Note: For security purposes, Windows Vista and Windows XP SP2 disable basic authentication in the Web Distributed Authoring and Versioning (WebDAV) Redirector. Therefore either use of HTTPS (SSL connection) is required or a special configuration (on the client) has to be done as described in the MS Knowledgebase: `http://support.microsoft.com/kb/841215/en-us`

Hints for BSCW administrators

In case your users encounter problems with WebDAV connections, the following tips have proven to help in most cases:

1. Make sure you're BSCW server is running the most recent version of the BSCW software. (Check the website www.bscw.de for updates.)
2. You may change the authentication method BSCW uses when user credentials are passed to BSCW. The configuration variable `AUTH_MODE` may be set to `'Basic'` (for basic access authentication) or `'Digest'` (for digest access authentication). Basic authentication may limit WebDAV access if SSL is not enabled (see above). **Attention**: digest authentication is not possible in combination with LDAP or with email address login.
3. If you operate more than one BSCW-Server instance on one host, you should make sure that all BSCW-Server instances are running the same (most recent) version of the BSCW software. You must then select one of the servers in order to handle WebDAV `PROPFIND`-Requests for the root resource (`'/'`). This is achieved by setting the `SCRIPTS_OTHER_ROOTS` variable in the server configuration file `<bscw-runtime-path>/conf/config.py` (see comments there).
4. When changing your server configuration (i.e. `<bscw-runtime-path>/conf/config.py`) regarding WebDAV, make sure you update the HTTP server configuration via "`bsadmin conf_apache -n`" and do not forget to restart your Apache Web Server.
5. Note that BSCW on Windows IIS does not support WebDAV!

Please refer to the following table to see if the WebDAV edit feature works:

Windows	IE	Office	Protocol	BSCW	WebDAV edit
XP 32bit	IE 8	Office 2003	HTTP	4.5.9	yes
XP 32bit	IE 8	Office 2003	HTTPS	4.5.9	yes
XP 32bit	IE 8	Office 2007	HTTP	4.5.9	no
XP 32bit	IE 8	Office 2007	HTTPS	4.5.9	yes
XP 32bit	IE 8	Office 2010	HTTP	4.5.9	no
XP 32bit	IE 8	Office 2010	HTTPS	4.5.9	no
Vista 32bit	IE 9	Office 2007	HTTP	4.5.9	no
Vista 32bit	IE 9	Office 2007	HTTPS	4.5.9	yes
7 64bit	IE 8	Office 2003	HTTP	4.5.9	no
7 64bit	IE 8	Office 2003	HTTPS	4.5.9	yes, *)

7 64bit	IE 9	Office 2007	HTTP	4.5.9	no
7 64bit	IE 9	Office 2007	HTTPS	4.5.9	yes
7 64bit	IE 9	Office 2010	HTTP	4.5.9	no
7 64bit	IE 9	Office 2010	HTTPS	4.5.9	yes
7 64bit	IE 9	Office 2013	HTTP	5.0.7	no
7 64bit	IE 9	Office 2013	HTTPS	5.0.7	yes
7 64bit	IE 10/11	Office 2010	HTTP	5.0.7	no
7 64bit	IE 10/11	Office 2010	HTTPS	5.0.7	yes
7 64bit	IE 10/11	Office 2013	HTTP	5.0.7	no
7 64bit	IE 10/11	Office 2013	HTTPS	5.0.7	yes
8 64bit	IE 11	Office 2013	HTTP	5.0.7	no
9 64bit	IE 11	Office 2013	HTTPS	5.0.7	yes

*) Please install the following patches:

1. Win Webfldrs-KB907306-DEU/ENU:
   - Description:
     Erroneous handling of 8bit characters in path names
     http://support.microsoft.com/default.aspx?scid=kb;en-us;892211
   - System-Requirements:
     Windows Server 2003, Windows Vista, Windows XP
     Download:
     http://www.microsoft.com/downloads/de-de/details.aspx?FamilyID=17c36612-632e-4c04-9382-987622ed1d64
2. enable WebClient-Service (set to "automatic") (Win 7)
3. deactivate proxy timeout for WebDAV requests
   Description:
   http://support.microsoft.com/kb/2445570

10.1.9 Hints for users of Windows Vista

In general, users of Windows Vista should have no problem in using the BSCW service via the web browser. However in case of certain components of the BSCW software that access services of the operating system special hints should be observed.

Hints regarding usage of the Briefcase feature in BSCW

When using the BSCW Briefcase feature (offline synchronisation of files between BSCW server and local PC) on Windows Vista and Internet Explorer (IE7) certain restrictions have to be regarded. A solution within BSCW cannot be achieved as these are general restrictions with Windows-Vista/IE7/Java. However a number of **alternative** workarounds is shown below:

1. Use the Firefox Browser on Vista. The BSCW Briefcase Applet works fine with the Firefox-Browser (Firefox still provides full access to system resources to the signed Java Applet). **OR**
2. Select a local directory for the BSCW Briefcase folder with a lower security level ("low level integrity") when using IE7/Vista. Choose an according directory within BSCW using "Options/Synchronisation", e.g.

   ```
   C:\Users\Name\AppData\LocalLow\BSCW
   ```

   OR
3. De-activate the "Protected Mode" in IE7 by declaring the URL of the BSCW server as a secure site. Add the URL of the BSCW server to the list of trusted sites (Extras / Internet Options / Security / Sites). The level "trusted sites" should have the option "enable protected mode" de-activated.
   OR
4. Change the security level of the local Briefcase folder. Open the windows console as administrator and execute the following command:

   ```
   icacls  /setintegritylevel (OI)(CI)low
   ```

Hint: We suggest to de-activate the "Protected Mode" in IE7 by declaring the URL of the BSCW server as a secure site.

Background information

The BSCW Briefcase Applet is a signed Java Applet and requires full access to the local file system in order to read and write files during synchronisation. The Java security concept ("sandbox model") does allow this, however recent changes in Vista/IE7 prevent Java Applets from accessing system resources. The reason is in the new security model introduced with MS Vista (Mandatory Integrity Control - MIC) and IE7 (IE7 Protected Mode). IE7 is running in a low security level and does only have write access to directories of the same security level. The same restrictions apply to components running within IE7 (such as Java Applets).

Oracle (developer of the Java technology) is informed about the problem and a general solution within the next version of the Java-Software (especially the Java-Plugin) is expected.

10.1.10 How do I destroy a workspace?

You may destroy a workspace by first removing all members of this workspace except yourself. Then the workspace can be deleted and moved into your trash. Afterward you can remove the workspace from the trash.

If you are not owner of the workspace and you remove the workspace without removing all members first, the workspace is only removed from your home folder. Other workspace members still have access to it.

If you are owner of the workspace and you remove the workspace from the trash, the system will automatically remove *all members of this workspace* so *no one* may further access it. The system will provide a warning message in this case.

Keywords: destroy workspace, remove members

10.1.11 How do I delete my account?

Only if enabled, it is possible to delete your own user account with `[Options > Destroy Account]`. If this menu entry is missing you cannot delete your account. If you do not want to receive daily email reports any more you may simply disable this in `[Options > Preferences] [Event Options]`. If you really want to be deleted please contact your local BSCW administrator.

Contact our Support team only if you are using a BSCW server that is operated by OrbiTeam. Don't forget to provide your username and server address (URL).

Keywords: delete account

10.1.12 How do I handle a JavaScript error?

Web browsers differ slightly in their JavaScript implementations which may cause problems when using BSCW. In some browsers like Internet Explorer you may disable this error messages. Please try to deactivate the error messages and test if BSCW still works with your browser. Alternatively you may upgrade to a new browser version to solve this problem.

In case your browser does not support JavaScript or if you have disabled JavaScript in your browser, BSCW will automatically choose a JavaScript-free representation of the user interface.

Keywords: JavaScript, JavaScript errors

10.1.13 How do I handle a BSCW error?

If you encounter a BSCW error message where the system message starts with `traceback` you may have found a bug in the software or a problem in the system configuration. Please first contact your local BSCW server administrator.

If the problem may not be solved, please contact the Support team at support@orbiteam.de and include the following information in your report:

- the complete error message (the `traceback`) you get,
- the URL of the BSCW server you use,
- describe exactly what you did before you got this error message,
- if you think more information about the computer system you are using is needed, include it.

Thank you very much for help.

Keywords: traceback, BSCW error, bug report

10.1.14 I reached the limit of my disk space - what shall I do?

The disk space an object occupies is subtracted from the quota of the owner of the workspace the object is in(and not to the quota of the creator of the object!). This is the reason why

- your quota does not decrease when you upload files to a shared workspace you are not the owner of,
- you are sometimes asked that another user has to delete files. He is the owner of the workspace you want to upload files to and his quota is exhausted.

There is a soft and a hard quota. You can exceed the soft quota temporarily for some days. After that it is not possible to upload files in the workspaces any more. You can never exceed your hard quota.

To upload files the owner of the workspace must have enough free space. Ask the owner of the workspace for help. He should delete some files. Normally there are files in the waste. Sometimes there are forgotten files in the clipboard. To empty the garbage

- please go to [Waste]
- select all files
- press [destroy]

If this doesn't resolve your disk space problem you may ask your system administrator to provide you with more disk space. This is not possible on the public BSCW server at http://public.bscw.de/.

Note:

- Disk space limiting is set per user.
- Disk space accounting concerns all workspaces you are the owner of.
- Disk space accounting is updated, whenever some user creates or deletes some document in your workspace.
- You can control your quota the following way:
  - Choose `[Options > Profile > View]`
  - On the info-page you'll find all necessary information about used disk space and quota.
  - You may check your quota limit of in MB (Megabyte) and the amount of currently used space.

Keywords: quota, disk space limit

10.1.15 Why does MS-Word mark a document as read-only?

This depends on the used version of MS-Office as well as on the configuration of your BSCW server. Recent versions of BSCW allow direct editing of documents using MS-Office. Please contact your local BSCW administrator if this feature is available on your BSCW server.

In case the direct editing of documents using MS-Office is not available on your BSCW server, then a Word document that is downloaded from a BSCW server and opened with the MS-Word application may be marked as Read-Only (because Word realized that this document came from a web resource and Word can not save it back to this web resource). If you want to edit the document, you have to save it locally on your PC ('Save as') and replace or revise the corresponding BSCW document on the BSCW server when you have finished editing.

Keywords: MS-Office, editing documents, Word-documents, Read-Only

10.1.16 Is there a restriction for the size of documents I upload?

No, there's no general restriction. If you run into problems while trying to upload large documents

- Please check your local network configuration (firewall, proxy etc.) Some networks restrict the size of files that may be uploaded through the network to a remote server. Contact your local system administrator for details.
- Please check your browser. Some browsers have problems with uploading large files.
- You may switch to a WebDAV client or use the BSCW Desktop to upload documents to BSCW.
- When using the BSCW Java drag & drop uploader applet the maximum upload size is restricted by the size of the Java virual machine memory size. In a default configuration you can upload files up to 60MB.

Keywords: restriction, size, upload

10.2 BSCW Server Software

10.2.1 How do I get the BSCW software?

The latest version of the software is always available for download from our download pages at http://www.bscw.de/english/download.html. Alternatively you may also order a free demo CD which contains the software at http://www.bscw.de/english/cdorder.html.

Usage of the BSCW server software is limited to a testing and evaluation period of 90 days and restricted to 200 users. After that period you have to acquire a license to continue usage. The distribution of BSCW licenses is handled by OrbiTeam Software GmbH & Co. KG, a spin-off company of FIT Fraunhofer Institute.

Schools and universities may apply for royalty free licenses for educational use only. In this case, BSCW must not be used commercially or in the context of funded projects. Any other use of the software requires the payment of a license fee.

For more information on licensing conditions and license fees, please contact our sales department at license@orbiteam.de.

Keywords: BSCW software, download, licensing

10.2.2 Can I try the BSCW software?

You may evaluate the BSCW software for 90 days free of charge. For this purpose you may either use a demo server provided by OrbiTeam and test the software on-line - or download the software and test it on your own server.

Please read the information on the evaluation process for further details at http://www.bscw.de/english/free_evaluation.html.

Please note that for the on-line trial all data you upload will be deleted after 90 days. Your BSCW structure cannot be copied to another BSCW server if you decide to rent a workspace after the trial ends.

Keywords: BSCW software, online trial, evaluation

10.2.3 How do I keep up to date with BSCW developments and new releases?

The best way to keep up to date is to subscribe to our announcement mailing list. You may **subscribe** to this mailing list on the website - and of course **unsubscribe** at any time.

You will find the archive of this list at http://lists.orbiteam.com/mail.cgi/archive/announceen

Customers will also be notified about new releases automatically (i.e. they are automatically subscribed to this list).

You may also want to follow us on Twitter (http://twitter.com/orbiteam) for more instant updates.

Finally you may want to check our website frequently to check for news and updates.

Keywords: new releases, update, developments, announcement mailing list

10.2.4 Why do I get an “Internal Server Error” on RHEL or derivatives?

We experienced crashes of the Python 2.6 interpreter in the Red Hat Enterprise Linux (RHEL) 6 or derivatives (CentOS/Scientific Linux) distributions, when opening a BSCW 5.0 dialog (e.g. when creating a new folder). In this case on the web interface an "Internal Server Error" is shown and the error logfile of the Apache web server contains entries like

```
[Mon Jul 09 17:14:25 2012] [error] [client 10.10.10.20] python:
Objects/stringobject.c:4098: str_subtype_new: Assertion `((((PyObject*)(tmp))-
>ob_type) == &PyString_Type)' failed., referer:
http://<hostame>/bscw/bscw.cgi/48?op=addfolder&id=48


[Mon Jul 09 17:14:25 2012] [error] [client 10.10.10.20] Premature end of script
headers: bscw.cgi, referer: http://<hostname>/bscw/bscw.cgi/48
```

The problem outlined above **has been fixed** by Red Hat beginning with version **6.4**. So it is advisable first to upgrade to version 6.4 (or higher).

If you still experience this kind of crashes, you may compile your own Python (2.7) interpreter, install it in `/usr/local` (to not interfere with the system Python 2.6 interpreter) and upgrade your BSCW server to Python 2.7.

RHEL Python 2.7 installation

1. Install the following required (development) RPM packages including required dependencies

   ```
   yum install gcc g++ make patch
   yum install db4-devel
   yum install sqlite-devel
   yum install openssl-devel
   yum install bzip2-devel
   yum install readline-devel
   yum install gdbm-devel
   yum install tk-devel
   ```

2. Download and extract the latest Python 2.7 source distribution

   ```
   wget http://python.org/ftp/python/2.7.6/Python-2.7.6.tgz
   md5sum Python-2.7.6.tgz
   bcf93efa8eaf383c98ed3ce40b763497  Python-2.7.6.tgz
   tar xf Python-2.7.6.tgz
   ```

3. Run the "configure" script with the given options and compile the sources

   ```
   cd Python-2.7.6
   export LDFLAGS="-L\${prefix}/lib -Wl,-rpath=\${prefix}/lib"
   ./configure \
       --enable-shared \
       --enable-ipv6 \
       --with-signal-module \
       --with-threads \
       --with-pymalloc
   make
   ```

4. Install the Python 2.7 interpreter environment (without interfering with other Python distributions)

   ```
   make altinstall
   ```

Keywords: “Internal Server Error”, RHEL

10.3 BSCW Server Administration

10.3.1 What facilities are available for server administrators?

BSCW provides a HTML and a command line interface for server administration.

To be able to access the HTML administration interface with `[Options > Admin]`, you must have an account on the BSCW server and your account name must appear in the `SERVER_ADMINS` list in the main server configuration file (`<bscw-runtime-path>/conf/config.py`).

Administrator users explicitly need to log in a second time with their password at `[Options > Admin]` to gain BSCW administrator rights. Without this additional administrator authentication no administrative rights are applied to their account.

The administrative command line interface is accessed via the `bsadmin` script which is located in the BSCW server instance path "bin" directory `<bscw-runtime-path>/bin/bsadmin`. Enter '`bin/bsadmin`' to get a list of all installed administrative modules or '`bsadmin <command>`' for instructions about the usage of a specific tool (see also section 7.2).

Keywords: Admintools, administrator interface, bsadmin scripts

10.3.2 How do I delete a user from the BSCW server?

Open the `[User administration]` page of the HTML administration interface `[Options > Admin]`. Find the respective user and select `[Destroy]`.

Using the command line interface '`bsadmin rmuser`' destroys a given user name.

Keywords: User administration, delete a user from server

10.3.3 How do I rename a user?

Open the `[User administration]` page of the HTML administration interface `[Options > Admin]`. Find the respective user and select `[Rename]`.

Using the command line interface '`bsadmin rename`' renames a given user name.

Keywords: User administration, rename a user

10.3.4 How do I register a new user (i.e. without sending email)?

This is possible through the "New User" page of the HTML administration interface. Enter the email address and then allocate the address to a new user name with a password.

The command line interface provides the '`bsadmin register`' script, use the following syntax to register a new user:

```
$ bin/bsadmin register -r <email> <login_name> <password>
```

Keywords: User administration, register new users

10.3.5 How do I restrict the creation of workspaces?

Workspaces are "created" by adding members to a folder. To disallow an user the creation of new workspaces her/his role may not contain actions from the "share view" so s/he is not able to invite members. Hence to effectively deny the creation of new workspaces requires a change of the "user role", which is by default the "Manager" role.

An BSCW administrator may enforce such a restriction in two ways:

1. To restrict single users edit the "user role" of or assign a new "user role" to her/his user object. The user role is inherited by the users' top level folders (home, clipboard, etc.) along the folder hierarchy:
   - Open the user objects info page:
     `http://<your server>/bscw/bscw.cgi/u<login_name>`
   - Edit the (default) user role "Manager" and select the actions you want to restrict/allow with
     `[> Access > Edit roles]`
   - Alternatively you may assign a more restrictive role to the user with
     `[> Access > Assign roles]`
2. If you want to generally disallow users to create workspaces it is advisable to define a server-wide more restrictive user role, see administration manual, section 7.8.2 for details.

Keywords: restrict user actions, restrict creation of new workspaces

10.3.6 How do I restrict the creation of new user accounts?

By default the BSCW server allows generally self-registration of email addresses and the creation of BSCW user accounts.

The `MAY_REGISTER` list in the main server configuration file `<bscw-runtime-path>/conf/config.py` restricts the ability to register new email addresses to the listed BSCW users. If the `MAY_REGISTER` list is not empty, only the listed users (beside BSCW administrators) are allowed to create new email addresses using the `[Access > Invite Member]` action (see also the `RESTRICT_MAIL` in `<bscw-runtime-path>/conf/config.py` for further methods to restrict registration.)

Keywords: restrict user account creation

10.3.7 How do I find the corresponding file for a BSCW document?

While the meta data of a BSCW document is kept in the database, the raw document itself is stored within the file system in a directory tree below the directory defined by `FILES` (in the main server configuration file `<bscw-runtime-path>/conf/config.py`) which points by default to '`<bscw-runtime-path>/var/data/Files/`'.

In general documents are named with an unique identifier assigned by the BSCW system at creation time. To store the raw document this unique identifier is split into number pairs (from the right to the left; if necessary padded with a leading zero) and copied in the corresponding `FILES` sub directory. The file name of the raw document is constructed by the left most number pair with the character '`F`' and the document type extension appended. For example, the content of a Word document with unique identifier `12345` is stored in a file named:

```
<FILES>/01/23/45F.doc
```

You may retrieve meta-information on a document using the '`bsadmin ls`' utility. To get information on the above document use

```
$ bin/bsadmin ls <FILES>/01/23/45F.doc
```

Keywords: BSCW document, document raw file

10.3.8 May I remove the contents of the BSCW "Temp" directory?

The BSCW "Temp" directory (`<bscw-runtime-path>/var/data/Temp` by default) holds temporary files and directories created during database updates and document uploads. **Before removing any**

files from Temp, shut-down the BSCW database server. After shut-down, all files or directories in the "Temp" may be removed.

Keywords: temp directory, remove files from temp-directory

10.3.9 How do I upgrade my BSCW server instance to a new version?

1. **Important**: Read **attentively** the upgrade hints in section 2.3. To perform an upgrade you need a **valid** BSCW license! Do not upgrade if your license has become invalid!
2. Unix:
   - Download and extract the BSCW distribution archive `bscw-5.0.9-3????-py26.tar.gz`

     ```
     $ tar xzf bscw-5.0.9-3????-py26.tar.gz
     ```
   - Enter the distribution directory `bscw-5.0.9-3????-py26` and perform the usual installation steps (see section 3.2) **on top** of your old BSCW instance in `<bscw-runtime-path>`. To start the installation extract the BSCW distribution archive and run the `install.sh` script as superuser:

     ```
     # id
     uid=0(root) gid=0(root) groups=0(root)
     # tar xf bscw-5.0.9-3????-py26.tar.gz
     # cd bscw-5.0.9-3????-py26
     # ./install.sh

     Enter BSCW system user name: [bscw]
     Enter BSCW base directory: [/opt/bscw]

     Extracting BSCW 5.0.9 distribution in /opt/bscw/lib

     Choose one of the following options:
     ( 0) update BSCW 5.0.7 [/opt/bscw/srv/bscw.domain.org]
     ( 1) create new BSCW instance
     Enter a number (0-2): 0
     ...
     ```
   - Adopt your Apache HTTP server settings (see section 3.3.1);
   - Edit the BSCW main server configuration file `<bscw-runtime-path>/conf/config.py` and adopt it to your needs, e.g. enable new features (be sure to configure the mandatory settings section (see section 3.3.2)).
3. Windows 7, 8/Server 2012, 2008, 2003:
   - Download and execute the BSCW distribution installer `bscw-5.0.9-3????-py26.exe`
   - Choose the BSCW instance you want to upgrade and follow the configuration dialog (see section 4.2)
   - Adopt your HTTP server settings if you are using Apache HTTP server (see section 4.4.2);
   - Edit the BSCW main server configuration file `<bscw-runtime-path>/conf/config.py` and adopt it to your needs, e.g. enable new features
4. If your license got invalid apply for a "change license":
   - Make sure you are BSCW administrator (if needed, insert your user name in `<bscw-runtime-path>/conf/config.py`: `SERVER_ADMINS`) and open

     ```
     [Options > Admin]
     ```

Log in a second time with your password to gain BSCW administrator rights for the current session and apply with

```
[Options > Admin > Upgrade license]
[x] new license
[OK]
```

- Fill in the form (be sure to enter a **valid** email-address!)
- Choose the license type:
  `"Change license for new server (royalty free)"`
- Please print and fax the shown license agreement to us.

Keywords: server upgrade, new version

10.3.10 How do I migrate a BSCW database to another host?

Note: BSCW servers version 3.2 or later must have a valid license before the migration (resp. upgrade). **If the license is not valid or is an evaluation license, you need to upgrade your license** before migrating.

The procedure is as follows:

1. Install the same BSCW server version in `<bscw-runtime-new>` on your destination host
   - edit `<bscw-runtime-new>/conf/config.py`:
     - adopt the import configuration settings of your old server (e.g. `<bscw-runtime-old>/conf/config.py`: `SERVER_ADMIN`, `SERVER_ADMINS`, `SMTP_HOST`)
     - set `SERVER_ROOT = 'https://<bscw.domain.org>/'`
       (see `<bscw-runtime-new>/README.txt`)
   - check if your newly installed BSCW server is fully operational
   - stop your new BSCW server.
2. Copy the old BSCW server (in `<bscw-runtime-old>`) data to you new BSCW server (in `<bscw-runtime-new>`)
   - stop your old BSCW server (in `<bscw-runtime-old>`)
   - copy the content of the `<bscw-runtime-old>/var/data` directory into the `<bscw-runtime-new>/var` directory of your new BSCW server.
   - start your new BSCW server (in `<bscw-runtime-new>`)
3. Make sure you are BSCW administrator (if needed, insert your user name in `<bscw-runtime-new>/conf/config.py`: `SERVER_ADMINS`) and open
   `[Options > Admin]`
   log in a second time with your password to gain BSCW administrator rights for the current session and press `[OK]`. Now apply for a new license with
   ```
   [Options > Admin > Upgrade license]
   [OK]
   ```
   Fill in the form (be sure to enter a **valid** email-address!)
   Choose the license type:
   `"Change license for new server (royalty free)"`
   Please print, sign and fax the shown license agreement to us.
4. As soon your license is granted you will receive an email notification.
   - follow the mentioned URL and perform a garbage collection.

Attention: To perform a migration the Python version installed at the target machine must be at least the same Python version. I.o.w. a migration from Python version 2.7 to Python version 2.6 is not possible.

Keywords: migrate database

10.3.11 Why do I get a "license expired" error?

You may get one of the following types of errors:

- The BSCW server responds with

  ```
  Error: license expired
  Cannot commit changes to database because the BSCW license has expired
  Error code: unauthorized
  ```

  In this case your BSCW database does not contain a valid BSCW license (e.g., you upgraded a BSCW server before version 3.2). To install a BSCW test license (90 days for 200 users) run the garbage collector.

- The BSCW server responds with

  ```
  Error: license expired
  Cannot commit changes to database because the BSCW license has expired
  Error code: ... <some message different from 'unauthorized'>
  ```

  Your BSCW license is invalid (a more descriptive reason is shown in the error code message). In this case you have apply for a new license. Use the "Upgrade license" operation in the administrator interface.

Keywords: BSCW-license, license expired

10.3.12 The BSCW server does not work, the database seems to be corrupted

Your database seems to be corrupted! This may only happen, if there is a (disk) hardware failure or your BSCW disk partition is overflown. A corrupted BSCW database is typically indicated by one (or all) of the following Messages (see `<bscw-runtime-path>/var/log/bscw.log`):

1. The BSCW server reports the following traceback to a client:

   ```
   Unanticipated Error:
   Traceback (innermost last):
     [...]
   TypeError: unsubscriptable object
   ```

2. The garbage collector reports the following traceback:

   ```
   GC init:
   GC started: objects: 1767 size: 1485369
   Bad object 1663 at 1468966
   Traceback (innermost last):
     [...]
   RuntimeError: Bad objects in database
   ```

3. The BSCW database server reports the following traceback

   ```
   Traceback (innermost last):
     [...]
   EOFError: EOF read where object expected
   ```

4. The BSCW database server reports the following error

   ```
   $ bin/bsadmin start
   Service start bs_servdb at ('localhost', 12964)
   FATAL ERROR. Server stopped
   ```

```
exceptions.ValueError at 1368966 (size 1485369): bad marshal data
```

5. The BSCW database server reports some other strange things in the `<bscw-runtime-path>/var/log/bscw.log` file.

The recommended fix is replacing the BSCW database (the file `var/ata/Store`) by some backup file. Use the following commands with extreme care to avoid any data loss. **Back up your database storage files!** If in doubt ask support@orbiteam.de for further advice!

1. on Unix systems:

```
$ bin/start_servers -k
$ bin/bsadmin getconfig RESTORE
<bscw-runtime-path>/var/data/Store      # database store
$ cp var/data/StoreA var/data/StoreA.bak
$ cp var/data/StoreB var/data/StoreB.bak
$ rm var/data/StoreA var/data/StoreB
$ rm -f var/data/Tables
$ cp var/data/Backup var/data/Store
$ bin/start_servers
```

2. on Windows:

```
> bin/bsadmin stop
> bin/bsadmin getconfig RESTORE
<bscw-runtime-path>\var\data\Store      # database store
> copy var\data\StoreA var\data\StoreA.bak
> copy var\data\StoreB var\data\StoreB.bak
> del  var\data\StoreA var\data\StoreB
> del  var\data\Tables
> copy var\data\Backup var\data\Store
> bin/bsadmin start
```

If your back-up is outdated, or the back-up files are corrupted either, you may fix the database by truncating corrupted objects using the command

```
bin/bsadmin dbscan
```

This command will print the offsets and class names of the last objects in the database. A good choice for truncation will be the offset of the last `AccessCount` or `Preference` object. Transactions in BSCW are normally finished by writing a bunch of `AccessCount` or `Preference` objects. The database will not become inconsistent if some of these objects are missing. However you **may not** truncate at an offset lower than the file size after the last garbage collection (see `<bscw-runtime-path>/var/data/bscw.log`).

For database truncation use on Unix

```
$ bin/start_servers -k
$ bin/bsadmin getconfig STORE        # get active database store
<bscw-runtime-path>/var/data/StoreA
$ cp var/data/StoreA var/data/StoreA.bak
$ bin/bsadmin dbscan -f offset
$ bin/start_servers
```

or on Windows:

```
> bin/bsadmin stop
> bin/bsadmin getconfig STORE        # get active database store
```

```
<bscw-runtime-path>\var\data\StoreA
> copy var\data\StoreA var\data\StoreA.bak
> bin/bsadmin dbscan -f offset
> bin/bsadmin start
```

The parameter offset needs not to be given, if the last object in the database is an `AccessCount` or a `Preference`. Otherwise, the best value for offset is the number shown before the last `AccessCount` or `Preference` object.

Keywords: BSCW database corruption

10.3.13 Why do I get connect problems during "Upgrade License"?

You are probably sitting behind a firewall which does not let you connect to our license server. Here is what to do:

1. Use the `[Upgrade license]` button, but now store the returned page on your locally using `[Save as]` in your browsers file menu. For example, store the page in file "`form.html`".

   Alternatively you may use the command line script `bsadmin license -r` which creates a file `<bscw-runtime-dir>/var/data/Temp/license.html`.

The next 2 steps must be performed on systems which can connect to our server http://bscw.orbiteam.de.

2. Open the previously stored "`form.html`" resp. "`license.html`" page on a system with internet access, select `[New license]`, choose the required license and submit the form.
3. If necessary print, sign, and send (fax) the resulting license agreement to OrbiTeam (`license@orbiteam.de`).
4. After your license is granted you will be notified by email. Open again the stored "`form.html`" page again on a system with internet access and select `[Get license]`. Then save the returned page on your local system (e.g. in file `lic.html`).
5. The last step again needs connection to your BSCW server (the one behind the firewall):
6. Open the stored license (URL `file:lic.html`) on a system with access to your BSCW server and select `[Upload license]`.

Keywords: "Upgrade license", connect problems, firewall

10.3.14 My BSCW database seems to be corrupt, what can I do?

If your BSCW database is corrupt, e.g. due to hardware failure, your BSCW database server can be enabled to do an "auto-repair" - version 3.4 onwards only!

1. Stop the database server
   ```
   $ bin/bsadmin stop
   ```
2. **Important**: Back up your database storage files!
   ```
   $ bin/bsadmin getconfig STORE
   <bscw-runtime-path>/var/data/StoreA    # active database store
   $ mkdir var/data/backup
   $ cp var/data/Backup var/data/backup
   $ cp var/data/StoreA var/data/backup
   $ cp var/data/StoreB var/data/backup
   ```

Note: the command "`bsadmin getconfig STORE`" will return the active database store (`StoreA` or `StoreB`) while "`bsadmin getconfig RESTORE`" will return the current value of variable "`STORE`" in your configuration file

- Set error condition (remove the table file if existing and create a file with "Error" appended instead)

```
$ tables=`bin/bsadmin getconfig TABLES`
$ rm -f "$tables"
$ echo > "$tables"Error
```

- Start the database server (auto-repair is enabled) **after you made a backup copy of your database storage files**

```
$ bin/start_servers
```

You might also use "`bsadmin start`" here.

- Check for inconsistencies
- To avoid user interferences set in "`<bscw-runtime-path>/conf/config.py`" `SYS_BUSY = 'sys_busy'` and repeat the following two steps until no errors are reported (there should be only a **few** repairable errors)

  BSCW >= 4.3:

```
$ bin/bsadmin dbcheck list
$ bin/bsadmin dbcheck repair
```

  BSCW < 4.3:

```
$ bin/bsadmin dbcheck
$ bin/bsadmin dbcheck -r
```

- Finally, if everything seems ok, set again in "`<bscw-runtime-path>/conf/config.py`" `SYS_BUSY = ''` and start the garbage collection

```
$ bin/bsadmin garbage
```

Keywords: BSCW database corrupt, database problems

10.3.15 Why do I get an error, when archiving at a BSCW server with IIS 6.0?

The Microsoft Internet Information Services (IIS) 6.0 do not grant the anonymous account (`IUSR_<computer_name>`) the necessary file system permissions to run programs. Therefore, the external archive commands cannot be executed.

To solve this problem, grant the anonymous account 'Read and Execute' access to `Cmd.exe` (by default `C:\Windows\System32\Cmd.exe`). This program is used by Python to start external commands.

For more information about this problem visit the Mircosoft Knowledge Base at

http://support.microsoft.com/?id=311481

Keywords: Windows 2003, IIS 6.0, archive errors

10.3.16 How can I upload files larger than 100MB when using IIS?

The Microsoft Internet Information Services (IIS) limits the upload size (for HTTP POST) to 30 MB by default. The BSCW installer increases this limit to 100 MB. To further increase this limitation the `maxAllowedContentLength` parameter must be set to a higher value. Run for a DOS shell `bsadmin conf_iis -m <maxAllowedContentLength>` (in Bytes).

Keywords: Windows, IIS, upload size limitation

10.3.17 Why can't BSCW provide WebDAV with Microsoft IIS Web server?

For some unknown reason the Windows WebDAV client sends within WebDAV HTTP requests an undocumented header '`translate: f`'. After receiving this header IIS does not execute the BSCW CGI-script. With other WebDAV clients you get access to a BSCW server via WebDAV under Windows with IIS.

To solve this problem, install your BSCW server using the Apache HTTP server 2.2 or 2.4 for Windows.

Keywords: WebDAV, Windows, IIS

10.4 BSCW Installation

10.4.1 What do I need to install the BSCW server software?

Generally you require a standard Web server (we recommend the Apache HTTP server 2.2 or 2.4 for Unix and Windows, which can be downloaded from `http://httpd.apache.org/`). You also require the interpreter and standard libraries for the Python programming language. The Python implementation is copyrighted, but is freely usable and can be downloaded from http://www.python.org/

Keywords: Prerequisites, BSCW server software, BSCW server installation

10.4.2 Where should I install the BSCW server software (Unix)?

The installation program of the BSCW software **must** be run as superuser (root). The BSCW install procedure will create a special BSCW system user `bscw` with an **own** group `bscw`.

It doesn't matter where you install the BSCW software, but it is necessary that your Web server have access to the file system where BSCW is installed. For best results put it on a file system that is local to the host where your Web server runs.

Keywords: BSCW server software, BSCW server installation, operating system, Unix

10.4.3 Why do I get a "500 Server Error" when I try to register myself?

When you try to register, i.e., when you go to location

```
http://<server>/pub/bscw.cgi?op=rmail
```

and you receive "`500 Server Error`" your Web server failed to start the BSCW '`bscw.cgi`' CGI script. Check that your Web server runs on the same host as the BSCW database server (started with the '`start_servers`' script) - both servers **must** run on the same host.

Also check that the paths to the BSCW CGI script `bscw.cgi`, the Python interpreter (usually `/usr/bin/python`) and the Python libraries (usually `/usr/lib/python2.?/`) are accessible from the server host machine for the user (group) ID that the Web server uses to execute CGI scripts.

Keywords: BSCW-registration, register, 500 Server Error,

10.4.4 Can I put the data files for the server on a separate disk?

On Unix systems you can change in `<bscw-runtime-path>/conf/config.py` the location of various data files by appending the directory definition for `ALARM_DIR`, `DATA_DIR`, `LOG_DIR`, `RUN_DIR`, `WWW_DIR`.

Note: if you provide relative paths directories are relative to `<bscw-runtime-path>`.

Keywords: data files, separate disk, operating system

10.4.5 What can I do if I get a "ServiceException: getState, ()" error after "start_servers"?

- In `<bscw-runtime-path>/conf/config.py` change (or add a line):
```
GSMOD_CAN_FLUSH = 0
```
- Stop the running database server:
```
$ bin/start_servers -k
```
- Check if stopping the database server was successful:
```
$ ps -ax | grep bsadmin
```
- There should be no process "`./bin/bsadmin start ...`" running. Otherwise manually kill this process:
```
$ kill <pid of bsadmin process>
```
(Be careful if you have other BSCW servers running on your machine that you don't want to kill.)
- Start database server
```
$ bin/start_servers
```

Keywords: `ServiceException`, start_servers

10.4.6 How can I provide user interface in different languages on our BSCW server?

BSCW was designed to allow installation of different language interfaces. Therefore server code and language dependent message files have been separated. All message files reside in a sub directory of the BSCW distribution `bscw-5.0.9-3????-py26/bscw/msg` (e.g. `msg/(de|en|es|fr)` come with the server distribution). Support for additional languages can be added by

- creating a sub directory in the BSCW instance directory `<bscw-runtime-path>/bsext/msg/??`
- copying all files from `bscw-5.0.9-3????-py26/bscw/msg/en` to the new directory
- translating all files in the new directory to the desired language

For more information please read section 7.1 of this manual. Please note that up to date information on available languages can be found at `http://www.bscw.de/english/extensions.html`. If you translated the BSCW user interface to your language, please send an email to `support@orbiteam.de` - we would like to provide it to all users of the BSCW system.

Keywords: user interface, languages, message files

10.4.7 Why do I get a "No module named crypt" error?

Your Python interpreter does not contain the crypt module. Therefore, you have to enable the crypt module in P`ython-2.?.?/Modules/Setup` and rebuild Python.

Keywords: No module named crypt

10.4.8 Why do I get "%1 is not a valid Windows application" error?

When you try to access the BSCW server you may get this error message. For some reason the application mapping for the extension '`.cgi`' is missing. Please install BSCW again or set it manually to:

```
'.cgi' : "\python.exe" -u "%s"
```

Keywords: operating system "%1 is not a valid Windows application", GGI scripts

10.4.9 Why do I get a "Permission denied" error? (Unix)

The path to your BSCW instance directory, the `<bscw-runtime-path>/var/www` directory and all directories below these directories must be readable and executable (searchable) for all users (e.g. mode `drwxr-xr-x`). The scripts `var/www/*.cgi` additionally must have the set-group bit set (e.g. mode `-rwxr-sr-x`). All other files below these directories must be readable for all. This is, because the HTTP server must have the right to find and execute the CGI scripts and to return icons and other public objects.

The scripts (or a wrapper program) will then set the effective group for further access to BSCW operations. All data below the BSCW installation directory should be readable by this group. This group needs also write access to the `var/data/` directory and all files and directories below that.

Access right problems like

```
Traceback (innermost last):
    [...]
OSError: [Errno 13] Permission denied: 'var/data/Files/01/23'
```

are caused by an erroneous installation of the `bscw.cgi` CGI script (no binary wrapper is installed, the script is not executed set-group-id of the BSCW users' group; the BSCW instance-path file system is mounted no-suid) or by incorrect manual manipulation of the BSCW instance-path access rights.

BSCW requires group-write permissions (therefore it requires an own exclusive group. Please check the section 3.2 of this manual for correct BSCW user and group setup.

Execute as BSCW user '`bscw`' with the group `bscw` the `bsadmin chkconfig` script:

```
# su bscw
$ id
uid=1234(bscw) gid=1234(bscw)
$ cd <bscw-runtime-path>
$ ./bin/bsadmin chkconfig
```

This should compile (if a compiler is found) and install a binary wrapper. If no compiler is found compile the wrapper manually and repeat `bsadmin chkconfig`. The `bscw.cgi` CGI script **must** run set-group-id and the complete BSCW `./var/data` directory needs `rws`-group access.

To fix erroneous file permission stop your BSCW server and perform the following commands (as root)

```
# cd <bscw-runtime-path>

# ./bin/start_servers -k

# chown -Rh bscw:bscw .

# find ./var/[^w]* -type d | xargs chmod 2770
# find ./var/[^w]* -type f | xargs chmod 660

# find ./var/www -type d | xargs chmod 2775
# find ./var/www -type f | xargs chmod 664
# chmod 2755 .

# chmod 2755 ./var/run/run_bscw
# chmod 2755 ./var/www/bscw.cgi
# chmod 2755 ./var/www/nj_bscw.cgi

# chmod 2771 ./var/data
# chmod 644  ./var/data/htpasswd
```

To avoid world read-access on `./var/data/htpasswd` (or `./var/data/Temp`) the file (or directory) alternatively must be owned by the webserver user (see the '`User`' directive in the main HTTP server configuration file).

```
# chown www-data     ./var/data/htpasswd
# chmod 660          ./var/data/htpasswd

# chown -Rh www-data ./var/data/Temp
# find ./var/data/Temp -type d | xargs chmod 2770
```

Since Linux environments do not execute forked processes set-group-id, archiving may not work anymore. To create archives it is necessary to recursively change the owner the `./var/data/Files` directory to the web server user.

```
# chown -Rh www-data ./var/data/Files

# ./bin/start_servers
```

Keywords: Permission denied, HTTP server, OSError, Unix

10.4.10 Why do I get a Python trace with "`RuntimeError: var/www/bscw.cgi: No setgid`"?

If your operating system does not support 'set-group-id' *scripts* (such as Linux) you have to use a binary wrapper program to allow 'set-group-id' operation of the `bscw.cgi` script. If your operating system supports 'set-group-id' *scripts* (such as Solaris), this problem is caused by a file mode/ownership problem.

Usually the BSCW CGI script (`<bscw-runtime-path>/var/www/bscw.cgi`) is executed with group ID set to the BSCW user:

```
$ cd <bscw-runtime-path>/var/www
$ ls -l bscw.cgi
-rwxr-sr-x 3 bscw bscw 771 Feb 21 13:12 bscw.cgi
```

Using this technique enables the BSCW CGI script (independently of the user and group ID setting of the executing HTTP server) to modify its database located in directory `<bscw-runtime-path>/var/data`:

```
$ cd <bscw-runtime-path>/var
$ ls -ld data
drwxrws--- 4 bscw bscw 512 Feb 21 14:05 data
```

The problem should be solved by changing file ownership and modes (using user and group ID of the BSCW user) as described in FAQ question 9.4.9

Keywords: Python traceback, `RuntimeError`, CGI scripts, operating system