## CONTENTS

1 How to read this Manual 1

2 Installation of the BSCW server 3
   2.1 General Requirements ................................. 3
   2.2 Security considerations ................................. 4
   2.3 EU - General Data Protection Regulation .................. 4
   2.4 Upgrading to BSCW 7.3.2 ................................. 5
      2.4.1 Upgrading on Unix ................................. 13
      2.4.2 Upgrading on Windows ............................... 17

3 Installation procedure for Unix 19
   3.1 System requirements .................................. 19
   3.2 Installation ........................................... 20
   3.3 Software for BSCW Preview .............................. 25
   3.4 Configuration .......................................... 29
      3.4.1 Apache HTTP Server Configuration ................. 29
      3.4.2 BSCW instance configuration ....................... 34
      3.4.3 Administrator account .............................. 35
      3.4.4 De-Installation .................................. 36
   3.5 Database Server Startup, Garbage Collection and Backup ... 36
      3.5.1 BSCW Startup ...................................... 37
      3.5.2 Garbage Collection ................................ 37
      3.5.3 Backup ........................................... 37
   3.6 Folder Mail Delivery ................................ 38
      3.6.1 BSCW mail delivery agent (MDA) ..................... 38
      3.6.2 Local Mail Transfer Agent (MTA) .................... 39

4 Installation procedure for Windows 43
   4.1 System requirements .................................. 43
   4.2 Installation and Configuration ......................... 44
   4.3 Software for BSCW Preview .............................. 54
   4.4 Database Server and Garbage Collection ................. 57
      4.4.1 Windows Service ................................... 58
      4.4.2 Task Scheduler ..................................... 58
   4.5 Further Configuration Details ......................... 60
      4.5.1 BSCW Server Root Definition ....................... 60
      4.5.2 Apache HTTP Server Configuration .................. 60
      4.5.3 IIS Configuration ................................ 64
      4.5.4 De-Installation .................................. 67
   4.6 Folder Mail Delivery ................................ 67

5 Configuration of BSCW Servers 69
   5.1 Authentication .......................................... 70
   5.2 conf/config.py ......................................... 71
      5.2.1 MANDATORY server settings ......................... 71
5.2.2 Mail handling ............................................. 75
5.2.3 Server access ........................................... 77
5.2.4 web/proxy server settings ......................... 83
5.2.5 BSCW appearance settings ....................... 91
5.2.6 Optional BSCW packages .......................... 98
5.2.7 BSCW database server settings ................. 103
5.3 conf/config_actions.py ............................... 109
5.4 conf/config_applet.py ................................. 109
5.5 conf/config_cal.py .......................... 110
5.6 conf/config_clientmap.py ............................ 110
5.7 conf/config_controls.py .............................. 110
5.8 conf/config_convert.py ............................... 110
5.9 conf/config_countries.py ......................... 112
5.10 conf/config_grid.py ................................. 112
5.11 conf/config_guided_tours.py ..................... 113
5.12 conf/config_help.py ................................. 113
5.13 conf/config_icon.py ................................. 113
5.14 conf/config_icons.py ............................... 113
5.15 conf/config_meet.py ................................. 113
5.16 conf/config_menus.py ............................... 113
5.17 conf/config_mimemsg.py ............................ 114
5.18 conf/config_mimegroups.py ....................... 114
5.19 conf/config_mimeicons.py ....................... 114
5.20 conf/config_mimeicons.py ....................... 114
5.21 conf/config_mimetype.py ......................... 114
5.22 conf/config_mimeicon.py ......................... 115
5.23 conf/config_mimetype.py ......................... 115
5.24 conf/config_prio_categ.py ....................... 115
5.25 conf/config_portlets.py ......................... 115
5.26 conf/config_quicksearch.py .................... 115
5.27 conf/config_search.py ............................. 115
5.28 conf/config_service.py ......................... 115
5.29 conf/config_service.py ......................... 115
5.30 conf/config_styles.py ............................ 115
5.31 msg/<lang>/lg_msgconfig.py ................... 116

6 BSCW Packages ........................................... 117

6.1 Content Search PyLucIndex ......................... 118
6.1.1 Configuration ...................................... 118
6.1.2 Command line tools .............................. 121
6.1.3 Index creation and update .................... 123
6.2 LDAP .................................................. 124
6.2.1 Installation ...................................... 124
6.2.2 LDAP Browser .................................... 127
6.3 Desktop Widgets ..................................... 127
6.4 Document Approval .................................. 128
6.5 Blog (Weblogs) ...................................... 128
6.6 Case – File Synchronisation ....................... 128
6.7 Chat .................................................. 129
6.8 Expire ............................................... 129
6.9 Export PDF ........................................... 130
6.10 Flow-Folder ........................................ 130
6.11 Http ................................................ 130
6.11.1 Enabling the BSCW HTTP server ........... 131
6.11.2 Disabling the BSCW HTTP server .......... 132
6.12 Incognito ............................................ 132
6.13 Metaprofiles ....................................... 133
6.14 Online Office ....................................... 133

6.14 Online Office ....................................... 133
# BSCW Help

## Administration of BSCW Servers

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administration using the Web Interface</td>
<td>143</td>
</tr>
<tr>
<td>BSCW status page</td>
<td>145</td>
</tr>
<tr>
<td>BSCW Access Management</td>
<td>146</td>
</tr>
<tr>
<td>Configuration</td>
<td>149</td>
</tr>
<tr>
<td>BSCW licence management</td>
<td>150</td>
</tr>
<tr>
<td>Administration using the bsadmin script</td>
<td>152</td>
</tr>
<tr>
<td>User administration</td>
<td>154</td>
</tr>
<tr>
<td>User status with bsadmin users</td>
<td>155</td>
</tr>
<tr>
<td>User registration with bsadmin register</td>
<td>156</td>
</tr>
<tr>
<td>User management with bsadmin (rename</td>
<td>chpwd</td>
</tr>
<tr>
<td>Additional anonymous users</td>
<td>158</td>
</tr>
<tr>
<td>Asynchronous Services</td>
<td>159</td>
</tr>
<tr>
<td>User Notification Services (UNO)</td>
<td>159</td>
</tr>
<tr>
<td>User account expiry</td>
<td>160</td>
</tr>
<tr>
<td>Automatic disk usage update</td>
<td>161</td>
</tr>
<tr>
<td>Public space deactivation</td>
<td>161</td>
</tr>
<tr>
<td>WebDAV</td>
<td>163</td>
</tr>
<tr>
<td>Microsoft Support for WebDAV</td>
<td>163</td>
</tr>
<tr>
<td>Known Problems</td>
<td>164</td>
</tr>
<tr>
<td>Quota - Disk Usage Limitation</td>
<td>164</td>
</tr>
<tr>
<td>Limit Classes</td>
<td>165</td>
</tr>
<tr>
<td>Quota Activation</td>
<td>166</td>
</tr>
<tr>
<td>Calculation of current disk usage</td>
<td>167</td>
</tr>
<tr>
<td>Report disk usage</td>
<td>167</td>
</tr>
<tr>
<td>Definition of Roles</td>
<td>168</td>
</tr>
<tr>
<td>The BSCW role concept</td>
<td>168</td>
</tr>
<tr>
<td>Role definition and default roles</td>
<td>170</td>
</tr>
<tr>
<td>Site-specific Roles</td>
<td>173</td>
</tr>
<tr>
<td>Site-specific banner</td>
<td>174</td>
</tr>
<tr>
<td>Server-wide template folders</td>
<td>175</td>
</tr>
<tr>
<td>Web Service API</td>
<td>176</td>
</tr>
<tr>
<td>Some useful hints</td>
<td>176</td>
</tr>
</tbody>
</table>

## BSCW Help

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Languages</td>
<td>179</td>
</tr>
<tr>
<td>Existing translations</td>
<td>179</td>
</tr>
<tr>
<td>Translation instructions</td>
<td>179</td>
</tr>
<tr>
<td>BSCW Updates</td>
<td>180</td>
</tr>
</tbody>
</table>

## BSCW license

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>License application</td>
<td>181</td>
</tr>
<tr>
<td>License changes</td>
<td>182</td>
</tr>
</tbody>
</table>

## Frequently Asked Questions (FAQ)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSCW Server Usage</td>
<td>183</td>
</tr>
<tr>
<td>What do I need to use BSCW?</td>
<td>183</td>
</tr>
<tr>
<td>Do I need a special application for uploading documents?</td>
<td>183</td>
</tr>
</tbody>
</table>
10.1.3 How is BSCW intended to be used ........................................... 183
10.1.4 I cannot log in. The server rejects me - what shall I do? .............. 184
10.1.5 How do I change my password? ............................................. 184
10.1.6 How do I configure my web browser? ..................................... 184
10.1.7 How do I connect to BSCW using WebDAV? ............................. 184
10.1.8 How do I destroy a workspace? ............................................. 187
10.1.9 How do I delete my account? ............................................... 187
10.1.10 How do I handle a BSCW error? ......................................... 187
10.1.11 I reached the limit of my disk space - what shall I do? .............. 188
10.1.12 Why does MS-Word mark a document as read-only? .................. 188
10.1.13 Is there a restriction for the size of documents I upload? ............ 188

10.2 BSCW Server Software .......................................................... 189
10.2.1 How do I get the BSCW software? ........................................ 189
10.2.2 Can I try the BSCW software? ............................................. 189
10.2.3 How do I keep up to date with BSCW developments and new releases? 189

10.3 BSCW Server Administration .................................................. 189
10.3.1 What facilities are available for server administrators? ............... 189
10.3.2 How do I delete a user from the BSCW server? ........................... 190
10.3.3 How do I rename a user? ................................................... 190
10.3.4 How do I register a new user (i.e. without sending email)? ........... 190
10.3.5 How do I restrict the creation of workspaces? .......................... 190
10.3.6 How do I restrict the creation of new user accounts? .................. 191
10.3.7 How do I find the corresponding file for a BSCW document? ........ 191
10.3.8 May I remove the contents of the BSCW “Temp” directory? .......... 191
10.3.9 How do I upgrade my BSCW server instance to a new version? ....... 192
10.3.10 How do I migrate a BSCW database to another host? ................ 193
10.3.11 Why do I get a “license expired” error? ................................ 193
10.3.12 Changing the “SERVER_ROOT” without service interruption ....... 194
10.3.13 The BSCW server does not work, the database seems to be corrupted 194
10.3.14 Why do I get connect problems during “Upgrade License”? ........ 196
10.3.15 My BSCW database seems to be corrupt, what can I do? .......... 196
10.3.16 How can I upload files larger than 100MB when using IIS? ......... 197
10.3.17 Why can’t BSCW provide WebDAV with Microsoft IIS Web server? 197

10.4 BSCW Installation ............................................................... 197
10.4.1 What do I need to install the BSCW server software? ................. 197
10.4.2 Where should I install the BSCW server software (Unix)? ............ 198
10.4.3 Why do I get a “500 Server Error” when I try to register myself? .... 198
10.4.4 Can I put the data files for the server on a separate disk? .......... 198
10.4.5 What can I do if I get a ServiceException: getState, () error .......... 198
10.4.6 How can I provide a BSCW user interface in different languages? 199
10.4.7 Why do I get a “Permission denied” error? (Unix) .................... 199
10.4.8 Why do I get a "RuntimeError: var/www/bscw.cgi: No setgid"? ... 201

Python Module Index ............................................................... 203
Index ......................................................................................... 205
Before installing your BSCW server you should read at least:

- the introduction to *Installation of the BSCW server* (in particular, section *Upgrading to BSCW 7.3.2* of chapter 2 if you are upgrading an BSCW instance),
- either section *Installation procedure for Unix* of chapter 3 or section *Installation procedure for Windows* of chapter 4, 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

- *Installation procedure for Unix* or
- *Installation procedure for Windows*

depending on your operating system.

**See also:**

Chapter 10: *Frequently Asked Questions (FAQ)*

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 *BSCW license*. If you have problems when upgrading your BSCW license, you should also have a look at the respective entries in *Frequently Asked Questions (FAQ)* or contact OrbiTeam (support@orbitteam.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 chapter 5 *Configuration of BSCW Servers* 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 also:**

Chapter 6 *BSCW Packages* 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 chapter 7 *Administration of BSCW Servers* and in chapter 10 *BSCW Server Administration.*
CHAPTER  
TWO

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 is supported on Linux and BSD - or a server host running Microsoft Windows 10, Server 2016/2019.

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 Intel Core/Xeon or a AMD EPYC/Opteron (>3.5 GHz) with at least 6 cores, 16 GB RAM and 500 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 website at http://www.python.org/). Therefore, besides the BSCW software, the installation of the BSCW server requires

• a Web Server
• the Node.js (v10 LTS) an event driven JavaScript runtime
• a Python 3.6 or 3.7 interpreter
• Python Jinja2 template engine
• (optional) extensions for Python (pylucene, ldap3)
• (optional) memcached to speed-up large folder handling (Unix)
• (optional) converter software for the BSCW preview feature, see sections 3.3 Software for BSCW Preview (Unix) or 4.3 Software for BSCW Preview (Windows) for details

The BSCW server can be installed on a CGI compliant Web server, e.g. the Apache HTTP server or the Microsoft’s Internet Information Server (IIS) (we recommend the Apache HTTP server version 2.4).

Additionally the Node.js runtime environment is required, which pushes occurring change notifications almost in real-time to active users, using a WebSocket-based events server. The Node.js runtime environment is freely available at the Node.js Foundation website (https://nodejs.org/en/download/). Best use the Node.js v10 LTS release.

Finally a Python 3 interpreter and the Python Jinja2 template engine is required to run the BSCW software. The Python interpreter is freely available on the Python Software Foundation website (https://www.python.org/downloads/). We currently support versions 3.6 or 3.7 of the Python interpreter only.

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 10 or Server 2016/2019. Therefore, the installation process for Unix and Windows is described separately in sections Installation procedure for Unix
and *Installation procedure for Windows* of chapters 3 and 4, respectively. You need to read only one of the two sections, depending on your platform.

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

**Note:** Please also consult chapter 10 *Frequently Asked Questions (FAQ)* in this manual - or the online version at [https://www.bscw.de/en/support/](https://www.bscw.de/en/support/) - 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

Newly installed BSCW instances do have the following possibly security relevant features enabled by 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 also:

   *BSCW database server settings* of chapter 5 for the description of the `ACCEPT_WEBSERVICES` directive page 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 inappropriate content). To disable the public space for all users see section 7.5 *Public space deactivation*.

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 from the environment of a running `bscw.cgi` process.

## 2.3 EU - General Data Protection Regulation

The General Data Protection Regulation (GDPR) (EU) 2016/679 is a regulation in EU law on data protection and privacy for all individuals within the European Union.

The GDPR aims primarily to give control to citizens and residents over their personal data and to simplify the regulatory environment for international business by unifying the regulation within the EU.
According to the GDPR BSCW has introduced the following measures:

- In order to simplify the information about processed personal data, the recording of events has been extended for user accounts accordingly.
- In addition to the user [i]nfo page on the web interface, for each user account an export of personal data is possible with the command line program `bsadmin userdata` in a machine-readable format (JSON).
- After the deletion of an user account, an audit log (history) of events is saved to an external file in the directory `<bscw-runtime-path>/var/data/rmuserarc` to document all processing operations (including deletion).

This audit log file is automatically removed at the end of the following year (see § 76 BDSG-Protokollierung (4)) by the `/etc/cron.daily/bscw` Cron script (see BSCW Startup for details).

With the legal validity of the EU - General Data Protection Regulation (GDPR), it will be necessary to provide a data protection declaration which sets out what personal information is collected during the use of the BSCW system and how these data are used.

You can find a template of a data protection declaration for BSCW on our website

https://www.bscw.de/files/dataprotection/Dataprotectiondeclaration-BSCW.docx
https://www.bscw.de/files/dataprotection/Datenschutzerklaerung-BSCW.docx

or in the BSCW software distribution in the directory `<HOME>/lib/bscw-7.3.2-<rev>-py3?/doc/dataprotection/`:

- Dataprotectiondeclaration-BSCW.docx
- Datenschutzerklaerung-BSCW.docx

Please edit the data protection declaration accordingly (e.g. adopt controller contact, organisation address and complete which data is required for BSCW registration) and create a PDF document which may be located in the BSCW runtime path `<bscw-runtime-path>/var/www` at:

```bash
$ cp Dataprotectiondeclaration-BSCW.pdf Datenschutzerklaerung-BSCW.pdf
```

Next define the runtime directives in the instance configuration file `<bscw-runtime-path>/conf/config.py`:

```python
PRIVACY_POLICY = '/pub/Dataprotectiondeclaration-BSCW.pdf'
PRIVACY_POLICY_DE = '/pub/Datenschutzerklaerung-BSCW.pdf'
```

**Note:** `/pub/` is the default "public" path, you can replace it by your SERVER_HOME (which is usually `/`).

If the `PRIVACY_POLICY` directives are defined the self-registration procedure requires new users to agree with this data protection declaration to complete the registration.

Afterwards recreate the BSCW instance index pages to reflect the changes with:

```bash
$ bin/bsadmin index_page
```

### 2.4 Upgrading to BSCW 7.3.2

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

**Note:**

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

```
$ ./bin/bsadmin manage_servers -l
/home/bscw/srv/bscw.domain.org:  BSCW 5.2.3
> bin\bsadmin manage_servers -l
C:\BSCW\srv\bscw.domain.org:  BSCW 5.2.3
```

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:

**Warning:** 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.
- 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.

See also:

*Upgrading on Unix* or *Upgrading on Windows* to consider the following advices when upgrading:

When **upgrading from BSCW 7.2.1 or lower**

(Security) The Apache HTTP configuration has been extended by a missing “Content Security Policy Header”. The upgrade procedure creates a new configuration for the Apache HTTP Server; please make sure to apply these changes from `conf/apache24/site.conf` to your current configuration.

For new BSCW instances the default values of BADPASS and ALLOW_MAIL_UNLOCK have been changed. For existing instances the values are not adopted, so it is recommended to manually set the values in `<bscw-runtime-path>/conf/config.py` to BADPASS = 10 and ALLOW_MAIL_UNLOCK = True.

When **upgrading from BSCW 7.1.0 or lower**

BSCW 7.3 release requires the installation of a Node.js runtime environment. See system requirements for Unix systems resp. system requirements for Windows systems.

A new “dark theme” has been introduced. To make it available to all users the THEMES list in the instance configuration file `<bscw-runtime-path>/conf/config.py` has to be extended with an 'default/default_dark' entry as follows:

```
THEMES = [
    'default/default',
    'default/default_dark',
]
```

When **upgrading from BSCW 7.0.0 or lower**

BSCW 7.3 release allows to upgrade previous versions. An upgrade is only possible with Python versions 3.6 or 3.7. Earlier Python 3 versions are not supported.

If you use Berkeley DB to maintain BSCW database tables (`DBMOD_TAB = 'bsddb4'`) you have to install the Python bindings for Berkeley DB (`bsddb3`):
• On Linux systems use the distribution package:
  – Debian based systems: python3-bsddb3
  – Fedora based systems: python3-bsddb3
• On Windows systems use pip3:

> pip3 install bsddb3

BSCW 7.3 does not support all (500+) features from BSCW 5 (or earlier) versions. If you are upgrading from a BSCW 5 version, it is advisable to run the `bsadmin migrationchecker` to see if all objects can be migrated. Please contact support@orbiteam.de for further assistance.

BSCW 7.3 requires after an upgrade to enable the `http` and the `chat` package with

```
$ cd <bscw-runtime-path>
$ bin/bsadmin package -e http
$ bin/bsadmin package -e chat
```

If the binary python package `setproctitle` is installed BSCW processes are displayed with more telling names. For BSCW 7.3 the `setprocfile` package is included for Python 3.6, 3.7 (Linux).

When upgrading from BSCW 5.2.3 or lower

BSCW 7.3 requires Python 3.6 or 3.7. Before upgrading BSCW to version 7 the Python 3 interpreter and all required packages must be installed. See system requirements for Unix systems resp. system requirements for Windows systems for a detailed description of required software packages.

Due to the exchange of document MIME icons, all style sheets for website folders must be adopted using [ → New → Style Definition].

When upgrading from BSCW 5.2.1 or lower

The IP address parsing for the `SERVER_ADMINS_IP` directive has been extended to support IPv6, prefix or netmask notation; old entries must be updated accordingly. Syntactically incorrect entries are ignored.

A new version (1.4.0) of the “ZOPE External Editor, BSCW Edition” with TLS 1.3 support is available at <https://www.bscw.de/classic/#externalEditor>.

When upgrading from BSCW 5.2.0 or lower

With the legal validity of the EU - General Data Protection Regulation (GDPR), it will be necessary to provide a data protection declaration with the BSCW system (see section 2.3 EU - General Data Protection Regulation for details).

When upgrading from BSCW 5.1.9 or lower

Please note, the upgrade procedure for BSCW 5.2 overwrites the default system message for new users in the `<bscw-runtime-path>/conf/msg/*/sys_msg0.html` files and recreates the converter configuration `<bscw-runtime-path>/conf/config_convert.py` file. If you are using a customized system message or an own converter configuration you may restore your old configuration from the most recent `<bscw-runtime-path>/conf/conf--<date>` directory after upgrading.

BSCW 5.2 introduces a new pre-forked HTTP server which greatly speeds up the processing of requests. Load tests have shown an average performance increase of 30% compared to the traditional Apache HTTP server CGI. To enable this feature, see the description in the `HTTP_LOCAL_PORT_START` directive section and the BSCW `http` package documentation.

BSCW 5.2 Classic introduces a new, modernized layout. To provide the old BSCW 5 layout of previous versions you may append the entry 'old' to the themes definition in the instance configuration file `<bscw-runtime-path>/conf/config.py`, e.g.:
Support for deprecated BSCW packages moin and SMS has been ended.

When **upgrading from BSCW 5.1.8 or lower**

The virus scan feature has been improved, virus scan is now performed asynchronously and additionally possible before download. After the upgrade virus scan must be re-enabled, please see `VIRUS_CHECK` for details.

Apache Tika server support has been added, please see for Unix systems section 3.3 (Apache Tika) and for Windows systems section 4.3 (Apache Tika) for details.

When **upgrading from BSCW 5.1.7 or lower**

Due to an internal reorganisation of object deletion, it might be necessary to fix changed workspace roles. Run `bsadmin dbcheck repair -f 'seeme undelete'` to fix affected access rights.

When **upgrading from BSCW 5.1.4 or lower**

(Unix) On systems using a systemd service to start BSCW instances, the service configuration must be changed before upgrading. The following configuration directives must be added to each unit files’ “[Service]” section, e.g.:

```
[Service]
GuessMainPID=no
RemainAfterExit=yes
```

After changing a unit file the systemd configuration must be reloaded with `systemctl daemon-reload` as root. The BSCW installer `install.sh` will check and notify you if these directives are missing in a BSCW unit file.

(Unix) Due to a text extraction bug not all uploaded documents are indexed properly by the BSCW “lucene” indexer (package `PyLucIndex`). To index affected documents run:

```
$ cd <bscw-runtime-path>
$ find var/data/Text -type f -size 0 | xargs rm
$ bin/bsadmin create_index -x
$ bin/bsadmin create_index -c
$ bin/start_servers
```

Additionally run to fix affected previews:

```
$ cd <bscw-runtime-path>
$ bin/bsadmin preview create -ff
```

When **upgrading from BSCW 5.1.3 or lower**

If you are using the BSCW “lucene” indexer (package `PyLucIndex`), a new sort order “by start” is enabled which requires to rebuild the index if users should be able to sort search results by this criteria. See section **Index creation and update** for a description how to create a new index.

When **upgrading from BSCW 5.0.10 or lower**

To support the new BSCW 5.1 preview feature additional converter software must be installed, please see section Software for BSCW Preview (3.3 (Unix) or 4.3 (Windows)) for details. To disable BSCW preview feature add a `CREATE_PREVIEWS = False` line to the instance configuration file `<bscw-runtime-path>/conf/config.py`. 

---

**THEMES** = ('bscw', 'bw', 'old')
The default path for authenticated access has been changed from /bscw/ to /sec/. The new /sec/ path is only applied for new installations, while upgraded instance keep the old path.

The ldap package configuration file <bscw-runtime-dir>/conf/ldap/config_ldap.py was (automatically) renamed to <bscw-runtime-dir>/conf/ldap/config.py during the upgrade.

The local instance package layout in <bscw-runtime-path>/bsext has been changed. If you created a local instance package please contact support@orbiteam.de before upgrading.

Python 2.6 support has been ended.

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.

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 user meta data to LDAP attributes. If you use the BSCW ldap package, please adapt 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-7.3.2-<rev>-py3?/bscw/conf/ldap/config.py.

Note: The update_bscw directive has been converted from a tuple to a dictionary

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 adapt the default role configuration) be aware the package might become disabled after an upgrade. If in doubt please ask support@orbiteam.de for advice on 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 Apache HTTP Server Configuration for Unix resp. Apache HTTP Server Configuration for 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 adapted 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 adapted, 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 web/proxy server settings of chapter 5 for details). After altering the authentication method bsadmin conf_apache -n, 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 version 3.6.2 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 bsadmin dbcheck repair -f m

**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 bsadmin update_defaults -s. This script will search the local system for archiver, encoder or converter commands.

See also:

Section 5.8 conf/config_convert.py

The Flow 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 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 adapted 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 has been adapted 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 Services (UNO) or the alarm service 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 Services (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 `bsadmin quota fix`.
2. PRO licensees may run alternatively the command `bsadmin quota report -vL`, which commits changes to the database after each user.

- The action `delete` changes the owner of an object: owner becomes the user who `deleted` the object (the object inherits the owner of the ambient folder (who is in this case the owner of 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 Services (UNO) which replaces the workspace activity report and the awareness service of previous BSCW versions. In order not to 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:

```ini
BSCW_LICENSE = 'http://bscw.orbiteam.de/pub/'
BSCW_LICENSE = 'https://bscw.orbiteam.de/pub/bscw.cgi/'
```

Important: Starting with BSCW 4.0.6 a new license mechanism was introduced. The new mechanism no longer binds 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.

See also:

Section 3.4.2 BSCW instance configuration for Unix or section 4.5.1 BSCW Server Root Definition 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 adapt your configuration to the new layout as given in `<bscw-runtime-path>/apache24/bscw.conf`.

See also:
Section 3.4.1 Apache HTTP Server Configuration for Unix or section 4.5.2 Apache HTTP Server Configuration or section 4.5.3 IIS Configuration 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 also:
Chapter 9 BSCW license

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 chapter 3 Installation procedure for Unix or chapter 4 Installation procedure 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.

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

2.4.1 Upgrading on Unix

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

2.4. Upgrading to BSCW 7.3.2
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 you 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

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

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

```bash
$ 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.

```bash
$ cd /tmp
$ tar xf bscw-7.3.2-<rev>-py3?.tar.gz
$ cd $HOME/lib
$ tar xf /tmp/bscw-7.3.2-<rev>-py3?/bscw-7.3.2-<rev>-py3?.tar
```

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

```bash
$ cd $HOME/lib/bscw-7.3.2-<rev>-py3?
$ python3 ./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: [/home/bscw]
Extracting BSCW 7.3.2 distribution in /home/bscw/lib
Choose one of the following options:
  ( 0) update BSCW 4.5.9 [/home/bscw/server]
  ( 1) update BSCW 5.2.3 [/home/bscw/srv/bscw.domain.org]
  ( 2) update other BSCW instance
  ( 3) create new BSCW instance
Enter a number (0-2): 1
```

```
target '/home/bscw/srv/bscw.domain.org' exists - checking...
stop pid=22823
2018-08-28 09:28:12 BSCW indexer (PyLucene 3.6.2) PID 22835 terminated
2018-08-28 09:28:12 Stopped bscw.adm.bs_servdb
2018-08-28 09:28:12 Stopped bscw.adm.bs_servuno
2018-08-28 09:28:12 Stopped bscw.adm.bs_servaccess
2018-08-28 09:28:12 Stopped bscw.adm.bs_servalarm
Loading EXTENSIONS from conf-20180828-0928
old msg -> conf/msg (copied)
old config_run.py -> conf/config_run.py (copied)
old ldap/config.py -> conf/ldap/config.py (copied)
New package airdesktop enabled
config.py updated
'/home/bscw/srv/bscw.domain.de/conf/__init__.py' updated
```
old apache2 -> conf/apache2 (copied)
old apache24 -> conf/apache24 (copied)
Import core modules ...
Link 'libexec' already exists - updating link...
Link destination '/home/bscw/lib/bscw-7.3.2-<rev>-py3?/extensions' does not exist
Found "Programs" (located):
    [...]
config_convert.py created
bsadmin update_defaults -v
bsadmin manage_servers -u
2018-08-28 09:28:14 bsadmin chkconfig -check-access
2018-08-28 09:28:14 access checks...
cc -o var/run/run_bscw var/run/run_bscw.c
2018-08-28 09:28:14 Actual license: OK
2018-08-28 09:28:14 bsadmin start
2018-08-28 09:28:15 Database version >= 2.1
2018-08-28 09:28:15 bsadmin bscw.adm.bs_convert30 -t
2018-08-28 09:28:15 Database version >= 3.0
2018-08-28 09:28:15 bsadmin bscw.adm.bs_convert31 -t
2018-08-28 09:28:15 Database version >= 3.1
2018-08-28 09:28:15 bsadmin bscw.adm.bs_convert33 -t
2018-08-28 09:28:15 Database version >= 3.3
2018-08-28 09:28:15 bsadmin bscw.adm.bs_convert40 -t
2018-08-28 09:28:15 Database version >= 4.0
2018-08-28 09:28:15 bsadmin bscw.adm.bs_convert45 -t
2018-08-28 09:28:15 Database version >= 4.5
2018-08-28 09:28:15 bsadmin bscw.adm.bs_convert50 -t
2018-08-28 09:28:15 Database version >= 5.0
2018-08-28 09:28:15 bsadmin bscw.adm.bs_convert51 -t
2018-08-28 09:28:15 Database version >= 7.0
2018-08-28 09:28:15 bsadmin bscw.adm.bs_convert70 -t
2018-08-28 09:28:15 Database version >= 7.1
2018-08-28 09:28:15 bsadmin bscw.adm.bs_convert71 -t
2018-08-28 09:28:15 Database version >= 7.3
2018-08-28 09:28:15 Converting to Version 7.3 ...  
2018-08-28 09:28:15 bsadmin garbage -map bscw.adm.bs_classtable30
2018-08-28 09:28:15 GC actual license: OK.
2018-08-28 09:28:16 GC start collection: size: 20792
2018-08-28 09:28:16 exit conversion commit:True converted:True
2018-08-28 09:28:16 Database version == 7.3
2.4. Upgrading to BSCW 7.3.2
BSCW Administrator Documentation, Release 7.3.2

2018-08-28 09:28:16 bsadmin bscw.adm.bs_fix_anonymous
2018-08-28 09:28:16 bsadmin http restart
restart pid=13582
2018-08-28 09:28:16 VERSION: BSCW 7.3.2
           Released: 20200214-1112-74856d2
bsadmin convert -check-access
Configure public prefix '/pub/' (Apache 24) ...
   (No authentication)
Configure secure prefix '/bscw/' (Apache 24) ...
   (HTTP_AUTHORISATION passed to BSCW)
   (Cookie authentication enabled)

Creating Apache HTTP server configuration files in
/home/bscw/srv/bscw.domain.de/conf/apache24
   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 '/home/bscw/srv/bscw.domain.de'

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

Installation succeeded. For next steps please check
/home/bscw/lib/bscw-7.3.2-<rev>-py3?/README.txt

Since Linux environments do not execute forked processes
set-group-id it is advisable to recursively change the owner the
preview cache and ./var/data files and directories 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 7.3.2:

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: [/home/bscw]
   Extracting BSCW |release| distribution in '/home/bscw/lib'

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

(continues on next page)
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 7.3.2 “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. /home/bscw/srv/<hostname>). First stop the BSCW server and then move it:

```
$ su -
$ cd /usr/local/bscw
$ ./server/start_servers -k

# BSCW_HOME=`su - bscw -c 'echo $HOME'` # e.g. BSCW_HOME=/home/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 5.2.3 [/home/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 7.3.2. You finally need to adjust the HTTP server configuration. See configuration section above.

### 2.4.2 Upgrading on Windows

Before upgrading a BSCW instance ensure to install the *Python for Windows Extensions* (pywin32) at least with Build 219. To upgrade an existing BSCW instance on Windows start the BSCW setup procedure by double-clicking

bscw-7.3.2-<rev>-py3?.exe

This will (re-)install the BSCW 7.3.2 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]]
[upgrade BSCW 5.2.3 [c:\bscw\srv\bscw.domain.org]]
install new BSCW instance

and click [Continue]. A console window is opened and the selected BSCW instance runtime directory is updated.

See also:
Section 4.2 Installation and Configuration for a detailed description

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.
CHAPTER THREE

INSTALLATION PROCEDURE FOR UNIX

These are the installation instructions for BSCW 7 on Unix machines. If you are upgrading an existing BSCW server instance please go through section 2.4 Upgrading to BSCW 7.3.2.

3.1 System requirements

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

- Intel Core/Xeon or AMD EPYC/Opteron (>3,5 GHz) 64-bit server system with at least 6 cores (or comparable systems of other manufacturers).
- 16 GB RAM
- at least 500 GB hard disk space (the BSCW installation requires approx. 200 MB disk space)

Additionally the following software is required:

- Apache HTTP Server 2.4
- Node.js (v10 LTS) an event driven JavaScript runtime
- a Python 3.6 or 3.7 interpreter
- Python Jinja2 template engine
- extensions for Python (optional)
  - pylucene - required for full text indexing support (package PyLucIndex)
  - ldap3 - required for LDAP/Active Directory bindings (package ldap)
- (optional) memcached to speed-up large folder handling
- (optional) converter software for the BSCW preview feature

Before installing BSCW, first install the Apache HTTP server, the Node.js runtime environment, the Python 3 interpreter, the Python Jinja2 template engine and the desired Python extension packages or converter software:

- OrbiTeam supports supports Debian based distributions (e.g Debian, Mint, Ubuntu) and Enterprise Linux/Fedors based distributions (e.g. Fedora, RHEL, CentOS).

- Generally it is recommended to choose a Unix distribution which has native support for the required software as the desired optional Python extensions or converter software. For example the LibreOffice suite should be available as installable package.

**Note:**

- We currently support versions 3.6 or 3.7 of the Python 3 interpreter only.

- On Fedora based systems you need to add the EPEL repository, see https://fedoraproject.org/wiki/EPEL for details.
• 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, the Node.js runtime environment, the Python 3 interpreter, the Python Jinja2 template engine, the desired Python 3 extension packages and the converter software are installed.

On Linux systems it is recommended to use a Debian or an Enterprise Linux/Fedora based distribution. Generally the Python packages of the distribution should be preferred.

The best way to install Nodes.js is using the package manager of your Linux distribution, see [https://github.com/nodesource/distributions/blob/master/README.md](https://github.com/nodesource/distributions/blob/master/README.md) for details:

- **Debian based systems:**
  ```bash
  $ su -
  # curl -sL https://deb.nodesource.com/setup_10.x | bash -
  # apt install -y nodejs
  ```

- **Fedora based systems:**
  ```bash
  $ su -
  # curl -sL https://rpm.nodesource.com/setup_10.x | bash -
  ```

See also [https://developers.redhat.com/rhel8/hw/nodejs/](https://developers.redhat.com/rhel8/hw/nodejs/) to install Node.js from the application stream repository:

```bash
$ su -
# yum module install nodejs:10
```

Remind after the installation of the BSCW software to enable the BSCW event service in the instance configuration file `<bscw-runtime-path>/conf/config.py`, see Node.js event server configuration.

Packages name(s) for these Linux distributions:

- **Debian based systems:** apache2 python3 python3-jinja2 python3-ldap3
- **Fedora based systems:** httpd python3 python3-jinja2 python3-ldap3

To increase the processing speed of large BSCW folders, you can optionally install the memory object caching system `memcached` in conjunction with the Python `memcache` client library. In this case, BSCW automatically detects the availability of the `memcached` daemon.

Packages name(s) for these Linux distributions:

- **Debian based systems:** python3-memcache memcached
- **Fedora based systems:** memcached and:
  ```bash
  $ su -
  # pip3 install python3-memcached
  ```

Additionally install the converter software required for BSCW preview, see [Software for BSCW Preview](#) for details.

The BSCW server software distribution is available as *tar* archive `bscw-7.3.2-<rev>-py3?.tar.gz`.

The name of the download file contains BSCW and Python version numbers – e.g. `bscw-7.3.2-<rev>-py3?.tar.gz` contains BSCW version 7.3.2 for Python 3.? 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 https://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 of section 3.4.3 Administrator account.

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).
- (Optional) if the binary python package setproctitle is installed BSCW processes are displayed with more telling names. For BSCW 7.3 the setprocfile package is included for Python 3.6, 3.7 (Linux). Alternatively you may install
  - Debian based systems: python3-setproctitle
  - Fedora based systems: python36-setproctitle
  or use:

  ```bash
  $ su -
  # pip3 install setproctitle
  ```

Ensure to disable the SELinux extension (which is enabled by default on Fedora based systems), e.g. usually set in /etc/selinux/config:

```bash
#SELINUX=enforcing
SELINUX=permissive
```

and reboot your system.

Generally the following file layout is proposed for BSCW instances

```bash
/home/bscw/                 # BSCW user home directory
/home/bscw/ .bscw/         # (as defined in /etc/passwd!)
/home/bscw/ .bscw/bscw.conf
/home/bscw/ .bscw/bscw_conf.py

/home/bscw/lib/            # BSCW distribution libraries
/home/bscw/lib/bscw-7.3.2-<rev>-py3?/ # BSCW distribution 7.3.2
/home/bscw/lib/bscw-7.3.2-<rev>-py3?-bin
/home/bscw/lib/bscw-7.3.2-<rev>-py3?-bscw
/home/bscw/lib/bscw-7.3.2-<rev>-py3?-doc
/home/bscw/lib/bscw-7.3.2-<rev>-py3?-etc
/home/bscw/lib/bscw-7.3.2-<rev>-py3?-lib
```

3.2. Installation
### The BSCW layout allows to install multiple BSCW instances in the runtime directory /home/bscw/srv, which all share the same BSCW program code located in the library directory /home/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 /home/bscw). If no BSCW user is found a new BSCW system user bscw with an own group bscw and a home directory /home/bscw is proposed and then created.

#### Note:

- /home/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 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 /home/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

```bash
$ su -
# id
uid=0(root) gid=0(root) groups=0(root)
# tar xf bscw-7.3.2-<rev>-py3?.tar.gz
# cd bscw-7.3.2-<rev>-py3?
# ./install.sh
```

It is highly advisable to use only the distribution install script ./install.sh as superuser. The script automatically determines required owner permission changes and performs configuration checks (systemd) which are not possible...
as BSCW system user “bscw”.

**Note:** If you do not want to run the `install.sh` script as superuser you may install BSCW completely manual as follows (necessary permission changes may not be performed then!):

- **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-7.3.2-<rev>-py3?.tar.gz
  $ cd $HOME/lib
  $ tar xf /tmp/bscw-7.3.2-<rev>-py3%/bscw-7.3.2-<rev>-py3?.tar
  ```

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

  ```
  $ cd $HOME/lib/bscw-7.3.2-<rev>-py3?
  $ python3 ./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: [/home/bscw]
Enter BSCW base directory: [/home/bscw]

Extracting BSCW 7.3.2 distribution in /home/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
/home/bscw/srv
the default [bscw.domain.org] is created):

BSCW instance name: [bscw.domain.org]
```
target '/home/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: admin@domain.org
BSCW login name: admin
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/sec/bscw.cgi (requires authentication) (may be left empty and defaults to 'sec')
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 = "admin@domain.org"
SERVER_ADMINS = [ "admin" ]

Are these settings correct (yes/no)? yes

conf/config.py updated
'/home/bscw/srv/bscw.domain.org/conf/__init__.py' updated
Import core modules ...
Link destination '/home/bscw/lib/bscw-7.3.2-<rev>-py3?/extensions' does not exist
Found "Programs" (located):
    [...] 
config_convert.py created
bsadmin update_defaults -v
bsadmin manage_servers -u
2018-08-28 09:28:46 bsadmin chkconfig -check-access

Chapter 3. Installation procedure for Unix
BSCW Administrator Documentation, Release 7.3.2

2018-08-28 09:28:46 access checks...
cc -o var/run/run_bscw var/run/run_bscw.c
2018-08-28 09:28:46 Actual license: OK (none)
2018-08-28 09:28:46 bsadmin start
2018-08-28 09:28:47 bsadmin garbage -license

is invalid for BSCW 7.3
Try installing Evaluation licence
Your server: org.domain.bscw:4435.sec
Evaluation licence expires: 20181127
Evaluation licence max users: 200

 [...] bsadmin convert -check-access
Configure public prefix '/pub/' (Apache 24)...
(No authentication)
Configure secure prefix '/sec/' (Apache 24) ...
(HTTP_AUTHORISATION passed to BSCW)
(Cookie authentication enabled)

Creating Apache HTTP server configuration files in
/home/bscw/srv/bscw.domain.org/conf/apache24
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 admin registered, address:
admin@domain.org: (is_owned_by_user)

BSCW server up and running in '/home/bscw/srv/bscw.domain.org'
BSCW instance created: '/home/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
/home/bscw/lib/bscw-7.3.2--<rev>--py3?/README.txt

Since Linux environments do not execute forked processes
set-group-id it is advisable to recursively change the owner the
preview cache and ./var/data files and directories 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 mes-
sages. The frequently asked questions (FAQ) list (https://www.bscw.de/en/support/) might also be helpful.

3.3 Software for BSCW Preview

The BSCW preview component displays thumbnail images for uploaded documents. If the user moves the mouse
pointer over an BSCW object icon in the type column, an image of the first page of an document is shown.
To enable the BSCW preview component the following additional software must be available on the hosting system:

1) Java Runtime Environment 8 or 11
   Java platform independent programming language
   • The Java runtime environment (JRE) of the distribution should be installed.
     Packages name(s) for common Linux distributions:
     – Debian based systems: openjdk-8-jdk resp. openjdk-11-jdk
     – Fedora based systems: java-1.8.0-openjdk resp. java-11-openjdk

2) PhantomJS 2.1 (http://phantomjs.org/)
   PhantomJS is a headless WebKit scriptable with a JavaScript API. It is available as distribution package:
   • Debian based systems: phantomjs
   Alternatively it can be can be downloaded from:
   
   [http://phantomjs.org/download.html](http://phantomjs.org/download.html)
   • For other Linux systems binaries are available at:
     [https://bitbucket.org/ariya/phantomjs/downloads/phantomjs-2.1.1-linux-x86_64.tar.bz2](https://bitbucket.org/ariya/phantomjs/downloads/phantomjs-2.1.1-linux-x86_64.tar.bz2)
   • Copy the binary bin/phantomjs in a location accessible by your PATH, e.g. in /usr/local/bin/phantomjs

3) Ghostscript 9 (https://ghostscript.com)
   Ghostscript is an interpreter for the PostScript language and for PDF
   • The Ghostscript interpreter version of the distribution should be installed. Additionally the standard Ghostscript fonts are required.
     Packages name(s) for common Linux distributions:
     – Debian based systems: ghostscript gsfonts
     – Fedora based systems: ghostscript ghostscript-fonts

4) GraphicsMagick (http://www.graphicsmagick.org)
   GraphicsMagick is a library for image processing
   • The GraphicsMagick version of the distribution should be installed.
     Packages name(s) for common Linux distributions:
     – Debian based systems: graphicsmagick
     – Fedora based systems: GraphicsMagick
     After installation check if GraphicsMagick correctly finds Ghostscript:
     ```
     $ gm convert -list Delegates
     ...
     ps<->pdf "gs" -q -dBATCH -dSAFER -dMaxBitmap=50000000 -dNOPAUSE
     -sDEVICE=pdfwrite "-sOutputFile=%o" -- "%i" -c quit
     ```

5) LibreOffice (https://www.libreoffice.org/)
   LibreOffice is a open source office suite
Note: At least LibreOffice version 5 is required, best use the current release LibreOffice 6.2 or 6.3

- The LibreOffice version of the distribution should be installed.

  Packages name(s) for common Linux distributions:
  - Debian based systems: libreoffice python3-uno
  - Fedora based systems: libreoffice libreoffice-pyuno

- For better conversion results install the Microsoft TrueType core fonts
  - Debian based systems: ttf-mscorefonts-installer
  - Fedora based systems: see http://mscorefonts2.sourceforge.net/

- Ensure the home directory of the Apache HTTP server user is writable for the Apache HTTP server user, because LibreOffice creates temporary files in the users’ home directory.
  - Debian based systems:
    ```bash
    $ su -
    # chown www-data:www-data /var/www
    ```
  - Fedora based systems:
    ```bash
    $ su -
    # chown apache:apache /usr/share/httpd # EL
    ```

Attention: Be sure the Python UNO bridge is installed!

6) Text/HTML converter

Install the markdown2 and html2text converters as follows:

- markdown2 converts text to HTML using the markdown markup.

  Packages name(s) for common Linux distributions:
  - Debian based systems: python3-markdown2
  - Fedora based systems: python3-markdown2

  If your distribution does not support a native version, use pip3 to download and install markdown2:

  ```bash
  $ su -
  # pip3 install markdown2
  ```

- html2text converts HTML to text using the markdown markup

  Packages name(s) for common Linux distributions:
  - Debian based systems: python3-html2text
  - Fedora based systems: python3-html2text

  Note: On Debian python3-html2text is installed as html2markdown.

  If your distribution does not support a native version, use pip3 to download and install html2text:

  ```bash
  $ su -
  # pip3 install html2text
  ```
7) Image converter

For image conversion the Python Imaging Library is required

Packages name(s) for common Linux distributions:

- Debian based systems: python3-pil
- Fedora based systems: python3-pillow

8) Apache Tika

BSCW utilizes the Apache Tika toolkit (https://tika.apache.org) to extract metadata and text from uploaded documents. To enable the Apache Tika a Java Runtime Environment 8 or 11 must be available on the server host.

To accelerate metadata extraction it is possible to install an optional standalone tika-server. For installation download the tika-server JAR archive from

https://www.apache.org/dyn/closer.cgi/tika/tika-server-1.??.jar

and copy it into the BSCW distribution

$ cd $HOME/lib/bscw-7.3.2-<rev>-py3?
$ cp tika-server-1.??.jar bscw/libexec/tika
$ chmod 644 bscw/libexec/tika/tika-server-1.??.jar

Additionally the tika Python package is required, use pip3 to download and install tika

$ su -
# pip3 install tika

If the prerequisites 1-7 are met run

- bsadmin update_defaults to generate a new BSCW converter configuration (<bscw-runtime-path>/conf/config_convert.py). Use the verbose option (-v) to check if BSCW found the required converter programs to create the previews files:

$ cd <bscw-runtime-path>
$ ./bin/bsadmin update_defaults -v
...
Converter auto-configuration:
Found Commands:
'gm': '/usr/bin/gm'
'java': '/usr/bin/java'
'phantomjs': '/usr/local/bin/phantomjs'
'unoconv': '%(py)s %(cnv)s/unoconv/unoconv --pipe=%(pid)s'
...
config_convert.py updated

Optionally you may create for all existing documents the required preview files using the bsadmin preview command:

$ ./bin/bsadmin preview
Usage:
bsadmin preview list
bsadmin preview create [-v|-q] [-f|-ff] [<oid0> ... <oidn>]
bsadmin preview delete [-v|-q] [<oid0> ... <oidn>]
bsadmin preview [-h]

Generate Document preview documents

positional arguments:
list print preview states and preview document file names
create created preview for documents in 'var/cache/preview'
delete deletes preview states and generated preview documents

(continues on next page)
optional arguments:
- \( \texttt{-f} \) force upgrade of all previews
- \( \texttt{-ff} \) force upgrade of previews with state 'FAILURE'
- \( \texttt{-v} \) verbose
- \( \texttt{-q} \) quiet
- \( \texttt{-h} \) show this help message and exit

Note:
- On large BSCW installations \texttt{bsadmin preview create} may take a very long period (weeks!)
- The execution of \texttt{bsadmin preview create} is not mandatory, because preview files are automatically scheduled for background creation the first time an existing folder is read by an user.

In the case of problems with automatic preview file generation enable logging by adding the following entry to \texttt{BSCW_LOGGING} in <bscw-runtime-path>/conf/config.py. The BSCW preview component will then log into <bscw-runtime-path>/var/log/prev.log:

\begin{verbatim}
BSCW_LOGGING = {
    'sys': ('WARN', 'sys.log'),
    'prev': ('DEBUG', 'prev.log'),
    # ...
}
\end{verbatim}

An preview log file entry:

\begin{verbatim}
2018-02-10 11:35:07 prev DEBUG pid 123 error: libexec/conv: Document #456 ...
gm convert: Unable to get type metrics...
\end{verbatim}

indicates that the ghostscript standard fonts are missing resp. are not properly installed.

Note: To disable the BSCW preview feature add an entry \texttt{CREATE_PREVIEWS} in <bscw-runtime-path>/conf/config.py:

\begin{verbatim}
CREATE_PREVIEWS = \texttt{False}
\end{verbatim}

\section{3.4 Configuration}

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

\subsection{3.4.1 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 Services (UNO) of section 7.4.1 or the alarm service) to access the BSCW database server via HTTP using the following URL:

\begin{verbatim}
http://localhost/pub/bscw.cgi/
\end{verbatim}

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

```
<bscw-runtime-path>/conf/apache24/mod.conf
<bscw-runtime-path>/conf/apache24/site.conf
<bscw-runtime-path>/conf/apache24/bscw.conf
```

which contain all necessary configuration instructions.

The `mod.conf` file ensures the following additional modules required by BSCW are loaded and may be included in the main Apache HTTP server configuration file:

- `cgid_module` (or `cgi_module`)
- `deflate_module`
- `expires_module`
- `headers_module`
- `rewrite_module`
- `ssl_module`
- `proxy_module` [1, 2]
- `proxy_http_module` [1, 2]
- `proxy_wstunnel_module` [2]

**Note:**

- The `suexec_module` must be disabled.

Anyway the preferred mechanism of your Unix distribution should be used to enable the required modules:

- **Debian based systems:**
  ```
  $ su -
  # a2enmod cgid deflate expires headers rewrite ssl
  # a2enmod proxy proxy_http # [1, 2]
  # a2enmod proxy_wstunnel # [2]
  # a2dismod suexec
  # systemctl restart apache2
  ```

- **Fedora based systems:**
  ```
  $ su -
  # vim /etc/httpd/conf.modules.d/00-base.conf # RHEL 7
  # vim /etc/httpd/conf.modules.d/00-proxy.conf # [1, 2]
  # vim /etc/httpd/conf.modules.d/00-ssl.conf
  # systemctl restart httpd
  ```

[1] Required for the BSCW pre-forked HTTP server (see `http` for details).

[2] Required for the Online Office or the Node.js event server (see `office` or `Node.js` installation for details).

The `site.conf` file contains several virtual host containers which can be used for Apache layouts which support site configuration file directories (e.g. Debian based systems `/etc/apache2/sites-available/`, Fedora based systems `/etc/httpd/conf.d/`).

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 adapt it to your local Apache HTTP server configuration. Especially you will need to install certificates for your SSL enabled server and adapt 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 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/apache24/bscw.conf
```

When using virtual web server container (`<VirtualHost> ... </VirtualHost>`) directives, it is possible to include the `<bscw-runtime-path>/conf/apache24/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
        AddType                  text/html en de es fr
        ForceLanguagePriority    Fallback
        Require                  all granted
    </Directory>
    Include "<bscw-runtime-path>/conf/apache24/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
    </Directory>
</VirtualHost>
```

(continues on next page)
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
    AddType text/html en de es fr
    ForceLanguagePriority Fallback
    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/apache24/bscw.conf"
</VirtualHost>
```

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

- If no option is used `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.

Chapter 3. Installation procedure for Unix
• 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.

• 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). These options require the deflate (-D) or the expires (-E) modules (these options are enabled by default).

• If option `-d` (instead of `-D`) also enables compression for BSCW responses.

**Warning:** Compression and TLS encrypted connections may allow an information disclosure attack (for more information search for “breach” attacks).

**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.

• 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.

• 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`). After altering these directives `bsadmin conf_apache` must be run again.

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.
3.4.2 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.

For BSCW 7 it is mandatory to enable the pre-forking BSCW HTTP server to speed up request processing. Refer to section http resp. Apache HTTP Server Configuration for a description how to enable the BSCW HTTP Server.

Local configuration details of your BSCW instance are held in the configuration file at `<bscw-runtime-path>/conf/config.py` (cf. section 5.2 `conf/config.py`). 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 `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 are using MS Exchange as MTA, you must explicitly allow the IP address of your BSCW server host to *relay* email.

BSCW refreshes its user interface constantly to reflect changes by other users, even if the current user does not interact with the system. To push changes almost real-time to active users, a WebSocket-based Node.js events server is required (see Node.js).

After the Node.js events server is installed, configure two endpoints for the events server in the BSCW runtime configuration `<bscw-runtime-path>/conf/config.py`:

```python
EVENTS_SERVER_WS = 'ws://127.0.0.1:3836'
EVENTS_SERVER_HTTP = 'http://127.0.0.1:3837'
```

`EVENTS_SERVER_WS` defines the WebSocket endpoint meant to accept incoming user connections. It is typically located at localhost nonetheless because the Apache HTTP server is running as a reverse proxy in front of it. `EVENTS_SERVER_HTTP` is the internal RPC endpoint used by BSCW only, to exchange event and user login data with the events server.

None of these endpoints use SSL. However, for the external user connections, the Apache HTTP will provide encryption, if your BSCW server is set up for HTTPS.

After editing `<bscw-runtime-path>/conf/config.py` enable the Apache modules `proxy`, `proxy_http` and `proxy_wstunnel`. Next run `bsadmin conf_apache` to set up the reverse proxy configuration and restart the Apache server.

Afterwards, restart BSCW. There should be an events_server.log log file in the BSCW runtime’s log folder. You might notice some log messages about missing dependencies on first startup; these dependencies are automatically installed (an active internet connection is required) and the events server restarts itself afterwards.

Note: If there is an Application Level Firewall installed at your site, be sure that it supports and allows WebSocket connections.

### 3.4.3 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/sec/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 file system without the `nosuid` option.
3. Your operating system does not support the *set-group-id* bit for scripts (e.g., 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:

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

### 3.4.4 De-Installation

To de-install BSCW perform the following manual steps:

- Disable your BSCW startup procedure (see *BSCW Startup* for details).
- Disable all BSCW related entries in the crontab (see *Garbage Collection*) and disable the backup procedure (*Backup*).
- Stop your BSCW instance

```bash
$ cd $HOME/srv/<bscw-runtime-dir>
$ ./bin/bsadmin stop
```

- Next remove all instance data, e.g.

```bash
$ cd $HOME/srv
rm -rf <bscw-runtime-dir>
```

**Note:** This step irrevocably destroys all user data!

- Finally remove the related BSCW distribution library, e.g.

```bash
$ cd $HOME/lib
rm -rf bscw-7.3.2-<rev>-py3?
```

**Note:** You may only remove the BSCW distribution library if no existing other BSCW instance requires this particular BSCW revision!

### 3.5 Database Server Startup, 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

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

- to stop BSCW database server, use

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

- to run the garbage collector, use

  ```bash
  $ <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.
3.5.1 BSCW Startup

First install the static configuration scripts in the according directory for your system. E.g. for Debian copy the files

```bash
$ sudo su -
# id
uid=0(root) gid=0(root) groups=0(root)
# cd /home/bscw/lib/bscw-7.3.2-<rev>-py3?/etc/posix/debian
# cp ./etc/default/bscw /etc/default
# cp ./etc/cron.daily/bscw /etc/cron.daily
# cp ./etc/cron.hourly/bscw_cleantmp /etc/cron.hourly
# cp ./etc/logrotate.d/bscw /etc/logrotate.d
# chmod 755 /etc/cron.daily/bscw /etc/cron.hourly/bscw_cleantmp
```
to the `/etc` directory. Afterwards edit the `/etc/default/bscw` (on Debian) resp. `/etc/sysconfig/bscw` (on Fedora) and `/etc/logrotate.d/bscw` files to adopt the BSCW user and the paths to your BSCW instances runtime directories.

To create a systemd service configuration run `bsadmin conf_systemd` and follow the given instructions:

```bash
$ bin/bsadmin conf_systemd
A systemd multiple instance service file ::

   bscw@.service

has been created. Please check the contents and perform the following commands as root user ::

$ sudo su -
# id
uid=0(root) gid=0(root) groups=0(root)
# cd /home/bscw/srv/<bscw-instance-name>
# cp ./conf/systemd/system/bscw@.service \
   /etc/systemd/system
# systemctl daemon-reload
# systemctl enable bscw@<bscw-instance-name>.service
```

3.5.2 Garbage Collection

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:

```bash
# 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!

3.5.3 Backup

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>.

**Note:**

- The var/data/Text and var/data/Index directories may be skipped while backup, because the contents may be reconstructed after restoration of a backup.
- You can use any incremental backup method to backup your BSCW instance

### 3.6 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 Postfix or sendmail).

#### 3.6.1 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:

```python
# MDA_MTA
# Specifies the local mail transfer agent (MTA), currently supported are:
# 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
```

(continues on next page)
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).

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 = ''
```

### 3.6.2 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"
```

**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)

```python
MDA_MTA = 'postfix'
```

**Note:** When using Python < 3.7 the delivery of mails with unicode characters in the subject fails. To workaround this set `LANG=en_US.UTF-8` within the postfix `import_environment` parameter in the Postfix configuration file `/etc/postfix/main.cf`, e.g.:

```bash
$ su -
# postconf -d import_environment
import_environment = MAIL_CONFIG MAIL_DEBUG MAIL_LOGTAG TZ XAUTHORITY DISPLAY
  ➔ LANG=C
# add to /etc/postfix/main.cf:
import_environment = MAIL_CONFIG MAIL_DEBUG MAIL_LOGTAG TZ XAUTHORITY DISPLAY
  ➔ LANG=en_US.UTF-8
```

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`:

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

Send a mail message to the prepared folder address and check in file: `/var/log/syslog` (or wherever your postfix MTA writes its log entries) if the local postfix MTA 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=<2018111542916.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/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:

```
2018-11-15 15:29:18 mda INFO invoked as 523/57
2018-11-15 15:29:18 mda DEBUG
    MDA_MTA = 'postfix'
    MDA_MBOX = 'lab'
    MDA_DOMAIN = 'bscw.de'
2018-11-15 15:29:18 mda INFO start delivery
2018-11-15 15:29:18 mda INFO sender addr in 'from': header.
```

(continues on next page)
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:

```plaintext
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:

```plaintext
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 `~/.forward` file:

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

or

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

```plaintext
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 **BCSW_LOGGING** directive in the BSCW server instance configuration file `<bscw-runtime-path>/conf/config.py`:

```
BCSW_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=<201811151600.97G0DWO8799@tormenta.orbiteam.de>,
  proto=ESMTP, daemon=MTA-IPv4, relay-mail [195.127.160.172]
Nov 15 15:29:17 maestral sendmail[5802]: g97G0Kp05801: 
  to=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:

```
2018-11-15 15:29:18 mda  INFO  invoked as 523/57
2018-11-15 15:29:18 mda  DEBUG MDA_MTA = 'sendmail'
2018-11-15 15:29:18 mda  INFO  start delivery
2018-11-15 15:29:18 mda  INFO  recipient in 'from': header.
2018-11-15 15:29:18 mda  INFO  set domain to 'bscw.de'
2018-11-15 15:29:18 mda  INFO  store document
2018-11-15 15:29:18 mda  INFO  message loaded
2018-11-15 15:29:18 mda  INFO  message stored size=2028
2018-11-15 15:29:18 mda  INFO  virus check OK
2018-11-15 15:29:18 mda  INFO  mag for Folder#118433 (access 'anybody');
2018-11-15 15:29:18 mda  INFO  mag from info <info@orbiteam.de> delivered.
```
CHAPTER FOUR

INSTALLATION PROCEDURE FOR WINDOWS

These are the installation instructions for BSCW 7.3 on Windows 10, Server 2016/2019 machines. If you are upgrading an existing BSCW server instance please go through section 2.4 Upgrading to BSCW 7.3.2

4.1 System requirements

For approximately 200 users BSCW requires the following server hardware on a Windows 10, Server 2016/2019 installation:

- Intel Core/Xeon or AMD EPYC/Opteron (>3.5 GHz) 64-bit server system with at least 6 cores (or comparable systems of other manufacturers).
- 16 GB RAM
- At least 500 GB hard disk space (the BSCW installation requires about 200 MB disk space)
- Windows 10, Server 2016/2019 with
  - Apache HTTP Server 2.4 or
  - Microsoft Internet Information Server (IIS 8/9/10)

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

- a Python 3.6 or 3.7 interpreter
- pywin32 Build 227 (Win32 Extensions and API for Python)
- Node.js (v10 LTS) an event driven JavaScript runtime
- Python Jinja2 template engine
- extensions for Python (optional)
  - pylucene - required for full text indexing support (package PyLucIndex)
  - ldap3 - required for LDAP/Active Directory bindings (package ldap)

- (optional) converter software for the BSCW preview feature, see Software for BSCW Preview for details

The “Python” interpreter, the “Win32 Extensions and API for Python” (pywin32) and the Python Jinja2 templating engine are copyrighted, but freely usable and can be downloaded from:

https://www.python.org/
https://github.com/mhammond/pywin32/releases/
http://jinja.pocoo.org/

The “Node.js” LTS version can be downloaded from:

https://nodejs.org/en/download/

Additionally you require a Web server. BSCW supports

- Apache HTTP Server 2.4
• Microsoft Internet Information Server (IIS 8/9/10)

To use the BSCW WebDAV functionality, you must use the Apache HTTP server. The Apache HTTP server is copyrighted, but is freely usable and can be downloaded from the Apache HTTP server project (http://httpd.apache.org). Windows binary distributions are available at http://www.apachehaus.com/ or http://www.apachelounge.com/

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

Note:

• BSCW requires at least Python for Windows Extensions Build 219. Please upgrade older pywin32 versions before running the BSCW installer bscw-7.3.2-<rev>-py3?.exe
• When installing pywin32 as a wheel package (using pip3 install pywin32) additionally the following command must be run from an elevated command prompt:

```bash
python C:\Program Files (x86)\Python3?-32\Scripts\pywin32_postinstall.py --install
```

• Before installing BSCW first install the desired Python extension packages (see above).
• If the installer fails with an error message like

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

please disable your virus scanner before running the BSCW installer bscw-7.3.2-<rev>-py3?.exe
• BSCW requires the use of a NTFS (local directory).
• After installing the Apache HTTP server it might be necessary to add an incoming firewall rule to your Windows Firewall for port 80 or 443.

### 4.2 Installation and Configuration

Before installing BSCW ensure the Web server, the Node.js runtime, the Python 3 interpreter, Python for Windows Extensions (pywin32), the Python “Jinja2” template engine and the desired Python 3 extension packages or converter software is installed.

Note:

• The Python 3 interpreter must be installed for all users in a system “program files” directory. Additionally the install directory must be appended to the system PATH. To achieve this enable the following installer options:

```plaintext
[x] Install launcher for all users (recommended)
[x] Add Python 3.? to PATH
```

• The easiest way to install “Nodes.js” is using the 64-bit Windows Installer (.msi) of the LTS version from:

```plaintext
https://nodejs.org/en/download/
```

Remind after the installation of the BSCW software to enable the BSCW event service in the instance configuration file <bscw-runtime-path>\conf\config.py, see Node.js event server configuration.

• When installing pywin32 as a wheel package (using pip3) additionally the following command must be run from an elevated command prompt:
• The Python “Jinja2” template engine can be installed using the Python packet manager pip3. Open a Windows command prompt with administrator privileges and run:

```
> pip3 install jinja2
```

• 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.

The name of the download installer contains BSCW and Python version numbers – e.g. `bscw-7.3.2-<rev>-py3?.exe` contains BSCW version 7.3.2 for Python 3.?. Please make sure to install the latest version of BSCW and always provide your version number when contacting support staff.

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

`bscw-7.3.2-<rev>-py3?.exe`

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

Then the setup program will try to install the BSCW version 7.3.2. Click [Next] and accept the license agreement.
and continue with [Next]. Read the HTTP server hints

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:
Click [Next] and choose as additional task to install or update a BSCW server instance:

Forward with [Next] to see a summary of the chosen locations:
To accept this click [Install] which will extract the BSCW 7.3.2 distribution files in the following locations

C:\BSCW\lib\bscw-7.3.2-<rev>-py3?\  # BSCW distribution 7.3.2
C:\BSCW\lib\bscw-7.3.2-<rev>-py3?\bin
C:\BSCW\lib\bscw-7.3.2-<rev>-py3?\bscw  # BSCW executable code
C:\BSCW\lib\bscw-7.3.2-<rev>-py3?\doc  # BSCW documentation
C:\BSCW\lib\bscw-7.3.2-<rev>-py3?\etc
C:\BSCW\lib\bscw-7.3.2-<rev>-py3?\lib  # BSCW third party modules

The BSCW 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:**

- The 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:\).
- Due to Windows access right restrictions is not possible to install a BSCW runtime directory in C:\Program Files, C:\Program Files (x86) or C:\Windows.

After installing the BSCW distribution files the setup program will run the BSCW instance setup to examine your system, and if a BSCW instance runtime is found the following selection is shown:

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 4.5.1 *BSCW Server Root Definition*

**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 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.

*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/](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/sec/](https://bscw.domain.org/sec/). It is recommended to install
BSCW as a service.

See also:

*Windows Service* for Further Information.

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

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.
BSCW refreshes its user interface constantly to reflect changes by other users, even if the current user does not interact with the system. To push changes almost real-time to active users, a WebSocket-based Node.js events server is required.

After the Node.js events server is installed, configure two endpoints for the events server in the BSCW runtime configuration `<bscw-runtime-path>\conf\config.py`:

```
EVENTS_SERVER_WS = 'ws://127.0.0.1:3836'
EVENTS_SERVER_HTTP = 'http://127.0.0.1:3837'
```

`EVENTS_SERVER_WS` defines the WebSocket endpoint meant to accept incoming user connections. It is typically located at localhost nonetheless because the Apache HTTP server is running as a reverse proxy in front of it. `EVENTS_SERVER_HTTP` is the internal RPC endpoint used by BSCW only, to exchange event and user login data with the events server.

None of these endpoints use SSL. However, for the external user connections, the Apache HTTP will provide encryption, if your BSCW server is set up for HTTPS.

When changing `EVENTS_SERVER_WS` or `EVENTS_SERVER_HTTP` settings the BSCW server instance must be restarted.

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 the setup process automatically generated configuration files in the directory `<bscw-runtime-path>\conf\apache24`, which contains all necessary configuration instructions.

  While using “virtual hosts” is recommended, the easiest option is to include the directive `Include <bscw-runtime-path>\conf\apache24\bscw.conf` to the file `httpd.conf`.

  After editing `<bscw-runtime-path>\conf\config.py` enable the Apache modules `proxy`,
proxy_http and proxy_wstunnel. Next run `bsadmin conf_apache` to set up the reverse proxy configuration and restart the Apache server.

**Attention:** Before including the `bscw.conf` file ensure all required Apache HTTP modules are loaded, see Section 4.5.2 *Apache HTTP Server Configuration* for a more complete Apache HTTP server configuration discussion.

- **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 10, Server 2016/2019. Finally the setup script launches your default Web browser to connect to your BSCW server.

  Be sure to enable WebSocket support. For further details see Section 4.5.3 *IIS Configuration*.

**Note:**

- It is recommended to use the Apache HTTP 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]/sec/bscw.cgi
```

(e.g. `http://bscw.domain.org/sec/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
you get a BSCW overview which contains links to your BSCW instance.

4.3 Software for BSCW Preview

The BSCW preview component displays thumbnail images for uploaded documents. If the user moves the mouse pointer over an BSCW object icon in the type column, an image of the first page of an document is shown.

To enable the BSCW preview component the following additional software must be available on the hosting system

1) Java Runtime Environment 8 (http://www.oracle.com/technetwork/java)
   
   Java platform independent programming language
   
   • The Java Runtime Environment (JRE) must be separately installed. Download the JRE from:


   • Ensure the bin and bin\client directories of the JRE are listed in the Windows system “Path”
   environment variable, e.g. add:

   C:\Program Files (x86)\Java\jre1.8.0_???\bin;
   C:\Program Files (x86)\Java\jre1.8.0_???\bin\client

   Attention: After upgrading your Java Runtime Environment (JRE) to a new release the new
   installation path must be adapted manually in the Windows system “Path” environment variable.
   Afterwards a system restart is required.

2) PhantomJS 2.1 (http://phantomjs.org/)
   
   PhantomJS is a headless WebKit scriptable with a JavaScript API and can be downloaded from:

   http://phantomjs.org/download.html

   • A binary is available at:

   https://bitbucket.org/ariya/phantomjs/downloads/phantomjs-2.1.1-windows.zip

   • Copy the static binary bin\phantomjs.exe to a suitable location, e.g. to C:\Program
   Files (x86)\PhantomJS\bin\phantomjs.exe and add C:\Program Files
   (x86)\PhantomJS\bin to the Windows system Path environment variable.

3) Ghostscript 9 (http://ghostscript.com)
   
   Ghostscript is an interpreter for the PostScript language and for PDF
   
   • The Ghostscript interpreter installer can be downloaded from:

   https://github.com/ArtifexSoftware/ghostpdl-downloads/releases/download/
   ~gs927/gs927w64.exe

   and the standard Ghostscript fonts (ghostscript-fonts-std-8.11.tar.gz) from:

   https://sourceforge.net/projects/gs-fonts/
• Execute the installer gs927w64.exe which installs Ghostscript to the default location C:\Program Files\gs\gs9.27. After successful installation add the C:\Program Files\gs\gs9.27\bin path to the Windows system Path environment variable.

• Next extract the Ghostscript fonts directly into the C:\Program Files\gs\gs9.27\lib directory.

Attention:
– include the font files directly in the lib directory and not inside a fonts sub directory!
– we experienced the most recent version of Ghostscript 9.50 fails with GraphicsMagick.

4) GraphicsMagick (http://www.graphicsmagick.org)

GraphicsMagick is a library for image processing

• The GraphicsMagick installer can be downloaded from:


• Execute the installer GraphicsMagick-1.3.33-Q16-win64-dll.exe which installs GraphicsMagick to the default location C:\Program Files\GraphicsMagick-1.3.33-Q16. After successful installation add the C:\Program Files\GraphicsMagick-1.3.33-Q16 path to the Windows system Path environment variable (This could be automatically achieved by selecting “Update executable search path” from within the installer).

• After installation check if GraphicsMagick correctly finds Ghostscript:

```bash
> gm convert -list Delegates
... ps<>pdf "C:\Program Files\gs\gs9.27\bin\gswin64c.exe" -q -dBATCH -dSAFER -dMaxBitmap=300000000 -dNOPAUSE -sDEVICE=pdfwrite -sOutputFile=%o -- "$i" -c quit
```

Note:
• GraphicsMagick requires the installation of the Microsoft Visual C++ 2008 Redistributable Package (x64).
• GraphicsMagick and Ghostscript must be installed for the same architecture. Mixing 32-bit and 64-bit installations of GraphicsMagick and Ghostscript might cause problems.

5) LibreOffice (https://www.libreoffice.org/)

LibreOffice is a open source office suite

Note: At least LibreOffice version 5 is required, best use the current release LibreOffice 6.2 or 6.3

The “LibreOffice” installer can be downloaded from:

https://www.libreoffice.org/download/download/

6) Text/HTML converter

Install markdown2 and html2text as follows

• markdown2 converts text to HTML using the markdown markup

Use pip3 to download and install markdown2:
> pip3 install markdown2

- **html2text** converts HTML to text using the markdown markup

Use **pip3** to download and install html2text:

> pip3 install html2text

7) **Image converter**

For image conversion the Python Imaging Library is required

Use the Python package manager **pip3** to download and install:

> pip3 install Pillow

8) **Apache Tika**

BSCW utilizes the Apache Tika toolkit (https://tika.apache.org) to extract metadata and text from uploaded documents. To enable the Apache Tika a Java Runtime Environment 8 must be available on the server host.

To accelerate metadata extraction it is possible to install an optional standalone tika-server. For installation download the tika-server JAR archive from

https://www.apache.org/dyn/closer.cgi/tika/tika-server-1.22.jar

and copy it into the BSCW distribution

> cd C:BSCWlibbscw-7.3.2-rev-py3?
> copy tika-server-1.??.jar bscwlibexectika

Additionally the **tika** Python package is required, use **pip3** to download and install **tika**

> pip3 install tika

If the prerequisites 1-7 are met run

- **bsadmin update_defaults** to generate a new BSCW converter configuration (`<bscw-runtime-path>`/`conf`/`config_convert.py`). Use the verbose option (-v) to check if BSCW found the required converter programs to create the previews files:

  ```
  > cd C:\BSCW\srv\bscw
  > bin\bsadmin update_defaults -v
  ...
  Converter auto-configuration:
  Found Commands:
  'gm': 'c:\\program files\\graphicsmagick-1.3.?\-q16\\gm.exe'
  'java': 'c:\\\program files (x86)\\java\\jre1.8.0_??\\bin\\java.exe'
  'phantomjs': 'c:\\\program files (x86)\\phantomjs\\bin\\phantomjs.exe'
  'unoconv': '"%(py)s" "%(cnv)s\\unoconv\unoconv"'
  ...
  config_convert.py updated
  ```

Optionally you may create for all existing documents the required preview files using the **bsadmin preview** command:

  ```
  > bin\bsadmin preview
  Usage:
  bsadmin preview list
  bsadmin preview create [-v|-q] [-f|-ff] [oid0] ... <oidn>
  bsadmin preview delete [-v|-q] [oid0] ... <oidn>
  bsadmin preview [-h]
  ```

(continues on next page)
positional arguments:
- list print preview states and preview document file names
- create created preview for documents in 'var/cache/preview'
- delete deletes preview states and generated preview documents

optional arguments:
- -f force upgrade of all previews
- -ff force upgrade of previews with state 'FAILURE'
- -v verbose
- -q quiet
- -h show this help message and exit

Note:
- On large BSCW installations bsadmin preview create may take a very long period (weeks!)
- The execution of bsadmin preview create is not mandatory, because preview files are automatically scheduled for background creation the first time an existing folder is read by an user.

In the case of problems with automatic preview file generation enable logging by adding the following entry to BSCW_LOGGING in <bscw-runtime-path>/conf/config.py. The BSCW preview component will then log into <bscw-runtime-path>/var/log/prev.log:

```python
BSCW_LOGGING = {
    'sys': ('WARN', 'sys.log'),
    'prev': ('DEBUG', 'prev.log'),
    # ...
}
```

An preview log file entry:

```
2018-02-10 11:35:07 prev DEBUG pid 123 error: libexec/conv: Document #456
...gm convert: Unable to get type metrics...
```

indicates that the ghostscript standard fonts are missing resp. are not properly installed.

Note: To disable the BSCW preview feature add an entry CREATE_PREVIEWS in <bscw-runtime-path>/conf/config.py:

```python
CREATE_PREVIEWS = False
```

### 4.4 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. 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).
Note: Controlling Windows services requires administrative privileges.

> 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.

### 4.4.1 Windows Service

On Windows 10, Server 2016/2019 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.

### 4.4.2 Task Scheduler

On Windows 10, Server 2016/2019 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:
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`.
- Do **not** run the same BSCW server instance *more than once!* This may seriously damage the BSCW database.
4.5 Further Configuration Details

All BSCW configuration parameters are stored (similar to the Unix version) in configuration files.

See also:

Chapter 5 Configuration of BSCW Servers

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 10, Server 2016/2019 specific configuration options are explained:

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

4.5.1 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 = 'http://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.

4.5.2 Apache HTTP Server Configuration

Download e.g from http://www.apachehaus.com/ or http://www.apachelounge.com/ and install an Apache HTTP server binary on your Windows host by following the instructions for your Apache HTTP server binary distribution.
Note: After installing the Apache HTTP server it might be necessary to add an incoming firewall rule to your Windows Firewall for port 80 or 443.

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 Services (UNO) of section 7.4.1 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.

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 configuration files

<bscw-runtime-path>/conf/apache24/mod.conf
<bscw-runtime-path>/conf/apache24/site.conf
<bscw-runtime-path>/conf/apache24/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
ssl_module
proxy_module
proxy_http_module
proxy_wstunnel_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.

4.5. Further Configuration Details
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 adapt it to your local Apache HTTP server configuration. Especially you will need to install certificates for your SSL enabled server and adapt the configuration in `site.conf`.

Note: When using the BSCW `site.conf` file you should **not** include the `httpd-vhosts.conf` file which is included in the main configuration `httpd.conf` by default:

```
# Virtual hosts
#Include conf/extra/httpd-vhosts.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/apache24/bscw.conf
```

When using virtual web server container `<VirtualHost>` ... `</VirtualHost>` directives, it is possible to include the `<bscw-runtime-path>/conf/apache24/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 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
    AddType        text/html en de es fr
    ForceLanguagePriority Fallback
    Require        all granted
  </Directory>
  Include "<bscw-runtime-path>/conf/apache24/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
  </Directory>
</VirtualHost>
```

(continues on next page)
AddType text/html en de es fr
ForceLanguagePriority Fallback
Require all granted
</Directory>

Include "<bscw-runtime-path>/conf/apache24/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
    AddType text/html en de es fr
    ForceLanguagePriority Fallback
    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/apache24/bscw.conf"
</VirtualHost>

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

- If no option is used `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.
- 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`). These options require the `deflate` (`-D`) or the `expires` (`-E`) modules (these options are enabled by default).
- Using the `-d` (instead of `-D`) also enables compression for BSCW responses.

**Warning:** Compression and TLS encrypted connections may allow an information disclosure attack (for more information search for “breach” attacks).

**Note:**

- Certain configurations (such as cookie authentication) imply option `-r`.
- 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.

### 4.5.3 IIS Configuration

WebSocket support must be manually enabled for BSCW 7 as follows:

- In `[Control Panel → Programs → Programs and Features → Turn Windows features on or off]`, activate `[Internet Information Services → World Wide Web Services → Application Development Features → Web-Socket Protocol]`.
- Next, install ARR from https://www.iis.net/downloads/microsoft/application-request-routing and restart IIS.
- Finally, in `[Computer Management → Services and Applications → Internet Information Services (IIS Manager)]`, browse to your BSCW servers’ site using the left-hand tree, select your configured pub URL prefix and double-click `[URL Rewrite]` on the right side. Click `[Add Rule]`, and create/edit a new `[Reverse Proxy]` rule that in the end looks like this:
The BSCW server requires additional virtual directory mappings of your Web server. They depend on the values specified for the `SCRIPTS` dictionary in the BSCW server configuration. By default the virtual directory mappings are

- `/sec` 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 10, Server 2016/2019 using the installation program `bsadmin conf_iis`.

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

**IIS 8/9/10 installation**

If the Internet Information Services 8/9/10 are not installed on Windows Server 2016/2019 follow these instructions to install IIS:

- (Windows Server 2016/2019) open the Server Manager Dashboard, open `[Manage → Add Roles and Features]` and follow the wizard.
[Before you login]
[Installation type]
[x] Role-based or feature based installation

[Server Selection]
[x] select a server from the server pool
[<this server>]

[Server Roles]
[x] Web Server (IIS)

[Web Server Role]

[Role Services]
[x] CGI

[Confirmation]

[Install]

Manual IIS 8/9/10 configuration

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

- 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:

<table>
<thead>
<tr>
<th>Directory</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>pub</td>
<td>&lt;bscw-runtime-path&gt;/var/www</td>
</tr>
<tr>
<td>sec</td>
<td>&lt;bscw-runtime-path&gt;/var/www</td>
</tr>
</tbody>
</table>

- 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...]:

<table>
<thead>
<tr>
<th>Request Path</th>
<th>Executable</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>.cgi</td>
<td>&quot;&lt;python-path&gt;/python.exe&quot; -u &quot;%s&quot;</td>
<td>Python Script</td>
</tr>
</tbody>
</table>

- 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 10, Server 2016/2019 by the BSCW administration command bsadmin conf_iis.

Note:

- On Windows Server 2016 only “Desktop Experience” installations are supported.
**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.

### 4.5.4 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 system's control panel.

### 4.6 Folder Mail Delivery

_BSCW does not support BSCW folder mail delivery on Windows._
CHAPTER
FIVE

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_applet.py` Configuration for the Java Applets
- `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 encoders, converters, programs
- `config_countries.py` Specification of country codes
- `config_grid.py` Webbrowser grid layout
- `config_guided_tours.py` Guided tour definitions
- `config_help.py` Contains online help mappings
- `config_html_ui.py` HTML user interface
- `config_icon.py` Icon definitions
- `config_meet.py` Configuration of social network facilities
- `config_menus.py` Configuration of the menu layout
- `config_metadata.py` Configuration of meta data
- `config_mimegroups.py` Application MIME-type grouping
- `config_mime_icons.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
- `config_quicksearch.py` Settings for quick search
- `config_search.py` Search configuration
- `config_service.py` Settings for Windows service
- `config_styles.py` Settings for style sheet handling
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.

2. **Web Server Authentication**

   **Web Server Authentication** used to be the classical” way BSCW handled authentication. To utilize the Apache web serves’ basic authentication module the (encrypted) user credentials are stored within a file `htpasswd` (see `PASSWD`) which was 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).

   **Note:** By default the creation of the `PASSWD` file is disabled, so no shared `htpasswd` file is generated.

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 also:
Section 3.4.1 Apache HTTP Server Configuration for more details

5.2 conf/config.py

This file defines the general server settings and server configuration of the BSCW server instance. Please note 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 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 3.4.2 BSCW instance configuration for configuration hints of SERVER_ROOT. For example

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

Note:
- You have to set SERVER_ROOT 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 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 = 'https://bscw.domain.org'

SERVER_ADMIN

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

SERVER_ADMIN = 'admin@domain.org'

5.2. conf/config.py
Define lists of users with different administrative rights:

SERVER_ADMINS is a list of BSCW users that have full administrator rights (including user administration), e.g.:

```
SERVER_ADMINS = ['admin', 'alice', 'bob']
```

USER_ADMINS defines a list of BSCW users that have restricted administrator rights to manipulate (create, remove, change etc.) users and mail addresses only, e.g.:

```
USER_ADMINS = ['carol', 'dave']
```

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

**See also:**

*SERVER_ADMINS_IP* for domain restrictions

```
SERVER_ADMINS = ['admin']
USER_ADMINS = []
```

**SERVER_ADMIN_CONTACT**

The mail contact address of the BSCW administrator. This is used to reference SERVER_ADMIN_CONTACT in the index page and the 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 = ''
```

**HTTP_LOCAL_PORT**

HTTP_LOCAL_PORT_START

HTTP_LOCAL_HOST_CHECK

HTTP_LOCAL_IS_IIS

HTTP_LOCAL_PORT defines the 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.

**Note:**

- 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/apache24/bscw.conf` also is included.
- See also the virtual host container template file `<bscw-instance-path>/conf/apache24/site.conf` for examples.
If `HTTP_LOCAL_PORT_START` is not None and the package `http` is enabled then `bsadmin start` automatically starts a HTTP server listening on `HTTP_LOCAL_PORT`. For example, with `HTTP_LOCAL_PORT_START = "-p 100 -r 128"` `bsadmin start` automatically executes `bsadmin http -p 100 -r 128 local` (spawning maximal=100 processes with maxlisten=128, this is the default).

Use `bsadmin conf_apache` in order reconfigure the Apache server to forward requests to this server instead of executing bscw.cgi scripts. You must reconfigure and restart Apache again if you reset `HTTP_LOCAL_PORT_START = None`, change `HTTP_LOCAL_PORT` or disable the `http` package!

**Note:** Currently `bsadmin http` and hence `HTTP_LOCAL_PORT_START` works only on Unix systems!

If `HTTP_LOCAL_PORT_START` is not None, then check for special authentication (used by `op_alarm`, `op_mailnotify` etc.) that `REMOTE_ADDR` is one of the given local host addresses, e.g: `HTTP_LOCAL_HOST_CHECK = ('::1', '127.0.0.1')`

If the local server is served by IIS, set `HTTP_LOCAL_IS_IIS` to True to enable some compatibility fixes.

```plaintext
HTTP_LOCAL_PORT = 80
HTTP_LOCAL_PORT_START = None
HTTP_LOCAL_HOST_CHECK = ('::1', '127.0.0.1')
HTTP_LOCAL_IS_IIS = False
```

**EVENTS_SERVER_WS**

Define local endpoints for the real-time events server. Requires a node runtime environment and additional setup steps. Please refer to the admin manual for details.

`EVENTS_SERVER_WS` defines the URL of events server websocket, e.g. `ws://127.0.0.1:3836`. User browsers connect to this endpoint through a reverse proxy, offered by the HTTP server (Apache/IIS).

**EVENTS_SERVER_HTTP** defines the URL of events server http, e.g. `http://127.0.0.1:3837`. Internal endpoint for event and login data exchange.

```plaintext
EVENTS_SERVER_WS = None
EVENTS_SERVER_HTTP = None
```

**COMMUNICATION_SERVER**

Defines an endpoint for real-time conferences via webcam and/or microphone. Requires a ready configured jitsi server (`https://github.com/jitsi/jitsi-meet`).

**COMMUNICATION_SERVER** is the URL of a jitsi server, e.g. `http://127.0.0.1`

```plaintext
COMMUNICATION_SERVER = None
```

**INHERIT_BANNER_TEXT**

**INHERIT_BANNER_IMAGE**
Defines whether a text or a background image in the area of the banner should be passed over the content display or not.

Set `INHERIT_BANNER_TEXT` to `True` if the text inside the banner should be passed over and `False` else.

Set `INHERIT_BANNER_IMAGE` to `True` if the background image inside the banner should be passed over and `False` else.

```
INHERIT_BANNER_TEXT = False
INHERIT_BANNER_IMAGE = True
```

---

**ENABLE_LIVE_COMMUNITY**

**SHOW_CURRENT_CONTEXT_INDICATION**

**SHOW_UPCOMING_PERSONAL_TASKS**

Enables or disables the live realtime information sharing of BSCW members relating to current context and next upcoming tasks

**ENABLE_LIVE_COMMUNITY**: Enables/Disables everything related to the live community. Set `True` if the live community should be activated and `False` else.

**SHOW_CURRENT_CONTEXT_INDICATION**: Enables/Disables the flying avatars which indicates the current context of the corresponding user. Set `True` if the live community should be activated and `False` else.

**SHOW_UPCOMING_PERSONAL_TASKS**: Enables/Disables the sharing of tasks in the personal task list of each member of the BSCW server who is a participant of the live community. Set `True` to activate and `False` else.

```
ENABLE_LIVE_COMMUNITY = True
SHOW_CURRENT_CONTEXT_INDICATION = True
SHOW_UPCOMING_PERSONAL_TASKS = True
```

---

**OFFICE_PROVIDER**

**OFFICE_HOST**

**OFFICE_PORT**

Defines the endpoints for Collabora Online Office (see `office` package for details).

**OFFICE_PROVIDER** defines the office provider, either “C”ollabora or “MS”

**OFFICE_HOST** defines

  - for “C” provider the IP address of Collabora Online Office service. You must execute `bsadmin conf_apache` after changing this IP. By default the IP address of Collabora Online Office is 127.0.0.1
  - for “MS” provider the full WOPI discovery URL, as provided by the hoster.

**OFFICE_PORT** defines the port of the Collabora Online Office service. You must execute `bsadmin conf_apache` after changing this port. By default the port of Collabora Online Office is 9980

```
OFFICE_PROVIDER = 'C'
OFFICE_HOST = '127.0.0.1'
OFFICE_PORT = '9980'
```

---

**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 non-SSL builds). 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 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).

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

**MDA_MTA**
**MDA_MBOX**
**MDA_DOMAIN**
**MDA_HDRMETA**
**MDA_EXTRACTMAIL**
**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 \texttt{MDA\_MTA = ' '} or any unknown MTA will disable the BSCW mail delivery feature (this is the default).

\texttt{MDA\_MBOX} defines the local mailbox name for BSCW mda (this is normally the BSCW user ID name)

\texttt{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)

\texttt{MDA\_HDRMETA} defines which headers are shown in the RFC822 meta profile of an uploaded email, e.g.:

\begin{verbatim}
MDA\_HDRMETA = ['RFC822:from', 'RFC822:to', 'RFC822:cc']
\end{verbatim}

if \texttt{MDA\_EXTRACTMAIL} evaluates to True, the ‘mailaccess’ form shows a preselected option “[x] extract emails into a folder”

\texttt{MDA\_DELMITER = None} (optional) allows to override the MTA default recipient delimiter:

\begin{verbatim}
MDA\_DELMITER = '+' (sendmail/postfix)
MDA\_DELMITER = '-' (qmail)
\end{verbatim}

\texttt{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.

\begin{verbatim}
MDA\_MTA = ''
MDA\_MBOX = 'bscw'
MDA\_DOMAIN = 'domain.org'
MDA\_HDRMETA = ['RFC822:from', 'RFC822:to', 'RFC822:cc']
MDA\_EXTRACTMAIL = False
\end{verbatim}

\textbf{SEND\_LIMIT}

\texttt{SEND\_LIMIT} is a tuple of \texttt{(soft\_limit, hard\_limit)}. If an email should be send by the send operation and the message becomes larger than the \texttt{soft\_limit}, the user gets an hint that, he will send a large email. If the message is larger then the \texttt{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 \texttt{k} (\texttt{K}), \texttt{M}, \texttt{G} or \texttt{T} suffix. E.g. valid values for ten mega-bytes are 10485760 or \texttt{'10M'}.

\begin{verbatim}
SEND\_LIMIT = ('10M', '20M')
\end{verbatim}

\textbf{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. \texttt{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 \texttt{m} ('h', 'd', 'w') suffix. E.g. valid values which specify one week are 604800 (or '604800'), '7d' or '1w'.

\begin{verbatim}
TOKEN\_EXP = None will entirely disable option to send tokens; links can then only be sent to registered users with “get” right.
\end{verbatim}

\textbf{Note:} \texttt{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**

**SEND_HDR**

**SEND_ONBEHALF**

define headers of outgoing BSCW user email messages

If **SEND_RETURN_PATH** is 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).

If **SEND_HDR** is True (default) BSCW appends a Sender: <SEND_ADMIN> and a Reply-To: <user-mailaddr> header to honor RFC 822/4021 agent submission. If additionally **SEND_ONBEHALF** is True (default: False) the From: header is rewritten as From: "BSCW-Administrator on behalf of <user>" <SEND_ADMIN>.

SEND_RETURN_PATH = ''
SEND_HDR = True
SEND_ONBEHALF = False

### 5.2.3 Server access

**SERVER_ADMINS_IP**

List (or tuple) of IPv4 or IPv6 addresses or networks. Networks may be specified in prefix (CIDR) or netmask notation. If not empty the remote address must match one of the given domains for a user in **SERVER_ADMINS** or **USER_ADMINS** to become BSCW Administrator (see below).

SERVER_ADMINS_IP = [  
    '1.2.3.4',  # administrator IP address  
    '1.2.3.0/24',  # administrator IP net (prefix)  
    '1.2.3.0/255.255.255.0',  # administrator IP net (netmask)  
]

SERVER_ADMINS_IP = []

**MAY_REGISTER**

**ALLOW_MAIL_AUTH**

**ALLOW_MAIL_UNLOCK**

MAY_REGISTER defines a list of BSCW users names or pattern tuples of email addresses assigned to 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** or **USER_ADMINS** who have this right anyway. Please see **RESTRICT_MAIL** for a description how to define pattern tuples, example
There are three special cases: if \texttt{MAY\_REGISTER} is:

- \texttt{[]} - 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.
- \texttt{None} - registration is allowed for all registered users, but self-registration is forbidden.

\textbf{Note:} Only \texttt{MAY\_REGISTER = [\]} allows self-registration by URL: \url{http://bscw.domain.org/pub/bscw.cgi?op=rmail}

If \texttt{ALLOW\_MAIL\_AUTH} is set \texttt{True} (default) users may reset their password via mail token authentication. If set \texttt{False} mail token authentication is disallowed (and only the administrator may reset forgotten passwords).

If \texttt{ALLOW\_MAIL\_UNLOCK} is set \texttt{True} users may unlock their account after being locked by the system (e.g. after multiple wrong login attempts as defined in \texttt{BADPASS}) via mail token authentication. If set \texttt{False} (default) users may only be unlocked by a BSCW administrator.

\textbf{MAY\_REGISTER = None}
\textbf{ALLOW\_MAIL\_AUTH = True}
\textbf{ALLOW\_MAIL\_UNLOCK = False}

\textbf{MAY\_CREATE\_MAILADDRESS}

\texttt{MAY\_CREATE\_MAILADDRESS} defines a list of BSCW user names or pattern tuples of email addresses that have the right to create external mail addresses - e.g. suggest participants for an appointment scheduling. This is in addition to users from \texttt{MAY\_REGISTER}, who have this right anyway. Please see \texttt{MAY\_REGISTER} and \texttt{RESTRICT\_MAIL} for a description of special values and how to define pattern tuples, example:

\begin{verbatim}
MAY\_CREATE\_MAILADDRESS = [
    'username',
    ('[^@]*@forbidden.com', 0),
    ('[^@]*@allowed.com', 1),
]
\end{verbatim}

There is another special case, which will restrict creation of new email addresses exactly to users from \texttt{MAY\_REGISTER}:

\textbf{MAY\_CREATE\_MAILADDRESS = False}

\textbf{Note:} This right also enables a user to find and disclose all email addresses that are known to the system!

\textbf{MAY\_CREATE\_MAILADDRESS = [\]}

\textbf{REGISTER\_DETAILS}
\textbf{TERMS\_AND\_CONDITIONS}
\textbf{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:

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

or allow registration without additional details, simply set:

```python
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.

The page referenced by TERMS_ANDCONDITIONS should describe the server’s terms of use (which the users agree to accept). The page referenced by PRIVACY_POLICY should contain rules that apply to data protection declaration (resp. “Datenschutzerklärung” [DE]). Both links will be shown in the menu of the index page and in the footer of all (container) pages.

To support language dependent links, add the language shortcut (uppercase) to the variable name, e.g. use TERMS_ANDCONDITIONS_DE for a German page.

**Note:**

- English is the default language which is displayed for all languages without language dependent link.
- With the legal validity of the EU - General Data Protection Regulation (GDPR), it will be necessary to provide a data protection declaration for BSCW instances operated in the European Union. See the BSCW administration manual section 2.3 EU - General Data Protection Regulation for further details.

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

**DEFAULT_TELL_LASTLOG**

Users’ default value for “Show my presence and last login date to other users”

```python
DEFAULT_TELL_LASTLOG = True
```

**ABOUT_SITE**

If ABOUT_SITE is defined it should point to a link (URL) that contains legal notice of the server’s website (aka ‘Publisher’ [EN] or ‘Impressum’ [DE]). The link will be shown on the index page’s menu and as visible link in the footer of all (container) pages.

To support language dependent links, add the language shortcut (uppercase) to the variable name, e.g. use ABOUT_SITE_DE for a German page.

**Note:** English is the default language which is displayed for all languages without language dependent link.
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

DEFAULT_USER_LANG

The default user language is taken as default language for new users (or email addresses) in case no language is assigned explicitly.

Note: upon (self-)registration, the browser language is assigned to the new user.

DEFAULT_USER_LANG = 'en'

RESTRICT_MAIL
RESTRICT_EXT_MAIL
RESTRICT_SEND

A list (or tuple) of pairs (pattern, boolean) to restrict email addresses:

- RESTRICT_MAIL restricts the set of email addresses that are accepted by BSCW for e.g. registration or invitation.
- RESTRICT_EXT_MAIL restricts the set of external email addresses that are accepted by BSCW functions, such as appointment scheduling.

Note: RESTRICT_EXT_MAIL extends RESTRICT_MAIL patterns, i.e. RESTRICT_MAIL patterns are applied before patterns defined in RESTRICT_EXT_MAIL.

- RESTRICT_SEND restricts the set of user email addresses that are allowed to send email via BSCW.

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_... do not apply, if the inviting user is in the list of SERVER_ADMINS or USER_ADMINS.

RESTRICT_MAIL = ()
RESTRICT_EXT_MAIL = ()
RESTRICT_SEND = ()
MINPASSLEN

EXPPASS

EXPACCT

LOG_EXPIRED_USERS

BADPASS

CRYPT_HASH

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/)
  – python3-cracklib package (http://www.nongnu.org/python-crack/)

are installed. To enable this feature set

MINPASSLEN = 'libcrack'

• If MINPASSLEN == 'libcrack' cracklib may be configured using the PYTHONCRACK dictionary as follows:

    PYTHONCRACK = {
        'MIN_LENGTH': 8,  # minimal password length
        'UP_CREDIT': -1,  # at least 1 upper case character
        'LOW_CREDIT': -1, # at least 1 lower case character
        'DIG_CREDIT': -1, # at least 1 digit character
        'OTH_CREDIT': -1, # at least 1 other character
        'DIFF_OK': 3,     # min. difference old and new passwd
    }

(see http://www.nongnu.org/python-crack/doc/crack-api.html)

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_HASH specifies the hash algorithm used when storing password hashes. Possible values for
CRYPT_HASH are:

• 'sha512_crypt' to use SHA512 (default)
The following hash algorithms are for legacy support:

- 'sha256_crypt' to use SHA256
- 'md5_crypt' to use the Linux MD5 Modular Format (deprecated)
- 'des_crypt' to use DES (deprecated)

```
MINPASSLEN = 8
EXPPASS = 0
EXPACCT = 0
LOG_EXPIRED_USERS = 'expired_users'
BADPASS = 0
CRYPT_HASH = 'sha512_crypt'
```

Settings for a virus scanner to scan files after upload or before download:

- To enable a virus scan on download allows potential undetected viruses at upload time to be detected later with ongoing newer virus definitions of the virus scanner.
- If a virus is found, the file will be quarantined in an encrypted ZIP archive with a predefined password (VIRUS_PASSWD).
- The download virus scan is repeated every VIRUS_DB_CHECK seconds, if a new virus database version is detected (VIRUS_DB_VERS).
- (Unix) When enabling the virus scan feature the 7-Zip file archiver must be installed.

VIRUS_CHECK defines in which situation files are scanned. Valid values are 0 (never), 1 (on upload) or 3 (on upload and download)

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_VIRUSES_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.

VIRUS_WAIT is the time (in seconds) the server will wait for a scan or quarantine process to complete before responding to a download request.

VIRUS_DB_VERS specifies the command string to get the version of virus database. If it is defined, a repetitive scan will not be done before the database has been updated.

VIRUS_DB_CHECK gives the interval (in seconds) between repetitive virus scans for a document (default: 86400 (1 day)). Within this interval, a document will not be re-scanned even if new virus definitions are available. The interval must be at least one hour.

VIRUS_PASSWD specifies the password that is used to protect access to a document that has been quarantined.

Example for Avast command line scan utility:
To grant access to BSCW files add the `avast` system user to the `bscw` group, e.g.:

```
$ su
# vigr
bscw:x:500:avast
```

Example for NAI McAfee VirusScan:

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

Example for ClamAV:

```
SCANFILE = "/usr/bin/clamdscan --fdpass --no-summary '%(file)s'"
NO_VIRUS_FOUND = [0,]
VIRUS_FOUND = [1,]
VIRUS_DB_VERS = "/usr/bin/clamdscan -V"
```

5.2.4 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 Social 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
```
By default BSCW authenticates users 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:

<table>
<thead>
<tr>
<th>timeout</th>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>do not verify authentication tag (low security!)</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>authentication tag does not expire</td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>authentication tag expires after n minutes</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>location</th>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>use op_login for BSCW authentication</td>
<td></td>
</tr>
<tr>
<td>URL</td>
<td>jump to URL for authentication</td>
<td></td>
</tr>
</tbody>
</table>

E.g.:

```python
COOKIE_AUTHENTICATION = ('bscw_auth', None, 120)
```

Cookie authentication is disabled with:

```python
COOKIE_AUTHENTICATION = None
```

If the Microsoft 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.

**Warning:** digest authentication is considered to be insecure. Better use https only with basic authentication.

**Note:** 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:

```python
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:

```python
OPEN_ID_DEFAULT = ('', '')
```

**Note:** COOKIE_AUTHENTICATION must be set and location (see above) must be None when OpenIDs are used.

Also the python3-openid package must be installed (https://github.com/openid/python-openid)
OpenID registration and authentication is disabled with:

```python
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 {}.

```python
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**:

```python
'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.

**Note:**

- 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 6.22.3 *Shibboleth Authentication* for further configuration hints.

```python
AUTH_MODE = 'Basic'
COOKIE_AUTHENTICATION = None
OPEN_ID_DEFAULT = None
FEDERATIONS = ()
```
If `COOKIE_AUTHENTICATION` is enabled the authentication tag expires after the specified time-out. Certain operations (like presence bar, portal widgets and editor auto-save) will send automated requests in short intervals which keeps the user’s session alive.

If `COOKIE_AUTH_FORCE_LOGOUT` is enabled however, a session timeout is forced if no ‘real’ user interaction happens within the specified time.

```
COOKIE_AUTH_FORCE_LOGOUT = False
```

**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.

```
POST_CHECK_REFERER = 1
```

**LOG_REG_USERS**

**LOG_REMOVED_USERS**

If `LOG_REG_USERS` is not empty, all newly registered users at the system are logged in the file `LOG_REG_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_REG_USERS = 'registered_users'
LOG_REMOVED_USERS = 'removed_users'
```

**PASSWD**

`PASSWD` - Web server password file

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

```
PASSWD = 'htpasswd'
```

**Note:**

- the web server password file is not used by BSCW any longer, but might be of interest for other web applications.
If you change the PASSWD file, you must also point your HTTP server to the new file.

See also:
Section 3.4.1 Apache HTTP Server Configuration for Unix
Section 4.5.2 Apache HTTP Server Configuration or 4.5.3 IIS Configuration for Windows

PASSWD = ''

SCRIPTS
SECURE_SCRIPTS
CREATE_SCRIPTS

Define the CGI scripts that may be called the HTTP server. Given the URL http://bscw.domain.org/testing/sec/bscw.cgi/25, the HTTP server will split this URL into:

the SCRIPT_NAME "/testing/sec/bscw.cgi" and
the PATH_INFO "/25"

The BSCW server splits the script name further into:

the prefix "/testing/sec/" 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.

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.

Note:

- 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 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 = '/sec/
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
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.

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; https://www.apereo.
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 unde-
defined or empty SSO support is disabled (default).

SSO_SCRIPT (optional) defines an additional alternate script name of the CGI script which is redi-
ected 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.:
SCRIPTS_ALIASES define alternate script alias prefix paths for the secure prefix:

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

A 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.:

```python
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.

```python
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.

```python
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:

```python
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 2007, 2010, 2013 or 2016 documents may be saved immediately on the BSCW server.

Prerequisites: WebDAV enabled, MS Internet Explorer 11 or above, MS Office 2007, 2010, 2013 or 2016 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.openxmlformats-officedocument.spreadsheetml.sheet':
        'Excel.Sheet',
    'application/vnd.ms-excel.sheet.macroenabled.12':
        'Excel.Sheet',
    'application/vnd.openxmlformats-officedocument.presentationml.presentation':
        'PowerPoint.Slide',
}

(continues on next page)
EDITOR_SETTINGS

EDITOR_SETTINGS defines the settings for the built-in HTML (TinyMCE) editor:

- 'code_sample' toggles source code formatting

EDITOR_SETTINGS = {
    'code_sample': False,  # disable formatting of source code
}

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.

See also:

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.openxmlformats-officedocument.spreadsheetml.sheet',
    'application/vnd.ms-excel.sheet.macroenabled.12',
    'application/vnd.openxmlformats-officedocument.presentationml.presentation',
    'application/vnd.ms-powerpoint.presentation.macroenabled.12',
    'application/vnd.openxmlformats-officedocument.wordprocessingml.document',
    'application/vnd.ms-word.document.macroenabled.12',
}

5.2.5 BSCW appearance settings

USER_SEARCH_LIMIT
Maximum number of matching hits by a User Search

\texttt{USER\_SEARCH\_LIMIT = 100}

\texttt{MEMBER\_SEARCH}

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

\texttt{MEMBER\_SEARCH = 1}

\texttt{MAX\_VERSIONS}

\texttt{MAX\_VERSIONS\_KEEP}

\texttt{MAX\_VERSIONS\_LIMIT}

Controls the autoversion behavior for newly created documents:

If \texttt{MAX\_VERSIONS} is

- \texttt{1} (default), new created documents are not set under version control.
- \texttt{0}, new created documents are automatically set under version control and all revised versions will be stored.
- \texttt{<n>}, new created documents are automatically set under version control, but only the latest \texttt{<n>} revised versions will be stored.

Revising version \texttt{<n>+1} will automatically remove the oldest revision.

If the \texttt{MAX\_VERSIONS\_KEEP} list is not empty, all version ids of a document are matched against the RE patterns. The boolean of the first matching pattern decides if a version id gets removed when \texttt{MAX\_VERSIONS} is set to store the latest \texttt{<n>} revised versions.

For example:

\begin{verbatim}
MAX\_VERSIONS\_KEEP = [
    ('.*\.0$', True),
]
\end{verbatim}

will avoid the removal of all version ids ending with "\texttt{.0}".

If \texttt{MAX\_VERSIONS\_LIMIT} is set to a value > 1, the maximum number of user configurable version revisions (cf. \text{[> Change > Properties] form}) is limited to \texttt{MAX\_VERSIONS\_LIMIT}.

\textbf{Note:}

- This does not affect the global setting (cf. \texttt{MAX\_VERSIONS})
- When defining \texttt{MAX\_VERSIONS\_LIMIT}, \texttt{MAX\_VERSIONS} must be equal or less \texttt{MAX\_VERSIONS\_LIMIT} and unequal 0.

\begin{verbatim}
MAX\_VERSIONS = 1
MAX\_VERSIONS\_KEEP = []
MAX\_VERSIONS\_LIMIT = 0
\end{verbatim}

\texttt{GRID\_BROWSER\_LIMIT}

When all entries of a folders’ contents grid are loaded at once, the grid can do certain things locally without additional server requests, like fast scrolling or sorting entries. In large folders, however, it is not advisable to load all entries at once to avoid exceptional memory consumption of client browsers.
GRID_BROWSER_LIMIT = 450

GRID_PAGE_SIZE
The number of entries that folder’s contents grids fetch at once. A larger number means less server requests, but longer load times per server request. Should never exceed GRID_BROWSER_LIMIT

GRID_PAGE_SIZE = 150

SCRIPT_POLL_INTERVAL
This is the time interval in which the JavaScript components will reload their contents. The value is set in seconds. So by default e.g. the contacts view will reload its contents every 60 seconds, and new microblog posts will pop up each 60 seconds.

SCRIPT_POLL_INTERVAL = 60

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 :menuselection:`[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
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 = {
    'default/default',
    'default/default_dark',
}

RATE_COLORS

Colors for URL ratings: ['none', 'poor', 'passable', 'fair', 'good', 'excellent']

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

AVATAR_COLORS

A list of colors with tuples of (background_color, font_color) for auto-generated user avatars (if user doesn’t provide a photo) Note: this feature requires GraphicsMagick: http://www.graphicsmagick.org

Both color names or hex values (e.g. ‘white’ or ‘#ffffff’) can be used - as supported by GraphicsMagick: http://www.graphicsmagick.org/color.html

AVATAR_COLORS = [
    ('#004F80', '#FFFFFF'), # dark blue 100%
    ('#00B8F2', '#000000'), # light blue 100%
    ('#74B917', '#000000'), # light green 100%
    ('#0778A5', '#FFFFFF'), # blue 100%
    ('#FFC819', '#000000'), # yellow 100%
    ('#F28502', '#000000'), # orange 100%
    ('#C40046', '#FFFFFF'), # red 100%
    ('#89D581', '#FFFFFF'), # gray 100%
    ('#89D48', '#FFFFFF'), # dark red 100%
    ('#23614E', '#FFFFFF'), # dark green 100%
]

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 = '*/'

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 indexde.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 1 and the URL has the form local:<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 = 1 and add (as admin) a URL named local:var/log/sys.log to a workspace to provide access to your <bscw-runtime-path>/var/log/sys.log.

Note:

• A user, not being administrator, cannot create a local:... URL, even if LOCAL_URL_PREFIX is not (yet) enabled. A leading slash will be interpreted as an absolute file path. I.e. the 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 = 0
```

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 = ''
```

**INDEX_PAGE_EXTRA_CONTENTS**

Define some folder id in here, where some extra contents for your index page are defined. Inside this folder you can add the following stuff.

- Define sub folders with files like (de.html, en.html, ...) that will be added as sub sites in your index page navigation bar.
- By default the folder name is used as page title in navigation bar. You can change the displayed name by setting some description for each individual html file.
- By adding some html file (de.html, en.html, ...) into your folder you can define some extra html message that is placed below the displayed login field.

**Note:** CSS from HTML file headers will be ignored. Just use the theme’s style.less file to add styles. Then rebuild your CSS and your index page. Besides, you can use inline styles.

```
INDEX_PAGE_EXTRA_CONTENTS = 1234
```

The folder with the id 1234 could look like this:
My Folder (with ID 1234) has the following contents:

<table>
<thead>
<tr>
<th>Subpage</th>
<th>de.html</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subpage</td>
<td>en.html</td>
</tr>
</tbody>
</table>

(also note: links will be sorted by name. To add some order just add some leading 1_, 2_, 3_, ... to your filename)

Note: Everytime after you have made changes inside your folder, you have to run “bsadmin index_page” once again.

INDEX_PAGE_EXTRA_CONTENTS = 0

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 = 'https://www.bscw.de/en/social/help'
SERVER_HELP_DE = 'https://www.bscw.de/social/help'

- SERVER_INFO - BSCW server info page - by default it shows the index page in the scripts directory for anonymous (see SCRIPTS).
- SERVERCANCEL - 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/

FMT_DISTINCT_NAME

A pattern format to build a distinct name from the favoured name, an extension and a number. The format must take name and extension as positional arguments and the number as keyword 'number'.

Examples:

```
'{}/{}({number}){}/'.format('README', '.txt', number=2) => README (2).txt
'{}/{}({number}){}/'.format('folder', '', number=2) => folder (2)
```

The special value None will disable the feature:

FMT_DISTINCT_NAME = None

BSCW_UI_NAME

Which name should be displayed at HTML user interface.

0: favoured name
1: favoured name, add extension
2: favoured name, strip extension
3: distinct name
4: distinct name, add extension
5: distinct name, strip extension

BSCW_UI_NAME = 0

5.2.6 Optional BSCW packages

PACKAGES

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

```
'approval',       # Document Approval
'blog',          # Blogs
'bsync',         # Synchronization between BSCW and Desktop
'chat',          # Chat
'expire'         # User account expiration
'exportpdf',     # Export views to PDF (requires reportlab)
'FlowFolder',    # Flow Folder
'http',          # pre-forking BSCW HTTP server
```

(continues on next page)
Attention: The PACKAGES list is maintained by the bsadmin package command. Do not manually edit the PACKAGES list.

To enable a package run

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

To disable a package run

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

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 “Periodic Report” or the “Direct Email” notification (depending on the UNO service configuration).

By default a daily report is sent to new users, but each user may decide to unsubscribe from 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.

Accordingly the direct email notification is enabled by default for each user so each user may decide to disable 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.:

```python
SERV_UNO_TIMES = {
    'TdelayDirect': 60.,
    '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': 300.: Delay the next direct notification for the same user five minutes. This is to avoid an overload of the user.

'TdelayDaily': 5.: Add a delay of five seconds between daily notification mails. This is to avoid an overload of the mail server 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': 2:2 retries that are delayed with TdelayRetry.

'TdelayFailed': 21600.: After MaxRetry the notification is delayed 6 hours (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': 4: Maximum number of parallel running mail processes. This will also determine the load of the BSCW server and the mailer. Note: For more throughput on big server machines this value might be increased.
'QueueInfo': 20: Show job queue status after 20 jobs are queued (use values: \(2^n \times \text{MaxJobs}\))

'ReportTime': 01:31: Start daily/weekly reports at 01:31 (must be \(\geq 00:00\) and \(< 07:00\))

'WeekReportDay': '7': Weekly reports on Sunday (must be \(\geq 1\) [Monday] and \(<= 7\) [Sunday])

\[\text{WSREPORT} = 1(0)\] enable (disable) the periodic (daily/weekly) report.

\[\text{WSREPORT\_DIRECT} = 1(0)\] enable (disable) the direct email report.

**Note:** When `bscw.adm.bs_servuno` does not run the periodic 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 periodic 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 for most of the awareness services (like periodic report or external services), with the values

\[
\text{read} = 1; \text{create} = 2; \text{move} = 4; \text{change} = 8
\]

By default all event types are subscribed, except of read events. (\(\text{create} + \text{move} + \text{change} = 14\))

`DEFAULT\_EVENTMASK\_DIRECT` defines the default event type subscription mask for the direct email notification. By default no event types are preselected, so users won’t be notified about any events immediately, but may select some event types for certain folders of interest only.

`DEFAULT\_EVENT\_REPORT\_DAILY` defines the default frequency for periodic email report which may either be daily (1) or weekly (0)

For example, set

\[
\begin{align*}
\text{WSREPORT\_DIRECT} &= 1 \\
\text{AUTOSUBSCRIBE\_REPORT\_DIRECT} &= 1 \\
\text{DEFAULT\_EVENTMASK\_DIRECT} &= 2
\end{align*}
\]

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.)

`WSREPORT\_EVENT\_LIMIT` defines a size limit of the periodic workspace report: in order to prevent too long notification emails the number of events can be limited (use `WSREPORT\_EVENT\_LIMIT = 0` for unlimited size)

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

\[
\text{REPORTLOG} = \text{'report.log'}
\]
WSREPORT_DIRECT = 1
AUTOSUBSCRIBE_REPORT = 1
AUTOSUBSCRIBE_REPORT_DIRECT = 1
DEFAULT_EVENTMASK = 14
DEFAULT_EVENTMASK_DIRECT = 0
DEFAULT_EVENT_REPORT_DAILY = 1
WSREPORT_EVENT_LIMIT = 500
REPORTLOG = ''

ALWAYS_CREATE_READEVENTS
CREATE_READEVENTS_OPTION

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.

Setting CREATE_READEVENTS_OPTION = 1 enables the manager of a workspace (respectively folder) to define (for this context) if read events should be tracked or not. If no option is defined, the setting is inherited from the parent folder. If no option is defined on any parent folder, the default option (i.e. ALWAYS_CREATE_READEVENTS) is used.

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 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 current store file. So it must be garbage collected from time to time (every day is recommended!). The garbage collector copies actual data from store file to GC file 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 store file 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

5.2. conf/config.py
Persistent alarm store, file store and temporary files. Relative paths are relative to the BSCW database directory ($<bscw-runtimepath>/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 10, Server 2016/2019. 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**
**RMUSR_ARC**
**RMUSR_VER**
**RMUSR_ENC**

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_VER - archive all versions of a document: True (default) or False
RMUSR_ENC - pathname encoding: 'UTF-8' (default)

```
RMUSR_DIR = 'rmuserarc'
```
SERV_ACCESS

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

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`.

```plaintext
SERVER_LOG = 'server.log'
BSCW_LOG = 'bscw.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.

```plaintext
BSCW_UMASK = 7
```

### DBMOD_TAB

- **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://www.oracle.com/database/berkeley-db) and the additional bsddb module. Python 3 requires the installation of the bsddb3 module (`python3-bsddb3`)

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.

  ```plaintext
  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).

  ```plaintext
  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:

  ```plaintext
  DBMOD_MINKEY = (9*(DBMOD_PAGESIZE/1024), 5*(DBMOD_PAGESIZE/1024))
  ```

A good working heuristic value pair seems to be:
# BSCW Administrator Documentation, Release 7.3.2

**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

---

**GSERV**

**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.

<table>
<thead>
<tr>
<th>GSERV</th>
<th>SERV_ALARM</th>
</tr>
</thead>
<tbody>
<tr>
<td>'DbSocket'</td>
<td>'AlarmSocket'</td>
</tr>
</tbody>
</table>

---

**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-7.3.2-<rev>-py3%/doc/devel/BSCW|relmaj|-API.zip

Please note that BSCW is distributed with some API examples. These Python scripts are included in the BSCW distribution in bscw-7.3.2-<rev>-py3%/etc/src-aux/remote_client

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).
BSCW can act as a proxy for “cross-domain requests”, required for some AJAX features like used in the portal. This proxy requires authentication and is thus only open to your BSCW users. You can turn the proxy on and off using XDPROXY_ENABLED. The proxy will refuse to download any URL that does not match any regular expression in XDPROXY_URLS.

If you have the libxml2 and libxslt Python packages installed, the proxy can also apply some data transformations, which are given in XDPROXY_TRANSFORMATIONS. Each transformation specifies the source mimetype (a regular expression that has to match the mimetype currently downloaded), the XSLT transformation to apply to it and the resulting target mimetype. Example:

```python
XDPROXY_TRANSFORMATIONS = {
    'somedata': (r'(?i)^text/xml;\s*?charset=UTF-8$',
                 '/opt/bscw/xslt/somedata.xsl',
                 'application/json; charset=UTF-8')
}
```

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:

```python
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:

```python
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.

```python
# BSCW_LOGGING = {
#     'sys': ('WARN', '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 (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'.

DOWNLOADBUTTON_CONFIRM defines the size limit from where quick downloads will be started only after some confirm request. The download button is placed inside the toolbar. By default everything smaller 100M will be downloaded directly. 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'.

```ini
WAIT_ARCHIVING = 10
ARCHIVE_LIMIT = '2G'
DOWNLOADBUTTON_CONFIRM = '100M'
```

packages_state

Please do not change packages_state. It controls automatic enabling/disabling of new/obsolete PACKAGES in bsadmin update_defaults.

```ini
packages_state = 4
```

5.3 conf/config_actions.py

The config_action.py configuration file allows to redefine roles.

5.4 conf/config_applet.py

The config_applet.py configuration file provides definitions for Java Applets.

5.5 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.
5.6 conf/config_clientmap.py

The `config_clientmap.py` configuration file defines the mapping between web browsers and their supported options.

See also:
The comments in this file for further descriptions.

5.7 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.

5.8 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 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 file

   Example:
   ```python
   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',
   ),
   ]
   ```

2. 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

(continues on next page)
of the conversion. If you have more than one tool for the same conversion, the one with the best quality is chosen.

The shell command to convert a file info is information about what is lost during the conversion

Example::

```python
Converters = [
    ('application/pdf', 'text/plain', '0.5',
    '/usr/bin/pdftotext -enc UTF-8 %(src)s %(dest)s',
    'layout/images',
    ),
    ('application/postscript', 'text/plain', '0.5',
    '/usr/bin/ps2ascii -sOutputFile=%(dest)s -q -dBATCH %(src)s',
    'layout/images',
    ),
    #...
]
```

3. The Programs list contains 5-tuples (name, path) with

<table>
<thead>
<tr>
<th>name</th>
<th>the external converter program name</th>
</tr>
</thead>
<tbody>
<tr>
<td>path</td>
<td>the system path to the external program name</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>%(py)s</th>
<th>absolute path of the python executable</th>
</tr>
</thead>
<tbody>
<tr>
<td>%(cnv)s</td>
<td>absolute path of the BSCW converters directory</td>
</tr>
<tr>
<td>%(src)s</td>
<td>the absolute path of the source file</td>
</tr>
<tr>
<td>%(dest)s</td>
<td>the base name of destination file</td>
</tr>
<tr>
<td>%(pid)s</td>
<td>process id of the converter process</td>
</tr>
</tbody>
</table>

For the Converters list additionally the following pattern can be used:

| %(charset)s | character set encoding for documents with a "text/*" content-type. |

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 avoid automatic manipulation of the following lists by `bsadmin update_defaults` enable the following line below (use at own risk, future updates may fail):

```python
__keep__ = ['Encoders', 'Converters']
```

To regenerate the converter file, e.g. after you installed new converters or adapted 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):

```bash
$ bin/bsadmin update_defaults -h
Usage: bsadmin update_defaults [-s|-e] [-i] [-v|-vv|...] [-w|-ww|...]
```
bsadmin update_defaults -h

Update conf/__init__.py and conf/config.py

optional arguments:
-s    skip import check
-e    exit on package error
-i    reinitialize conf/__init__.py
-v    verbosity
-w    warning level
-h    show this help message and exit

$ bin/bsadmin update_defaults -s -v
...
Found "Programs" (located):
  '7z': '/usr/bin/7za'
  'a2ps': '/usr/bin/a2ps'
  'antiword': '/usr/bin/antiword'
  'bzip2': '/bin/bzip2'
  'cjpeg': '/usr/bin/cjpeg'
  'compress': '/usr/bin/compress'
  'convert': '/usr/bin/convert'
  'djpeg': '/usr/bin/djpeg'
  'gif2tiff': '/usr/bin/gif2tiff'
  'gm': '/usr/bin/gm'
  'gzip': '/bin/gzip'
  'html2ps': '/usr/bin/html2ps'
  'html2text': '/usr/bin/html2markdown'
  'java': '/usr/bin/java'
  'latex2html': '/usr/bin/latex2html'
  'lynx': '/usr/bin/lynx'
  'markdown2': '/usr/local/bin/markdown2'
  'pdftotext': '/usr/bin/pdftotext'
  'perl': '/usr/bin/perl'
  'phantomjs': '/usr/local/bin/phantomjs'
  'pildriver': '/usr/bin/pildriver.py'
  'ps2ascii': '/usr/bin/ps2ascii'
  'tar': '/bin/tar'
  'tiff2ps': '/usr/bin/tiff2ps'
  'uncompress': '/bin/uncompress'
  'unoconv': '%(py)s %(cnv)s/unoconv/unoconv --pipe=%(pid)s'
  'unzip': '/usr/bin/unzip'
  'uuencode': '/usr/bin/uudecode'
  'uunencode': '/usr/bin/uuencode'
  'w3m': '/usr/bin/w3m'
  'wkhtmltopdf': '/usr/local/bin/wkhtmltopdf'
  'zip': '/usr/bin/zip'

conf/config_convert.py : updated...

5.9 conf/config_countries.py

The config_countries.py file defines country codes (based in ISO 3166 standard) for selections lists.

5.10 conf/config_grid.py

The config_grid.py file defines the appearance and column layout of the Web interface grid.
5.11 conf/config_guided_tours.py

The `config_guided_tours.py` configures the available user guided tours.

5.12 conf/config_help.py

The `config_help.py` file defines mappings from the BSCW context sensitive help to online help HTML pages.

5.13 conf/config_html_ui.py

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:

```python
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': [
            # (colbit, position, presets, ui_profiles)
            (col_icon, 100, V_ANY, ui_yes, ),
            (col_name, 200, V_ANY, ui_yes, ),
            # ...
        ]
    }
}
```

5.14 conf/config_icon.py

The `config_icon.py` file allows to define mappings of names to a icon resources - i.e. a CSS class names.

5.15 conf/config_icons.py

The `config_icons.py` file maps BSCW objects to image files.

5.16 conf/config_meet.py

This is the configuration file for synchronous collaboration tools, i.e. social networks.

5.17 conf/config_menus.py

The `config_menu.py` file specifies the BSCW 7 menu configuration.
5.18 conf/config_metadata.py

The config_metadata.py file specifies the meta data for BSCW objects.

5.19 conf/config_mimegroups.py

The config_mimegroups.py file maps MIME-types of different applications in groups, eg. Microsoft Office.

5.20 conf/config_mime_icons.py

The config_mime_icons.py file configures for the MIME-types icons in BSCW 7.

5.21 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.

5.22 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:

```python
name = (MIME-Type, suffix0, suffix1, ...)
```

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);
- suffix0 ... suffixn are 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.

See also:
http://www.iana.org/assignments/media-types/ for more information on MIME-types

Examples of entries in the list are:

```python
access = ('application/vnd.ms-access', 'mdb')
aiff = ('audio/x-aiff', 'aif', 'aiff')
...  
zip = ('application/zip', 'zip')
```
5.23  conf/config_mobile_ui.py

The config_mobile_ui.py file specifies the user interface setting for the BSCW mobile interface.

5.24  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.

5.25  conf/config_portlets.py

The config_portlets.py file provides configuration custom portlets.

5.26  conf/config_prio_categ.py

The config_prio_categ.py file configures settings for priorities and categories.

5.27  conf/config_quicksearch.py

The config_quicksearch.py file provides configuration for the BSCW quick search.

5.28  conf/config_search.py

The config_search.py file allows to configure the BSCW main search keys. Configured search keys are indicated with MainSearchKey() entries in this file. All defined main search keys are immediately available in the “+” menu of a search operation, e.g.

```
MainSearchKey('org:theme')
MainSearchKey('org:region')
MainSearchKey('org:location')
MainSearchKey('org:status')
```

5.29  conf/config_service.py

The config_service.py file provides configuration the Windows service.

5.30  conf/config_styles.py

The config_styles.py file provides configuration for style sheet handling.
5.31 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-7.3.2-<rev>-py3/*/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-7.3.2-<rev>-py3/*/bscw/msg/en/lg_msgconfig.py file to this location. Now you can edit <bscw-runtime-path>/bsext/msg/en/lg_msgconfig.py and adapt 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 file for English is the “default” language 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.
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
     ```
   - to disable distributed a BSCW package run:
     ```
     bin/bsadmin package -d <pkg-name>
     bin/bsadmin package -d ldap
     ```
   - to re-enable a distributed BSCW package (and update installed resources) run:
     ```
     bin/bsadmin package -r <pkg-name>
     bin/bsadmin package -r 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 enable an external BSCW package run:
     ```
     bin/bsadmin package -e <pkg-name> <path>
     bin/bsadmin package -e fhg_fit bsext/fhg_fit
     ```
   - to disable an external BSCW package run:
     ```
     bin/bsadmin package -d <pkg-name>
     bin/bsadmin package -d fhg_fit
     ```
   - to re-enable an external BSCW package (and update installed resources) run:
     ```
     bin/bsadmin package -r <pkg-name>
     bin/bsadmin package -r ldap
     ```

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 this 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/

Some pre-build binaries are provided by the pylucene-extra project at http://code.google.com/a/apache-extras.org/p/pylucene-extra/

BSCW 7.3.2 supports pylucene 3.6.2

Important:

• Additionally pylucene requires an installed Java Runtime Environment (JRE) 8
• (Windows) After upgrading your Java Runtime Environment (JRE) to a new release the new installation path must be adapted manually in the Windows system “Path” environment variable. Afterwards a system restart is required.

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.

6.1.1 Configuration

This package is not enabled by default and requires some software installation (i.e. pylucene - see above) and allows optional 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 as described in section Software for BSCW Preview (see Unix or Windows).

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 5.8 conf/config_convert.py) 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 `bsadmin create_index`

- **INDEX_QUERY_HELP**
  link to the query syntax documentation

**Note:** this actually depends on the installed version of pylucene! (see **INDEX_QUERY_OPERATOR_AND** below for possible changes in BSCW)

- **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](image)

  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](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.

  **Warning:** 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`).

The following configuration directive is configured in the BSCW package configuration file `<bscw-runtime-path>/conf/PyLucIndex/config.py`

- **INDEX_JVM_MAXHEAP**

  Max heap for Java VM (lucene only) (default: '512m'). Increase this value if you experience `OutOfMemoryError` exceptions while index creation, e.g.:

  `INDEX_JVM_MAXHEAP = '2048m'`
• **LUCENE_JVM_ARGS**

  additional arguments passed to lucene’s JVM via `lucene.initVM(vmargs)` should be list of string arguments or empty list:

  ```
  INDEX_JVM_ARGS = ['-Djava.awt.headless=true',]
  ```

• **INDEX_MAX_TXTSIZE**

  Max document size for text documents to be indexed. Lucene's Java VM may fail with OutOfMemoryError on very large documents that are typically binary files with wrong MIME-Type. BSCW uses some heuristics to detect binary files, but will also skip files with certain size anyway. Default limit is 50 MB text file size (= 52428800 bytes):

  ```
  INDEX_MAX_TXTSIZE = 52428800
  ```

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](http://lucene.apache.org) for details on how this affects indexing.

### 6.1.2 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. You may check the status of the indexer using

   $ `bin/bsadmin create_index -v`

   and stop a running indexer process using

   $ `bin/bsadmin create_index -x`

   To start the indexing process on Unix systems you may use for example:

   ```
   $ nohup bin/bsadmin create_index -cqt >/dev/null 2>&1 &
   ```

The commandline usage is as follows:

$ `bin/bsadmin create_index`

**Usage:**

`bsadmin create_index`  `-c` `-cu` `-i` `-s` `-o` `-t` `-U` `-h`

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-c</code></td>
<td>create new index (forced if no index exists)</td>
</tr>
<tr>
<td><code>-cu</code></td>
<td>create new index &amp; force update of document text representations</td>
</tr>
<tr>
<td><code>-i</code></td>
<td>incremental index update</td>
</tr>
<tr>
<td><code>-s</code></td>
<td>scan database continuously</td>
</tr>
<tr>
<td><code>-o</code></td>
<td>suppress periodic optimization (optimize only on start)</td>
</tr>
<tr>
<td><code>-t</code></td>
<td>display timer info at exit</td>
</tr>
<tr>
<td><code>-U</code></td>
<td>unlock at first (dangerous)</td>
</tr>
</tbody>
</table>

(continues on next page)
-v verbose mode (or status report if used as single option)
-q quiet
-r <min> report interval (default 30 min, 0: no report)
-R <hour> automatic restart in '+<hour>' or at 0 < <hour> < 24
-x stop indexer
-xu stop indexer and cleanup document text representations
-xz stop indexer and cleanup index files
-xuz stop indexer and cleanup all indexer files
-h show this help message and exit

Note: option -u is only possible in conjunction with option -c (i.e. all text files will be removed before new index is created) or in conjunction with option -x (i.e. all text files will be removed after indexer is stopped - allows fresh restart).

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.

Option -v can be used (as single option) to check the indexer status. The indexer is typically running as a background process and automatically started with the BSCW server. More details may also be found in the indexer logfile (in <bscw-runtime-path>/var/log/index.log)

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:

```bash
$ bin/bsadmin search
query pylucene index
positional arguments:
query query
optional arguments:
-h, --help show this help message and exit
-s show index statistics
-a search all fields (default: content search)
-i search by ID only
-c show hit count only
-v verbose
-l lang language
```

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"
```

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.

### 6.1.3 Index creation and update

If the package is enabled 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 `-sr60` is used).

Thus recommended usage of the indexer is to initially create the index manually by invoking the following commands:

```
$ bin/bsadmin package -e PyLucIndex
$ bin/bsadmin create_index -cqt
```

and then let BSCW update the index continuously.

For this purpose you only need to (re)start your BSCW server after the `bsadmin create_index` finished to create the initial index e.g.:

```
$ bin/start_servers -k  # UNIX
$ bin/start_servers
> bin\bsadmin stop [-s]  # Windows
> bin\bsadmin start [-s]
```

**Note:** The indexer logs progress and errors to the configured log file (in `<bscw-runtime-path>/var/log/index.log`). Startup (or failure to start the indexer) during start/stop of the BSCW server is also logged in the main BSCW log file (in `<bscw-runtime-path>/var/log/bscw.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 -iU
```

manually to incrementally index all missing objects. Again, after `bsadmin create_index` finished updating the index restart your BSCW server, e.g.:

```
$ 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:

- option `-xz` will stop the indexer and remove the index files. This allows a quick rebuild without updating text representations (which is time consuming).
• option *-xuz* will stop the indexer and remove the index files and all document text representations. This is the ultimate “from-scratch” solution as all index-related data is cleaned up before rebuilding the index (you may also want to `rm var/log/index.log`).

In both cases you may then re-create the index using `bsadmin create_index -cqt`.

Finally restart the BSCW server again as described above, to let BSCW update the index continuously (see `create_index` above). This method will result in a ‘fresh’ (and up-to-date) index and newly created text representation of all indexable documents (if option `-u` is given).

To re-create the index simply use the following command sequence:
```
$ nohup /bin/sh -c "bin/bsadmin create_index -xz; bin/bsadmin create_index -cq; → bin/start_servers" > /dev/null 2>&1 &
```

# 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 ldap3 package. Ldap3 natively implements an RFC 1823 API (see OpenLDAP [http://www.openldap.org](http://www.openldap.org)).

## 6.2.1 Installation

To install the BSCW LDAP module

1. The BSCW LDAP module needs the `ldap3` Python package. `ldap3` implements a native Python LDAP client library.
   - On Linux systems the `ldap3` package of the distribution should be installed.
     Packages name(s) for common Linux distributions:
     - Debian based systems: `python3-ldap3`
     - Fedora based systems: `python3-ldap3`
     Alternatively use the Python package manager `pip`:

     ```
     $ su -
     # pip3 install ldap3
     ```
   - On Windows systems install `ldap3` using the Python package manager `pip`:

     ```
     > pip install ldap3
     ```

2. To enable the BSCW `ldap` copy the default template file to the instance configuration directory as follows

   ```
   # su - bscw
   $ cd $HOME
   $ mkdir -p srv/<bscw-instance>/conf/ldap
   $ cp lib/bscw-7.3.2-<rev>-py3?/bscw/conf/ldap/config.py srv/<bscw- →instance>/conf/ldap
   ```
   and run:

   ```
   $ cd srv/<bscw-instance>
   $ bin/bsadmin package -e ldap
   ```

3. Adapt the configuration file `<bscw-runtime-path>/conf/ldap/config.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 a 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.

• **may_register_ldap** is a list of BSCW users that have the right to register LDAP users - i.e. invite new users to the system or to a workspace. This is in addition to `SERVER_ADMINS`, who have this right anyway.

There are two special cases: if `may_register_ldap` is

[]: then registration of new LDAP users is allowed for all users. This allows all users and anonymous to invite new users to the system.

None: then registration of new LDAP users is allowed for all but anonymous.

---

**Note:**

- only `may_register_ldap = []`, allows self-registration by LDAP user login
- `may_register_ldap` behaves equal to `MAY_REGISTER` for found LDAP user objects. By default self-registration of found LDAP user objects is allowed (which is the behaviour of previous BSCW versions)
- alternatively may want to use the setting of `MAY_REGISTER` also for `may_register_ldap`. In this case define:

```python
from conf.config import MAY_REGISTER
may_register_ldap = MAY_REGISTER
```

• **auto_registration** defines DN patterns and search patterns for auto_registration during login. If a user is not registered at BSCW but her DN can be found on a LDAP server with one of the patterns listed in auto_registration, then BSCW makes an attempt to register the user automatically and assigns (binds) the DN to the user object if the registration process succeeds. three patterns are possible here (%s is substituted by the login name):

- a pattern that expands to the DN directly:

```python
'cn=%s,o=snakeoil,c=de'
```

- a 2-tuple that specifies the LDAP server default binding (base DN) and a search expression for user name search:

```python
('o=snakeoil2,c=de', '(uid=%s)')
```

- a 3-tuple that specifies the LDAP server default binding (base DN) and a search expression for user name search and a search expression for email address search:

```python
('o=snakeoil2,c=de', '(uid=%s)', '(mail=%s)')
```

The latter two patterns result in a 2-step process for the required binding: At first the DN is looked up on the LDAP-server using the default binding. Then a bind is tried with the resulting DN (must be unique) and the given password. In case a 3-tuple is given the search pattern is determined by the given login name. If the login name contains a '@' character the mail address search pattern ('mail=%s'), otherwise the user name search pattern is used.

• **auto_registration_email** allows to send a registration mail. Use `auto_registration_email = 'reg_done'` if you want the standard registration mail
sent to an automatically registered user. You might set the registration mail language using:
```
auto_registration_email_lang = 'de'
```

- **auto_registration_roles** defines initial roles, quota limit class or auto-invitation to communities for automatically registered users. The list consists of tuples:

```
('attribute=value', 'R[012]rolename'),
('attribute=value', 'R[012]rolename', 'limitclass'),
('attribute=value', 'R[012]rolename', 'limitclass', 'community-id').
```

**Note:**
- the role 'R[012]rolename' must be assignable for user objects i.e. it must appear in the list `cl_action.user_roles`.
- the quota limit class 'limitclass' must be defined with `bsadmin quota limit` in advance.
- the community with the object-id 'community-id' must be created in advance.
- 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.

Possible patterns:
```
('ou=pupil', 'R2restricted'),
('ou=development', 'R2manager', '@manager'),
('ou=development', '', '@manager'), # No user role is assigned
('ou=development', 'R2manager', '@manager', '12345'),
```

- **auto_may_register** defines DN patterns and search patterns to determine if an user has the right to register mail addresses (see `<bscw-runtime-path>/conf/config.py`: `MAY_REGISTER`). If an user matches a given DN or search pattern in `auto_may_register` and the configuration directive `MAY_REGISTER` restricts the registration of mail addresses, this user is additionally allowed to register mail. Three patterns as described above at `auto_registration` are possible here.

- **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:**
- BSCW does password checking by LDAP only if the BSCW server and not the HTTP server does authentication, e.g. when cookie authentication is enabled or BSCW gets the `HTTP_AUTHORIZATION` value). Because this is not a very fast way to do authentication, it might be an alternative to configure the HTTP server to do LDAP authentication (e.g. via the
Apache HTTP server auth_ldap module) instead of setting use_ldap_passwords = 1 which requires all users to pass LDAP authentication.

- If the Apache HTTP Server auth_ldap module is used use_ldap_passwords must be set to 3, otherwise the BSCW change password action interferes with the auth_ldap modules internal password cache.

- When using BSCW authentication, digest authentication is not possible in combination with LDAP, because BSCW requires the plain (textual) password to authenticate the credential against LDAP.

* ldap_searches defines a list of member search options (qid, pattern) for the workspace invite member action (op_addmb):

- qid is an unique identifier for the search and must be translated in <bscw-runtime-path>/conf/msg/*/ldap/lg_msgconfig.py.

- pattern is a LDAP query where '%(query)s' is replaced by the user input of the addmb search form

- search subtree of defined base DN(s) for the given query:

  ```python
  ('mb_search ldap_uid', 'cn=%(query)s*'),
  ('mb_search ldap_uid', '(|(cn=%(query)s*)(uid=%(query)s*))'),
  ('mb_search ldap_uid', '(sAMAccountName=%(query)s*)'),
  ```

6.2.2 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.

- 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 Desktop Widgets

The AirDesktop package provides a feature for BSCW to provide desktop widgets using Adobe Air. Widgets may display information stored within BSCW (folders, tasks etc).

This package is enabled by default in a new BSCW server instance. 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 airdesktop
```
6.4 Document Approval

The approval package supports a standardized quality approval process while document production. After document creation the document may be checked by one more persons and is finally released. The state of documents running through this approval process is displayed at the user interface.

You may want to provide different global defaults for your users in the by creating the configuration file <bscw-runtime-path>/conf/approval/config.py. The possible configuration directives and their defaults are as follows:

- MAY_RESET_APPROVAL
  controls if the approval process is reset after an approved document is edited or replaced. (Default: True)

- APPROVAL_UNIQUE_REVIEWER
  enforce if reviewers must be unique in an approval, i.e. when enabled any reviewer may participate only once in a review process. (Default: False)

This package is enabled by default in a new BSCW server instance. 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 approval
```

6.5 Blog (Weblogs)

The blog package 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 instance. 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
```

6.6 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 instance. 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 (%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
    ```
  – This feature is only available for Windows Systems (client-side).
  – This feature is only available in the professional edition of BSCW.

  **See also:**
  Chapter 8 *BSCW Help* for further details.

### 6.7 Chat

The chat package allows you to send multimedia messages to other BSCW users in real time, creating a feeling similar to a spoken conversation. No further configuration is required.

### 6.8 Expire

The expire package sends an email notification to the user when the account was expired with additional informations. The notification email may be customized by creating the configuration file `<bscw-runtime-path>/conf/expire/config.py` with the following configuration directives:

• **EXPIRE_DELETE_DAYS**
  defines the number of days after expiration when the account will be deleted:
  ```
  EXPIRE_DELETE_DAYS = 30
  ```
  **Note:** This defines just a hint for the email notification, account deletion must be done manually by the administrator.

• **EXPIRE_CONTACT_MAIL**
  defined an email address for questions (defaults to `SERVER_ADMIN`):
  ```
  EXPIRE_CONTACT_MAIL = None
  ```

How to enable automatic account expiry see *user account expiry*.

This package is not enabled by default in a new BSCW server instance. No additional software installation or configuration is required on server-side. The package may be enabled by running:

```
bin/bsadmin package --enable expire
```
6.9 Export PDF

The `exportpdf` package 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/Pillow):
     https://pypi.python.org/pypi/Pillow
   - The ReportLab PDF Library:
     https://pypi.python.org/pypi/reportlab
   - On Linux systems use preferred the packages of your distribution:
     - Debian based systems: python3-pil python3-reportlab
     - Fedora based systems: python3-pillow python3-reportlab
   - On Windows systems or if your Linux distribution does not provide packages you need the Python package manager `pip` to install the packages:

   ```
   > pip install pillow
   > pip install reportlab
   ```

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.

6.10 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 instance. 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
```

6.11 Http

The `http` package implements a pre-forking BSCW HTTP server. This means a master process pre-loads the BSCW code library, spawns a pool of separate worker HTTP processes and routes requests to spare worker processes.

Using this technique greatly speeds up request processing. Incoming requests are immediately served on arrival without the overhead of creating new processes or loading BSCW code modules. Load tests have shown an average performance increase of 30% compared to the traditional Apache HTTP server CGI.
This package is **not** enabled by default in a new BSCW server instance and is **only** available on Unix based BSCW systems. No additional software installation is required on server-side.

To enable the pre-forking BSCW HTTP server the `HTTP_LOCAL_PORT_START` directive must be defined and the `http` package must be enabled as follows:

**Important:** When using the pre-forking BSCW HTTP server all configuration changes become only active after a **restart** of the BSCW HTTP server, which is performed from the CLI running:

```
bin/bsadmin http restart
```

or on the administration BSCW status page [Options → Admin → Status] by clicking the [Restart integrated http service] entry.

## 6.11.1 Enabling the BSCW HTTP server

1. Stop the BSCW instance services:

   ```
   bin/bsadmin stop
   ```

2. Enable the `http` package:

   ```
   bin/bsadmin package -e http
   ```

3. Edit the instance configuration file `<bscw-runtime-path>/conf/config.py` and define a unused `localhost` port for the pre-forking BSCW HTTP server, e.g.:

   ```
   HTTP_LOCAL_PORT = 8080
   ```

   **Note:** The `localhost` port must be free and may not be occupied by another service (such as the Apache HTTP server).

   Next define a BSCW HTTP server start command, e.g.:

   ```
   HTTP_LOCAL_PORT_START = "-p 100 -r 128"
   ```

4. Start the BSCW instance services:

   ```
   bin/bsadmin start
   ```

   Beside the usual BSCW services additionally this starts a pre-forking BSCW HTTP server with a maximum of 100 worker processes and a maximum listen queue length of 128 requests.

5. Update your Apache HTTP server configuration:

   ```
   bin/bsadmin conf_apache
   ```

   Ensure your Apache HTTP server enabled the `proxy` and `proxy_http` modules and restart the HTTP server as root user:

   - Debian based systems:

     ```
     $ su -
     # a2enmod proxy proxy_http
     # systemctl restart apache2
     ```

   - Fedora based systems:
6.11.2 Disabling the BSCW HTTP server

1. Stop the BSCW instance services:

   `bin/bsadmin stop`

2. Disable the `http` package:

   `bin/bsadmin package -d http`

3. Restore in the instance configuration file `<bscw-runtime-path>/conf/config.py` the `HTTP_LOCAL_PORT` to a Apache HTTP server controlled localhost port, e.g.:

   `HTTP_LOCAL_PORT = 80`

   and set a BSCW HTTP server start command to `None`:

   `HTTP_LOCAL_PORT_START = None`

4. Start the BSCW instance services:

   `bin/bsadmin start`

   This starts the BSCW services without the pre-forking BSCW HTTP server again.

5. Update your Apache HTTP server configuration:

   `bin/bsadmin conf_apache`

   Disable the Apache HTTP server `proxy` and `proxy_http` modules (if not longer required) and restart the HTTP server:

   - Debian based systems:

     ```
     $ su -
     # a2dismod proxy proxy_http
     # systemctl restart apache2
     ```

   - Fedora based systems:

     ```
     $ su -
     # vim /etc/httpd/conf.modules.d/00-base.conf     # RHEL 7
     # vim /etc/httpd/conf.modules.d/00-proxy.conf
     # systemctl restart httpd
     ```

6.12 Incognito

The `incognito` package provides an optional feature for BSCW to anonymize read events in a certain workspace. When enabled each role shows an additional access right “Get (Incognito)”. When activated all read event in this workspace are anonymized.

This package is **not** enabled by default in a new BSCW server instance. No additional software installation or configuration is required on server-side. The package may be enabled again by running:
The *metaprofiles* package allows to provide user-defined metadata profiles for BSCW objects.

This package is enabled by default in a new BSCW server instance. 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 metaprofiles
```

### 6.14 Online Office

The *office* package provides integration with WOPI based office applications, e.g. Collabora Online.

**Note:**
- The free version of Collabora Online allows editing 10 documents with 20 concurrent users. For more users a commercial license is required (please ask license@orbitteam.de for licensing details).
- WOPI based office applications require a HTTPS enabled web server.

#### 6.14.1 Collabora Online configuration

The easiest way to run a Collabora Online instance is to use the pre-configured Docker container image, see https://www.collaboraoffice.com/code/ for details. The container image is deployed using the commands:

```
$ docker pull collabora/code
$ docker run -t -d -p 127.0.0.1:9980:9980 \
   -e 'domain=<dot-escaped-domainname>' \
   -e 'username=admin' \
   -e 'password=<password>' \
   --name "office" \
   --restart always --cap-add MKNOD collabora/code
```

The `<dot-escaped-domainname>` requires “dot-escaping” of the BSCW instance name `SERVER_ROOT`, e.g.:

```
bscw.domain.org
=>
bscw\.domain\.org
```

Generally (escaped) regular expressions are allowed as `<dot-escaped-domainname>`, such as:

```
.*\\.domain\\.org
=>
\\.*\\.domain\\.org
```

The Collabora Online instance operates in read only mode and does not allow writing documents. To allow editing (including writing) of documents the host IP address of each BSCW instance must be whitelisted in a configuration file within the container image.

For this purpose, the host IP address of the BSCW instance must be inserted in the file `/etc/loolwsd/loolwsd.xml` (note you have to escape dot characters (. -> \.)).

For example, suppose the IP address of your BSCW instance is `10.11.12.13`. In this case the following entries:
must be added in the `<post_allow>`...</post_allow> section of the loolwsd.xml file as follows:

```xml
<net desc="Network settings">
  <proto type="string" default="all" desc="Protocol to use IPv4, IPv6 or all for both"></proto>
  <listen type="string" default="any" desc="Listen address that loolwsd binds to. Can be 'any' or 'loopback'."></listen>
  <service_root type="path" default="" desc="Prefix all the pages, websockets, etc. with this path."/>
  <post_allow desc="Allow/deny client IP address for POST(REST)." allow="true">
    <host desc="The IPv4 private 192.168 block as plain IPv4 dotted decimal addresses.">192.168.[0-9]{1,3}.[0-9]{1,3}</host>
    <host desc="Ditto, but as IPv4-mapped IPv6 addresses">::ffff:192.168.[0-9]{1,3}.[0-9]{1,3}</host>
    <host desc="The IPv4 loopback (localhost) address.">127.0.0.1</host>
    <host desc="Ditto, but as IPv4-mapped IPv6 address">::ffff:127.0.0.1</host>
    <host desc="The IPv6 loopback (localhost) address.">::1</host>
    <host desc="BSCW instance (IPv4-mapped IPv6)">::ffff:10.11.12.13</host>
    <host desc="BSCW instance (IPv4)">10.11.12.13</host>
  </post_allow>
</net>
```

You may also want to disable the signing server endpoint URL by removing the URL https://app.vereign.com from the `<document_signing_url></document_signing_url>` entry as follows:

```xml
<document_signing_url desc="The endpoint URL of signing server, if empty the document signing is disabled" type="string" default=""></document_signing_url>
```

Edit the `/etc/loolwsd/loolwsd.xml` file within the Collabora Online container as follows:

1. Copy the loolwsd.xml file from the Collabora container to your local host and enter the above mentioned entries for the BSCW instance host IP address:

   ```bash
   $ docker cp <containername>:/etc/loolwsd/loolwsd.xml .
   $ docker exec <containername> chown lool:lool /etc/loolwsd/loolwsd.xml
   $ docker exec <containername> mv /etc/loolwsd/loolwsd.xml /etc/loolwsd/loolwsd.xml~
   $ docker stop <containername>
   [containername](continues on next page)
   ```

2. Edit the loolwsd.xml file and add the additional IP entries as described above:

   ```bash
   $ vi loolwsd.xml
   ```

   Be aware the `docker run` from above altered the following entries in the loolwsd.xml file:
   - the `domain=` definition edits the `<wopi>` and the `<webdav>` entries.
   - the `admin=` definition edits the `<admin_console>` entries.

3. Copy the loolwsd.xml file back to the Collabora Online container:

   ```bash
   $ docker cp loolwsd.xml <containername>:/etc/loolwsd/loolwsd.xml~
   $ docker exec <containername> chown lool:lool /etc/loolwsd/loolwsd.xml~
   ```

4. Restart the Collabora Online container:

   ```bash
   $ docker stop <containername>
   $ docker start <containername>
   ```

(continues on next page)
To update the Collabora Online image proceed as follows:

```bash
$ docker stop <containername>
$ docker rm <containername>
```

and repeat all steps from above.

### 6.14.2 BSCW office package configuration

The `office` package is not enabled by default on new BSCW servers and requires external components. Before enabling the `office` the following additional packages `requests` and `pycryptodome` are required.

Packages name(s) for these Linux distributions:

- Debian based systems: `python3-requests`
- Fedora based systems: `python3-requests`

or use `pip3` to download and install the packages:

```bash
$ su -
# pip3 install requests
# pip3 install pycryptodome
```

If disabled, the package may be enabled by running:

```bash
bin/bsadmin package -e office
```

After the package is activated, the package must be configured using the following directives in the instance configuration (`<bscw-runtime-path>/conf/config.py`):

- `OFFICE_PROVIDER` defines the office provider, either “C”ollabora or “MS”
- `OFFICE_HOST` – for “C” provider
  defines the IP address of Collabora Online Office service. You must execute `bsadmin conf_apache` after changing this IP. By default the IP address of Collabora Online Office is `127.0.0.1`:

  ```python
  OFFICE_HOST = '127.0.0.1'
  ```

  – for “MS” provider
  defines the full WOPI discovery URL, as provided by the hoster.
- `OFFICE_PORT` defines the port of the Collabora Online Office service. You must execute `bsadmin conf_apache` after changing this port. By default the port of Collabora Online Office is `9980`:

  ```python
  OFFICE_PORT = '9980'
  ```

Ensure your Apache HTTP server modules `proxy`, `proxy_wstunnel`, `proxy_http`, and `ssl` are enabled and restart the HTTP server as root user:

- Debian based systems:
Fedora based systems:

```bash
$ su -
# vi /etc/httpd/conf.modules.d/00-base.conf  # RHEL 7
# vi /etc/httpd/conf.modules.d/00-proxy.conf
# vi /etc/httpd/conf.modules.d/00-ssl.conf
# systemctl restart httpd
```

If the OFFICE_HOST or OFFICE_PORT directives have been changed the Apache HTTP server configuration should be recreated with:

```
bin/bsadmin conf_apache
```

This updates the Apache HTTP server configuration file (<bscw-runtime-path>/conf/apache24/site.conf) with a Collabora Online configuration block which must be integrated on the virtual host configuration of the BSCW instance in order to run the BSCW `servdiscovery` service.

### 6.15 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/poll/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

- **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’

## 6.16 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 [https://www.apereo.org/products/cas](https://www.apereo.org/products/cas)), Shibboleth, a standards-based, open source middleware software which provides SSO even across organizational boundaries (see [https://www.shibboleth.net/](https://www.shibboleth.net/)) and OpenID (see [https://openid.net](https://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:

   ```python
   CAS_URI = 'http://sso.domain.org:8080/cas'
   
   SSO_PREFIX = '/cas/
   SSO_COOKIE = ('bscw_cas', None, 120)
   
   (SSO_PREFIX, { 'mode': AUTH_MODE, 'cookie': SSO_COOKIE })
   in SCRIPTS_ALIASES, e.g.:
   
   SCRIPTS_ALIASES = {
   '/sec/': [
   (SSO_PREFIX, { 'mode': AUTH_MODE, 'cookie': SSO_COOKIE })),
   ]
   }
   
   2. Create a new Apache HTTP server configuration with
$ ./bin/bsadmin conf_apache -n
Configure public prefix '/pub/' (Apache 24) ...
(No authentication)
Configure secure prefix '/sec/' (Apache 24) ...
(HTTP_AUTHORISATION passed to BSCW)
(Cookie authentication enabled)
Configure secure prefix '/cas/' (Apache 24) ...
(HTTP_AUTHORISATION passed to BSCW)
(Cookie authentication enabled)

Creating Apache HTTP server configuration files in
<bscw-runtime-path>/conf/apache24
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.:

> su -
# systemctl restart apache2
# systemctl restart httpd

## 6.16.2 OpenID

In order to activate OpenID single-sign-on registration and authentication see https://openid.net.

The BSCW OpenID module needs the *python3-openid* Python package.

- On Linux systems the *python3-openid* package of the distribution should be installed.

  Packages name(s) for common Linux distributions:

  - Debian based systems: *python3-openid*
  - Fedora based systems: *python3-openid*

  Alternatively use the Python package manager *pip*:

  $ su -
  # pip3 install python3-openid

- On Windows systems install *python3-openid* using the Python package manager *pip*:

  > pip install python3-openid

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.

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.

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/ on how to install and configure a Shibboleth Service Provider in your environment. Another good source of information with configuration examples are the “guides for SWITCHaai” at https://www.switch.ch/aai/guides/.

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 should 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.

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 commented entry for 'BscwTest':

```python
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),
}
```

Note:

- 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:

```plaintext
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:

<table>
<thead>
<tr>
<th>Shibboleth 2.x</th>
<th>Shibboleth 1.3</th>
<th>BSCW key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shib_Application_ID</td>
<td>HTTP_SHIB_APPLICATION_ID</td>
<td>shib_app_id</td>
</tr>
<tr>
<td>Shib_Identity_Provider</td>
<td>HTTP_SHIB_IDENTITY_PROVIDER</td>
<td>shib_idp</td>
</tr>
<tr>
<td>mail</td>
<td>HTTP_SHIB_INETORGPERSON_MAIL</td>
<td>email</td>
</tr>
<tr>
<td>givenName</td>
<td>HTTP_SHIB_INETORGPERSON_GIVENNAME</td>
<td>givenname</td>
</tr>
<tr>
<td>sn</td>
<td>HTTP_SHIB_PERSON_Surname</td>
<td>surname</td>
</tr>
<tr>
<td>org-dn</td>
<td>HTTP_SHIB_SWISSEP_HOMEORGANIZATION</td>
<td>org</td>
</tr>
<tr>
<td>telephoneNumber</td>
<td>HTTP_SHIB_PERSON_TELEPHONENUMBER</td>
<td>phone</td>
</tr>
<tr>
<td>homePhone</td>
<td>HTTP_SHIB_INETORGPERSON_HOMEPHONE</td>
<td>homephone</td>
</tr>
<tr>
<td>mobile</td>
<td>HTTP_SHIB_INETORGPERSON_MOBILE</td>
<td>mobile</td>
</tr>
<tr>
<td>preferredLanguage</td>
<td>HTTP_SHIB_INETORGPERSON_PREFERREDLANGUAGE</td>
<td>language</td>
</tr>
</tbody>
</table>
ACCEPT_WEB SERVICES = 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.

## 6.18 Tasks

This package 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 also:

Chapter 8 *BSCW Help* for further details.

## 6.19 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 creating the configuration file `<bscw-runtime-path>/conf/WebFolder/config.py` and appending one of the following variables:

- **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 newTreetemplate and newQuerytemplate 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 definition is newDdefaultstyle.

• 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-used page types.

See also:

Chapter 8 BSCW Help for end-user help concerning Website Folders.
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 `conf/config.py`),
- by direct editing the configuration files described in section 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 user menu in the right upper corner of the interface.
Administrator users explicitly need to log in a second time with their password to gain BSCW administrator rights. Without this additional administrator authentication no administrative rights are applied to their account.

After successful login the administrator status is indicated by an additional [Admin] menu in the right upper corner of the interface:

Using the administrative menu allows to perform different administrative tasks. The [Admin] menu contains the following entries:

- the [Status] entry displays the BSCW status page,
- the [Server Monitor (JSON)] entry creates an access token to get statistics data for monitoring in JSON format,
- the [New User] and [User administration] the entries provide BSCW user access management functionality to search, modify, create or delete new users,
- the [Send E-Mail to all users] entry allows to send administrative email to all users,
- the [Workspace List] entry displays a table of all existing shared workspaces,
- the [Server Configuration], [MIME-Types], [Converters] and [Encoders] menu entries allow BSCW server configuration via the web interface,
- the [Upgrade licence] entry summarizes the BSCW licence management and provides functionality to apply for a new licence by contacting the OrbiTeam licence service,
• the [Exit admin mode] entry disables the administrative rights of the current user again.

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.

The Services section shows all running BSCW services: the database server (bs_servdb), the user notification service (bs_servuno), the access service (bs_servaccess), the pre-forked BSCW HTTP server (op_http) and the alarm service (bs_servalarm). The alarm service additionally displays the actual queue length of pending alarms or jobs which are scheduled for execution.
Note:

- If the alarm service (bs_servaccess) in the service section shows the status **not running** there is a problem with the web server **localhost** configuration. Please refer to the system log file `<runtime>/var/log/sys.log` to get a more detailed error description and fix the webserver configuration, see *Apache HTTP Server Configuration* for details.

- The queue length may increase quickly due to schedules preview calculation jobs which will be executed sequentially one by one.

If you click on [Start garbage collection] the garbage collector is started, which will become necessary if you have downloaded a new licence and want to install it.

The [Restart integrated http service] option is only available if the pre-forking BSCW HTTP server is enabled (see *http*). Using the BSCW HTTP server requires a restart after each configuration change.

The [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 licence] form allows to perform licence 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] menu 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:

![User administration form](image)

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:
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 User administration.

Clicking on [New User] shows the following form, which is used to add a new email address to the system.

After entering an email address and clicking on [OK] the new email address must be allocated to a (new) user. When selecting the option [x] Send self registration message to e-mail address a self-registration notification is sent to email address.

Alternatively when selecting the default option [x] Allocate e-mail address to user an user is allocated to the new email address within the next form:
The allocation of an e-mail address opens the chat view of the new user. To change settings of the new user, open the info page by clicking the white (i) overlay in the users’ icon:

The info page allows to apply Administrator actions such as

- renaming the user (Name)
- changing the password or user locking (Password)
- altering the profile settings of the user (Personal Profile)
- changing the users’ preferences (Preferences)
- editing the users’ roles (Add Role, Edit Role, Assign Role)
- destroying the user (Permanently destroy)
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 User administration).

7.1.3 Configuration

The configuration section of the administrative menu allows the BSCW configuration via the web interface. The entries

- Server Configuration
- MIME Types
- Messaging Services
- Converters
- Encoders

are related to the configuration files described in chapter 5 Configuration of BSCW Servers. Opening an entry will display a form to modify the corresponding configuration file.

Clicking on [Server Configuration] 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.
Note: Be careful when editing the BSCW instance configuration. Configuration errors may lead to an dysfunctional BSCW instance.

Clicking on [Converters] or [Encoders] will show the a form to configure existing converters or encoder commands:

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 5.8 `conf/config_convert.py`, which provides a method to automatically locate required conversion commands on the BSCW server system).

### 7.1.4 BSCW licence management

The BSCW licence management allows to apply for a licence resp. to prolong an expired licence and to install a granted licence. To apply for a licence open the Upgrade Licence form by clicking the [Upgrade Licence] button:
Next press [OK] which allows to edit a licence request, to download or to install a licence:

- When applying for a licence please fill in the licence request form and press [OK] to submit the licence request to OrbiTeam. Further details about the licence acquisition process are given in chapter 9 *BSCW license*.

- When downloading a granted licence you have to accept the licence agreement.
and run a garbage collection by pressing [Start garbage collection] on the BSCW status page:

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 10, Server 2016/2019 systems. Therefore the normal usage of the bsadmin script is only as:

```
bin/bsadmin start
bin/bsadmin stop
bin/bsadmin 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@orbitteam.de, otherwise you may damage your database).

When using the bsadmin command without any arguments, it displays the list of possible arguments as follows:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bsadmin archive</td>
<td>archive an artifact via command line</td>
</tr>
<tr>
<td>bsadmin chkconfig</td>
<td>check configuration make directories and cgi scripts (I)</td>
</tr>
<tr>
<td>bsadmin chkfiles</td>
<td>check for missing document files</td>
</tr>
<tr>
<td>bsadmin chkjobs</td>
<td>check for blocked jobs</td>
</tr>
<tr>
<td>bsadmin chksearchbag</td>
<td>checks SearchBag for superfluous entries</td>
</tr>
<tr>
<td>bsadmin chkurl</td>
<td>change URL object locations</td>
</tr>
<tr>
<td>bsadmin chkworkspaces</td>
<td>rebuild workspace list</td>
</tr>
<tr>
<td>bsadmin chpwd</td>
<td>change user password and lock/unlock user</td>
</tr>
<tr>
<td>bsadmin chtype</td>
<td>change content type of given document</td>
</tr>
<tr>
<td>bsadmin clean_anon</td>
<td>remove objects in &quot;anonymous&quot; top level folders</td>
</tr>
<tr>
<td>bsadmin conf_apache</td>
<td>BSCW Apache web server configuration</td>
</tr>
<tr>
<td>bsadmin conf_crontab</td>
<td>BSCW crontab configuration</td>
</tr>
<tr>
<td>bsadmin conf_systemd</td>
<td>BSCW systemd configuration</td>
</tr>
<tr>
<td>bsadmin conf_iis</td>
<td>BSCW IIS configuration</td>
</tr>
<tr>
<td>bsadmin conf_tzdata</td>
<td>configure timezone data</td>
</tr>
<tr>
<td>bsadmin create_index</td>
<td>generate search index</td>
</tr>
<tr>
<td>bsadmin db2to3</td>
<td>Convert python2 database to python3 database</td>
</tr>
<tr>
<td>bsadmin dbcheck</td>
<td>database check/repair</td>
</tr>
<tr>
<td>bsadmin dbcopy</td>
<td>Copy database (D)</td>
</tr>
<tr>
<td>bsadmin dbfindaddr</td>
<td>Find addresses accepted for maildelivery into folders (D)</td>
</tr>
<tr>
<td>bsadmin dbfindid</td>
<td>Find all database offsets for object with given id (D)</td>
</tr>
<tr>
<td>bsadmin dbfindmodules</td>
<td>Find (all) modules in which classes are looked up (D)</td>
</tr>
<tr>
<td>bsadmin dbfindobj</td>
<td>Find (all) objectids for given classes</td>
</tr>
<tr>
<td>bsadmin dbfindref</td>
<td>Find (all) references of given objects (i.e. ids)</td>
</tr>
<tr>
<td>bsadmin dblist</td>
<td>List, dump or debug database records (D)</td>
</tr>
<tr>
<td>bsadmin dbscan</td>
<td>scan database; print position, key, class and id (D)</td>
</tr>
<tr>
<td>bsadmin dbsizes</td>
<td>Print record sizes in database file (Store) (D)</td>
</tr>
<tr>
<td>bsadmin dbsummary</td>
<td>print a summary of all classes in the database (D)</td>
</tr>
<tr>
<td>bsadmin dbtable</td>
<td>Check (or print) database tables (D)</td>
</tr>
<tr>
<td>bsadmin dbtruncate</td>
<td>Truncate database at offset (D)</td>
</tr>
<tr>
<td>bsadmin du</td>
<td>show/update BSCW database disk usage</td>
</tr>
<tr>
<td>bsadmin extract</td>
<td>extract an artifact from commandline</td>
</tr>
<tr>
<td>bsadmin fix_event_queue</td>
<td>Fix Event Queue</td>
</tr>
<tr>
<td>bsadmin fix_keys</td>
<td>remap mail address and user keys</td>
</tr>
<tr>
<td>bsadmin fsck</td>
<td>check file tree for obsolete files and directories</td>
</tr>
<tr>
<td>bsadmin garbage</td>
<td>BSCW garbage collector</td>
</tr>
<tr>
<td>bsadmin getconfig</td>
<td>get configuration info from config.py</td>
</tr>
<tr>
<td>bsadmin index_page</td>
<td>generates an index page for the script directories</td>
</tr>
<tr>
<td>bsadmin info</td>
<td>prints basic info about BSCW server configuration</td>
</tr>
<tr>
<td>bsadmin ldapbind</td>
<td>change user LDAP binding(s)</td>
</tr>
<tr>
<td>bsadmin ldapupdate</td>
<td>synchronize BSCW users with LDAP</td>
</tr>
<tr>
<td>bsadmin ldif</td>
<td>export users to LDIF format</td>
</tr>
<tr>
<td>bsadmin level</td>
<td>manage level of proficiency</td>
</tr>
<tr>
<td>bsadmin license</td>
<td>request a new licence, check licence details or warn about licence expiry</td>
</tr>
<tr>
<td>bsadmin listmeta</td>
<td>export metadata as CSV list</td>
</tr>
<tr>
<td>bsadmin listmetakeys</td>
<td>list standard meta elements</td>
</tr>
<tr>
<td>bsadmin listws</td>
<td>list (shared) workspaces, update workspace list</td>
</tr>
<tr>
<td>bsadmin ls</td>
<td>list documents given by file path</td>
</tr>
<tr>
<td>bsadmin lstevents</td>
<td>list events</td>
</tr>
<tr>
<td>bsadmin mailaccess</td>
<td>list all folders w/ enabled mail access</td>
</tr>
</tbody>
</table>

(continues on next page)
bsadmin mailaslink list all documents w/ mail access token
bsadmin manage_servers manage BSCW servers machine-wide
bsadmin members add or remove users from workspaces
bsadmin mkfolder creates folders
bsadmin namespaces List obsolete namespace objects
bsadmin oauth list oauth consumers
bsadmin openid list openid
bsadmin package (un)install a BSCW package
bsadmin preview generate Document preview documents
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 renameaddr rename mail addresses using regular expressions
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 (3)
bsadmin sendmail BSCW mailer (D)
bsadmin servaccess BSCW access control service debugging (D)
bsadmin servuno BSCW user notification service debugging (D)
bsadmin service manage Windows NT platform BSCW service (3)
bsadmin start start BSCW instance servers
bsadmin statistics statistics from BSCW database
bsadmin stop stop BSCW instance servers
bsadmin syncf synchronizes BSCW folder with file system directories
bsadmin sysmsg modify user sys_msg counter
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 versions list/remove versions from document version stores
bsadmin virusfound list document scan information
bsadmin wstat print workspace statistic

(1) only on Windows 10, Server 2016/2019
(2) only on POSIX systems
(3) 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) 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:

• a 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 Administration using the Web Interface or section 7.2 Administration using the bsadmin script), 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 address 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:

```bash
$ ./bin/bsadmin users
Usage:
bsadmin users -{m|a|p|n} [-T|-E|-I] {<-o|-O} <ndays> [-L<f>] [<u1> ... <un>]
bsadmin users [-h]

List users and mail addresses

positional arguments:
-m print username(s) and primary mailaddress
-a print username(s) and all mailaddresses
-p print username(s) and passwords (htpasswd format)
```

(continues on next page)
of all or given users <ui> ... <un>;
-n print username(s)
sub-options:
-1 append creation, last-access time stamps
-E append account-expiry, passwd-expiry, passwd-change time stamps
('=' marks user individual account/password expiry date)
-I append last ip address

optional arguments:
-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
-h show this help message and exit

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 register
Usage:
bsadmin register <addr> show info about email address
bsadmin register -a [-o<n>] print email addresses
bsadmin register -b [-f<f>] [-o<n>] [-d] print bounced email addresses
bsadmin register -b [-m] <addr> set bounced
bsadmin register -c <addr> [<lang>] create pending email address
bsadmin register -d <addr> delete email address
bsadmin register -e [-o<n>] [-d] print external email addresses
bsadmin register -e <addr> [<lang>] set external
bsadmin register -i <addr> print user/<unknown>/<pending>
bsadmin register -n <addr> <newaddr> rename (change email 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> [<pwd> [<lang>]] register new user
bsadmin register -u [-o<n>] print allocated email addresses
bsadmin register -u <addr> <user> [<lang>] allocate secondary email address
bsadmin register -U <addr> <user> [<lang>] allocate primary email address
bsadmin register [-h] show this help message and exit

Registration of email addresses (and new users)

Optional arguments:
-f<f> consider email addresses with flags <f> ::= {n|f|a}+
   n - address w/o invitation
   f - address w/ invitation to a folder (workspace)
   a - address w/ invitation to an appointment
   (if option '-f<f>' is omitted, the default '-fnfa' is assumed,
    i.e. all <pending> email addresses are listed)
-m do not send email notification to user, who invited the address
-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 and provides the following options:

```
$ ./bin/bsadmin rename -h
usage: bsadmin rename [-h] [-n] oldname newname

rename a user

positional arguments:
oldname     existing name
newname     name to change to

optional arguments:
-h, --help  show this help message and exit
-n          don't 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] [-a | <user>]
bsadmin chpwd [-v] [-E <date>|-p] [-a | <user>]
bsadmin chpwd [-v] [-r] [-e|-p] [-a | <user>]
bsadmin chpwd [-h]

Change user password and lock/unlock user

positional arguments:
-user       username
-user [pwd]  set new password for user

optional arguments:
--l         lock user
--u         unlock user
-e          expire user (lock account if user is expired)
-notify     send email notification BCC to email address
-notify     send email notification to locked user only
-a          all users
-E <date/period> expire user at the time of 'date' or at the end of 'period'
            ('yyyy-mm-dd[ hh:mm]' or '3d' or '5w')
-notify     reset password (-p) / account (-e) expiry
-p          expire password (force password renewal at next login)
-v          verbose (shows account details if no other option is given)
-h          Show this help message and exit
```

Note: User accounts with administrative rights do not expire.

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 <owner>] [--dry-run] <user>
```

(continues on next page)
bsadmin rmuser [-n|b|v] -m <nowner> [-d <user>]
bsadmin rmuser [-n|b|v] [-f] <user>
bsadmin rmuser [-h]

Remove a user

options:
- n do not send an email notification
- b set user email address(es) invalid ("bounced")
- m <nowner> merge workspaces to <nowner>.
Without option -b also merge email addresses.
- 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
- h show this help message and exit
- 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 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
  Require all denied
  # some dedicated hosts (fqdn) or IP addresses may access
  Require host bscw.server.org
  Require ip 10.23.45.67
  # ...
</Location>
```

To automatically generate this configuration within your <bscw-runtime-path>/conf/apache24/bscw.conf file you have to create a <bscw-runtime-path>/conf/apache24/u_<username>.txt file which contains 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/apache24/u_intranet.txt` and enter the following Require directives:

```plaintext
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:

```python
SCRITPS = {
    '/sec/': (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/apache24/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`

```python
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`, see `HTTP_LOCAL_PORT`). 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 3.4.1 *Apache HTTP Server Configuration* (unix) or section 4.5.2 *Apache HTTP Server Configuration* (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 `<bscw-runtime-path>/conf/config.py`.

- **WSREPORT = 1** enables the daily workspace report

- **WSREPORT_DIRECT = 1** enables the direct email notification

- **AUTOSUBSCRIBE_REPORT** and **AUTOSUBSCRIBE_REPORT_DIRECT** define the default subscription for all users. 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
  AUTOSUBSCRIBE_REPORT_DIRECT = 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 with exception of read events are subscribed (create + move + change = 14).

- **DEFAULT_EVENTMASK_DIRECT** defines the default event type subscription mask for the direct email notification. By default no event types are preselected, so user won’t be notified about any events immediately, but may select some event types for certain folders of interest only.

- **DEFAULT_EVENT_REPORT_DAILY** defines the default frequency for periodic email report which may either be daily (1) or weekly (0)

  For example, set `WSREPORT_DIRECT = 1`, `AUTOSUBSCRIBE_REPORT_DIRECT = 1` and `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.)

- **WSREPORT_EVENT_LIMIT** defines a size limit of the periodic workspace report: in order to prevent too long notification emails the number of events can be limited (use `WSREPORT_EVENT_LIMIT = 0` for unlimited size)

- **REPORTLOG** points to a file in which a protocol about the reports is logged, for example:

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

### 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-7.3.2-<rev>-py3?/etc/bin/expire.sh` to the `<bscw-runtime-path>/bin` directory, e.g.
$ cd /home/bscw/lib/bscw-7.3.2-<rev>-py3?
$ cp ./etc/bin(expire.sh <bscw-runtime-path>/bin
$ cd <bscw-runtime-path>
$ chmod 700 bin/expire.sh

• create a “crontab” entry as follows:

```bash
$ crontab -e
10 1 * * * <bscw-runtime-path>/bin/expire.sh >> <bscw-runtime-path>/var/log/expired.log
```

• to notify the user about account expiration enable the package `expire` with:

```bash
$ bin/bsadmin package -e expire
```

**Note:** 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 BSCW status page) the `bsadmin du` script (disk usage) is available:

```bash
$ bin/bsadmin du -h
usage: bsadmin du [-h] [-v] [-u]
show/update BSCW database disk usage

optional arguments:
  -h, --help show this help message and exit
  -v verbose
  -u update (re-calculate) BSCW database disk usage
```

To periodically re-calculate the storage usage configure the following “crontab” entry:

```bash
$ 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. Open the action [Admin] in the user menu in the right upper corner of the interface and login a second time with your password to gain administration rights. The administrator status is indicated by an additional [Admin] menu in the right upper corner of the interface.

2. Enter the Communities folder of the anonymous user by opening the URL:

   https://bscw.domain.org/sec/bscw.cgi:/anonymous

3. Delete the public folder as shown:
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 create a new folder with the name public in the Communities folder. In particular follow this procedure:

1. Open the action [Admin] in the user menu in the right upper corner of the interface and login a second time with your password to gain administration rights. The administrator status is indicated by an additional [Admin] menu in the right upper corner of the interface.

2. Enter the Communities folder of the anonymous user by opening the URL:

https://bscw.domain.org/sec/bscw.cgi/:anonymous

3. Open the [Assign role] form in the members menu and assign the manager role to your account by the selection of [x] Manager. Afterwards click [OK]:

4. Create a new folder with [(+) New → Folder]. Enter the name public and click [OK].

5. Open the [Assign role] form in the members menu again and click [Remove specific role assignments] and afterwards [OK]

6. Finally run from the command line

$ cd <bscw-runtime-path>
$ ./bin/bsadmin fix_anonymous
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. 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/apache24/` files (cf. section 3.4.1 Apache HTTP Server Configuration (unix) / section 4.5.2 Apache HTTP Server Configuration (windows)).

7.6.1 Microsoft Support for WebDAV

More recent Microsoft Windows and MS Office versions (Office 2010, 2013, 2016) 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
     ```plaintext
     https://bscw.domain.org/sec/bscw.cgi/9620
     https://bscw.domain.org/sec/bscw.cgi/home
     ```
   - 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.
7.6.2 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:
Section 10.1.7 *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

<table>
<thead>
<tr>
<th>Command Pattern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bsadmin quota limit</code></td>
<td>defines and lists all limit classes;</td>
</tr>
<tr>
<td>`bsadmin quota { on</td>
<td>off }`</td>
</tr>
<tr>
<td><code>--users;</code></td>
<td></td>
</tr>
<tr>
<td>`bsadmin quota { check</td>
<td>fix }`</td>
</tr>
<tr>
<td><code>--users;</code></td>
<td></td>
</tr>
<tr>
<td>`bsadmin quota { report</td>
<td>class }`</td>
</tr>
</tbody>
</table>

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 }
bsadmin quota [-h]

User disk quota commands

options:
check|fix       checks/fixes current disks and objects usage for all users
class         report users for all specified classes
report         report quota for all or specified users
report -a     usage accumulation of specified users
report -l     only quotas exceeding soft limits are shown
report -L     only quotas exceeding limits are shown
on            (re-)enable quota for all or specified users
on -c <c>     set and enable quota class <c> and for all or specified users
on -R         reset quota timer for all or specified users
off           disable quota for all or specified 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
-h            show this help message and exit

With the following option parameters:

| <u1> ... <un> | string | list of registered BSCW user names
| <c>           | string | limit class name
| <soft>        | integer | limit value in bytes or in kilo (mega, giga, tera) bytes with unit token 'K' ('M', 'G', 'T').
| <hard>        | integer | limit value in seconds or in minutes (hours, days, weeks) with time token 'm' ('h', 'd', 'w').
| <time>        | integer | 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:

```bash
bsadmin quota limit [-v]
bsadmin quota limit <c>  { disk | objects }  <soft> <hard> <time>
bsadmin quota limit -d <c>  { disk | objects }
```
Listing of Limit Classes

The command `bsadmin quota limit` prints a list of all defined limit classes.

```
$ ./bin/bsadmin quota limit

<table>
<thead>
<tr>
<th>Disk Objects</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>soft</td>
<td>hard</td>
<td>time</td>
<td>soft</td>
<td>hard</td>
</tr>
<tr>
<td>default</td>
<td>0</td>
<td>0</td>
<td>0s</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>develop</td>
<td>40M</td>
<td>80M</td>
<td>2w</td>
<td>400</td>
<td>800</td>
</tr>
</tbody>
</table>
```

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` (or `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 bob and claire and assign them to the develop limit class:

$$ ./bin/bsadmin quota on -c develop bob claire $$

• Change quota limit class for user alice to class default:

$$ ./bin/bsadmin quota on -c default alice $$

**Disable Quota**

Quota may be disabled for all or individual user(s) with the `bsadmin quota off` command. Examples:

• Disable quota for user dave

$$ ./bin/bsadmin quota off dave $$

• Reset the quota limit timer for soft quotas

$$ ./bin/bsadmin quota off -R dave $$

• 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).

### 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.

- 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:

$$ ./bin/bsadmin quota report $$

<table>
<thead>
<tr>
<th>User</th>
<th>Disk usage</th>
<th>soft</th>
<th>hard</th>
<th>time</th>
<th>Objects usage</th>
<th>soft</th>
<th>hard</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>alice</td>
<td>11.1M</td>
<td>10M</td>
<td>15M</td>
<td>3.3d</td>
<td>150</td>
<td>200</td>
<td>300</td>
<td></td>
</tr>
</tbody>
</table>

(continues on next page)
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 claire alice

<table>
<thead>
<tr>
<th>User</th>
<th>Disk usage</th>
<th>Disk soft</th>
<th>Disk hard</th>
<th>Disk time</th>
<th>Objects usage</th>
<th>Objects soft</th>
<th>Objects hard</th>
<th>Objects time</th>
</tr>
</thead>
<tbody>
<tr>
<td>alice</td>
<td>11.1M</td>
<td>10M</td>
<td>15M</td>
<td>3.3d</td>
<td>150</td>
<td>200</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>claire</td>
<td>12M</td>
<td>40M</td>
<td>80M</td>
<td></td>
<td>94</td>
<td>400</td>
<td>800</td>
<td></td>
</tr>
</tbody>
</table>

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, 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, Waste do not inherit 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 users’ home). 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, 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 <bscw-runtime-path>/conf/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 artifacts, 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 the following different views (all views have language dependent names defined in <bscw-pkg-dir>/bscw-7.3.2-<rev>-py3/?/bscw/msg/<lang>/lg_msgconfig.py):

<table>
<thead>
<tr>
<th>action view</th>
<th>value</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>view('get')</td>
<td>1</td>
<td>actions involve 'read' access to an object, e.g. the get operation itself, copy, or convert</td>
</tr>
<tr>
<td>view('get_ext')</td>
<td>2</td>
<td>actions involve 'read' access to meta data (description, info page), e.g. info.</td>
</tr>
<tr>
<td>view('add')</td>
<td>4</td>
<td>actions create a new object</td>
</tr>
<tr>
<td>view('add_ext')</td>
<td>8</td>
<td>actions create a new object</td>
</tr>
<tr>
<td>view('change')</td>
<td>16</td>
<td>actions involve 'write' access to an object, e.g. actions that modify an object</td>
</tr>
<tr>
<td>view('change_ext')</td>
<td>32</td>
<td>actions move an object, i.e. change both the source and the target container</td>
</tr>
</tbody>
</table>

(continues on next page)
view('owner') 64 actions exclusively for the owner of an object, i.e. the destroy action.
view('share') 128 actions affect the access rights of an artifact (excluding role management), e.g. adding a member.
view('share_ext') 256 actions for role management, e.g. assigning a role.
view('edit') 512 actions for editing articles, attachments or appointments.
view('user') 1024 actions concern user information, e.g. editing user details, sending e-mail.
view('waste') 2048 actions possible in the waste, e.g. destroy and undelete.
view('lock') 4096 actions for lock/unlock objects.
view('attend') 8192 actions allowed for attendees of an appointment.
view('creator') 16384 actions for the creator of an artifact, e.g. edit and cut actions.
view('responsible') 32768 actions for the responsible of a task.
view('specialtags') 65536 actions to tag roles/artifacts.

A view comprises all actions that have this view assigned. The definition of a new view is done with the view() function.

Next 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:

```python
standard_views = (view_get | view_get_ext | view_add | view_add_ext | view_change | view_change_ext | view_waste)
complete_views = (standard_views | view_user | view_share | view_edit)
default_roles = {
    'R0creator': view_edit | view_creator,
    'R0hidden': 0,
    'R0other': 0,
    'R0owner': view_owner,
    'R1anonymous': view_get,
    'R1restricted': view_get | view_get_ext,
    'R2lockbag': view_lock,
    'R2associate': standard_views,
    'R2attendee': 0,
    'R2manager': complete_views | view_share_ext,
    'R2member': standard_views | view_user | view_share,
    'R2responsible': complete_views | view_responsible,
    'R2waste': view_waste,
}
```

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.

```text
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 (appointment)
```

(continues on next page)
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, 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, 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 file (`<bscw-runtime-path>/conf/config_action.py`). The action configuration file allows to adapt BSCW kernel (core) actions and roles:

```plaintext
<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 adapt the BSCW system roles setting use the

```plaintext
$ ./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.

In the following, we give an example for extending BSCW system defined roles (as described above) by adapting the action configuration file `<bscw-runtime-path>/extensions/customroles/conf/config_actions.py` in an new package `customroles`.

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 5.28 `msg/<lang>/lg_msgconfig.py`

```plaintext
<bscw-runtime-path>/extensions/customroles/msg/de/lg_msgconfig.py
<bscw-runtime-path>/extensions/customroles/msg/en/lg_msgconfig.py
```

Here are the file contents:

```plaintext
###########################################################################
# File
# <bscw-runtime-path>/extensions/customroles/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>/extensions/customroles/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'] = \n   default_roles['R2dommanager'] = \n   default_roles['R2fldmanager'] = \n   default_roles['R2eduadvisor'] = \n   default_roles['R2member']
```

(continues on next page)
# Note: Obviously it makes some sense to define different Roles that have the same actions allowed *initially*

###########################################################################
# File
# `<bscw-runtime-path>/extensions/customroles/msg/en/lg_msgconfig.py`
# User friendly names for new roles defined in
# `<bscw-runtime-path>/extensions/customroles/conf/config_action.py`

R2learner = 'Learner'
R2author = 'Author'
R2dommanager = 'Domain manager'
R2fldmanager = 'Field manager'
R2eduadvisor = 'Educational advisor'

###########################################################################
# File
# `<bscw-runtime-path>/extensions/customroles/msg/de/lg_msgconfig.py`
# User friendly names for new roles defined in
# `<bscw-runtime-path>/extensions/customroles/conf/config_action.py`

R2learner = 'Lerner'
R2author = 'Autor'
R2dommanager = 'Domänenmanager'
R2fldmanager = 'Branchenmanager'
R2eduadvisor = 'Aus- und Weiterbildungsberater'

After defining the custom roles the package `customroles` must be enabled with:

```
bin/bsadmin package -e customroles
```

### 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>/var/www/local/icons` where to store your (customized) resource icons:

   ```
   $ cd <bscw-runtime-path>
   $ mkdir -p ./var/www/local/icons
   $ chmod 755 ./var/www/local ./var/www/local/icons
   ```

2. Copy a index logo or banner logo PNG image of your organisation called `logo_index.png` resp. `logo_banner.png` into the extensions resources icon directory (`<bscw-runtime-path>/var/www/local/icons`):

   ```
   $ cd <bscw-runtime-path>
   $ cp logo_index.png logo_banner.png ./var/www/local/icons
   $ chmod 644 ./var/www/local/icons/icons/logo_index.png
   $ chmod 644 ./var/www/local/icons/icons/logo_banner.png
   ```

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):
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 BSCW appearance settings).

6. Run `bsadmin index_page` to generate new BSCW index pages

### 7.10 Server-wide template folders

By using the action 

```
[(+) 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 users’ Personal Templates folder

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.

   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-7.3.2-<rev>-py3?/etc/doc_templates/` for some basic template documents

To create a system-wide template folder:

1. Open the action `[Admin]` in the user menu in the right upper corner of the interface and login a second time with your password to gain administration rights. The administrator status is indicated by an additional `[Admin]` menu in the right upper corner of the interface.

2. Enter the Personal Templates folder of the anonymous user by entering the URL:

```
https://bscw.domain.org/sec/bscw.cgi/ranonymous
```

**Note:** The leading lowercase “r” is a shortcut to address the template folder of anonymous user.

3. Open the [Assign role] form in the members menu and assign the manager role to your account by the selection of `[x] Manager`. Afterwards click `[OK]`.

---

7.10. Server-wide template folders 175
4. Create a new template folder with [(+) New → Template Folder]. Enter a name and click [OK].

5. Open the [Assign role] form in the members menu again and click [Remove specific role assignments] and afterwards [OK]

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 `ACCEPT_WEBSERVICES` variable to 1 (default) in the BSCW instance configuration file `<bscw-runtime-path>/conf/config.py`. If `ACCEPT_WEBSERVICES` set to 0, web service protocol requests will be rejected by BSCW with the HTTP error code 501: content_unsupported.

Please note that BSCW is distributed with API documentation in HTML format and some API examples as Python scripts:

- the API documentation in HTML format is located in
  `<bscw-path>/lib/bscw-7.3.2-<rev>-py3?/doc/devel/BSCW|relmaj|-API.zip`
- the API examples as Python scripts are located in
  `<bscw-path>/lib/bscw-7.3.2-<rev>-py3?/etc/src-aux/remote_client`

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/sec/bscw.cgi/<shortname><username>

or:

https://bscw.domain.org/sec/bscw.cgi/<shortname><emailaddress>

where <shortname> is a single character of the following list:

<shortname> ::= { @ #addrBook | _ #waste | ` #case | $ #lockbag | + #calendar | * #bookmarks | r #resources | : #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 "alice" and the email address "alice@orbiteam.de" the URL:

- https://bscw.domain.org/sec/bscw.cgi/malice@orbiteam.de 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/sec/bscw.cgi/ualice returns the info page for user alice 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/sec/bscw.cgi/@alice shows the users’ address book;
- https://bscw.domain.org/sec/bscw.cgi/_alice shows the users’ waste basket;
- https://bscw.domain.org/sec/bscw.cgi/`alice shows the users’ briefcase;
- https://bscw.domain.org/sec/bscw.cgi/$alice shows the locks that the user currently holds on documents;
- https://bscw.domain.org/sec/bscw.cgi/\alice shows the users’ clipboard;
- https://bscw.domain.org/sec/bscw.cgi/+alice shows the users’ calendar;
- https://bscw.domain.org/sec/bscw.cgi/*alice shows the users’ bookmarks;
- https://bscw.domain.org/sec/bscw.cgi/ralice shows the users’ personal template folder;
- https://bscw.domain.org/sec/bscw.cgi/:alice shows the users’ home page;
- https://bscw.domain.org/sec/bscw.cgi/=alice shows the users’ portal;
- https://bscw.domain.org/sec/bscw.cgi/~alice shows the users’ task list.

7.12. Some useful hints
The BSCW help is available for from your BSCW server instance:

- https://<server>/pub/static/help/english/
- https://<server>/pub/static/help/german/

Alternatively you may access the help on our web page:

- https://www.bscw.de/social/help/ (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 https://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 https://www.bscw.de/en/social/#languages 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].

#### 8.1.2 Translation instructions

You can add support for new languages by creating a sub directory in your BSCW instance `<bscw-runtime-path>/conf/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>/conf/msg/*` the distributed translations are located in the `<bscw-path>/lib/bscw-7.3.2-<rev>/bscw/msg/*` directories. The distributed directories `bscw/msg/en/*` contain all relevant language dependent strings 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/templates 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 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>/conf/msg/<your-language-two-letter-code> and copy each relevant file from <bscw-path>/lib/bscw-7.3.2-<rev>~py3?/bscw/msg/en/* into this directory. Next 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/<your-language>/<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!

**Note:**

• 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 (' / ') 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 (e.g. AIR Widgets, Java Applets) are not included in the above mentioned files – contact OrbiTeam support@orbiteam.de to translate these 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 (https://www.bscw.de/en/social/#download). Before upgrading to a new version please see section 2.4 Upgrading to BSCW 7.3.2.
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.

### 9.1 License application

A BSCW administrator may commence a license upgrade process by clicking the “Upgrade licence” link, which is provided in the administrator interface of the BSCW server:

- 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 press [OK]. Now apply for a new license with:

  ```
  [Options > Admin > Upgrade license] [OK]
  ```

- The next [OK] action will connect you to the license server configured in the variable BSCW_LICENSE (see <bscw-runtime-path>/conf/config.py). Fill in/update the form (be sure to enter a valid email-address!) and choose your desired license type.

Generally when choosing a license type 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 licensor. After reception of the signed license agreement, licensor will decide if licensee qualifies for a royalty free license. As a rule, licensor 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:

- As soon your license is granted you will receive an email notification. Follow the mentioned URL, resp. open:

```
[Options > Admin > Upgrade Licence]
[OK]
```

and accept the licence agreement with:

```
[I accept licence agreement]
```

- Finally perform a garbage collection and restart the BSCW database server to install the new 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 https://bscw.domain.org with the script path /sec/bscw.cgi using HTTPS looks like:

```
org.domain.bscw:443S.sec
```

### 9.2 License changes

The BSCW license will become invalid whenever the `SERVER_ROOT` or the secure prefix path within the `SCRIPTS` dictionary is changed! This applies for example when the `SERVER_ROOT` is changed from HTTP to HTTPS.

To change your license without service interruption proceed as follows:

1. Change your `SERVER_ROOT` variable and apply for a “Change licence for new server (royalty free)” license (see License application above). Please print and sign the shown license agreement and fax or send document (scanned) by email to us.

   **Note:** This change has no impact on the running BSCW database server, since the new `SERVER_ROOT` is only (re)loaded after a BSCW database server restart.

2. Change your `SERVER_ROOT` variable back to the original (valid) server root definition and wait until your license is granted (you will receive an email notification).

3. After you received the email notification change the `SERVER_ROOT` again to the new definition.

4. To activate the new license run a garbage collection (or wait for the nightly automatic garbage collection run) and restart your BSCW database server after the garbage collection has been completed.
CHAPTER
TEN

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, calendars 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 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

10.1.5 How do I change my password?

You may change your password using the user menu [Change Password] item in the right upper corner of the interface.

Keywords: password, change password

10.1.6 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_doc_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).

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

10.1.7 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 GNOME, 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).

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 10.

**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 10

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 `/sec/bscw.cgi` resp. `/bscw/bscw.cgi`) in the “Internet or network address field”.

For example, enter `https://bscw.domain.org/sec/bscw.cgi` (resp. `https://bscw.domain.org/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 10 you have to enable the “WebClient service” by setting the service Startup type to “Automatic”.

### 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.domain.org/sec/bscw.cgi` (resp. `https://bscw.domain.org/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. If the password dialog pops up again and contains a hostname in front of your username (e.g. “serversmith”), 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).

4. 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.domain.org\sec\bscw.cgi` (resp. `\\bscw.domain.org\bscw\bscw.cgi`)

5. Make sure you have installed all recent updates and service packs.

6. Make sure your BSCW server is running the most recent version of the BSCW software. If in doubt ask your BSCW administrator for help.
Hints for Windows 10

On Windows 10 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` in the Programs list.
2. Locate and then click the following registry key:
   
   ```plaintext
   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 your computer.

**Note:** For security purposes, Windows disables 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](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](http://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:
### 10.1.8 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

### 10.1.9 How do I delete my account?

Only if enabled, it is possible to delete your own user account with [Destroy Account] in the user menu in the right upper corner of the interface. 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 [Preferences] [Notifications] [Active Event Services]. 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

### 10.1.10 How do I handle a BSCW error?

If you encounter a BSCW error message System error 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 URL of the BSCW server you use,
- the time the error occurred and the complete error message you get,
- 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:** BSCW error, bug report
10.1.11 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. To empty the garbage

- please enter your [Trash]
- press [Clear waste]

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 https://public.bscw.de/.

Note:

- Disk space limiting is set per user.
- Disk space accounting concerns all workspaces you are the owner of.
- You can control your quota the following way:
  - Open the chat view your user by clicking on your username and open the info page by clicking the white (i) overlay in the users’ icon
  - On the info-page you’ll find all necessary information about used disk space and quota.
  - You may check your quota limit and the amount of currently used space.

Keywords: quota, disk space limit

10.1.12 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, 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 MS-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

10.1.13 Is there a restriction for the size of documents I upload?

No, there is 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.
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 https://www.bscw.de/en/social/#download

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 for the BSCW social or BSCW classic product provided by OrbiTeam and test the software online - or download the software and test it on your own server.

Please note that for the on-line trial all data you upload will be deleted after 90 days.

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 https://lists.bscw.de/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 or Facebook for more instant updates.

Finally you may want to check our website BSCW frequently to check for news and updates.

Keywords: new releases, update, developments, announcement mailing list

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 [Admin] in the user menu in the right upper corner of the interface, 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 [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 `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 *Administration using the bsadmin script*

**Keywords:** admin tools, 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 [Admin]. Find the respective user and select [Permanently 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 [Admin]. Find the respective user and select [Change Name].

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] action 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, etc.) along the folder hierarchy:

   - First open the chat view of the user:

   ```
   https://<your server>/sec/bscw.cgi/u<username>
   ```
and then open the users’ info page by clicking the white (i) overlay in the users’ icon.

- Edit the (default) user role “Manager” and select the actions you want to restrict/allow using [Edit role] from the Administrator actions.

- Alternatively you may assign a more restrictive role to the user with [Assign role] from the Administrator actions.

2. If you want to generally disallow users to create workspaces it is advisable to define a server-wide more restrictive user role, see section 7.8.2 Role definition and default roles 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 [Invite Member] action in the member menu (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.docx`.

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.docx
```

**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 beginning with a `@` in “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 2.4 Upgrading to BSCW 7.3.2. 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-7.3.2-<rev>-py3?.tar.gz
     
     ```
     # tar xzf bscw-7.3.2-<rev>-py3?.tar.gz
     ```
   
   - Enter the distribution directory bscw-7.3.2-<rev>-py3? and perform the usual installation steps (see Installation 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-7.3.2-<rev>-py3?.tar.gz
     # cd bscw-7.3.2-<rev>-py3?
     # ./install.sh
     ```
   
   Enter BSCW system user name: [bscw]
   Enter BSCW base directory: [/home/bscw]

   Extracting BSCW 7.3.2 distribution in /home/bscw/lib

   Choose one of the following options:
   ( 0) update BSCW 5.2.3 [/home/bscw/srv/bscw.domain.org]
   ( 1) create new BSCW instance
   Enter a number (0-2): 0

   - Adopt your Apache HTTP server settings (see section 3.4.1 Apache HTTP Server Configuration);
   - Edit the BSCW main server configuration file <bscw-runtime-path>/conf/config.py and adapt it to your needs, e.g. enable new features (be sure to configure the mandatory settings section (see section 3.4.2 BSCW instance configuration)).

3. Windows 10, Server 2016/2019:
   - Download and execute the BSCW distribution installer bscw-7.3.2-<rev>-py3?.exe
   - Choose the BSCW instance you want to upgrade and follow the configuration dialog (see section 4.2 Installation and Configuration)
   - Adopt your HTTP server settings if you are using Apache HTTP server (see section 4.5.2 Apache HTTP Server Configuration);
   - Edit the BSCW main server configuration file <bscw-runtime-path>/conf/config.py and adapt 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 [Admin] in the users’ menu in the right upper corner of the interface.
   
   Log in a second time with your password to gain BSCW administrator rights for the current session and apply with

   ```
   [Admin > Upgrade license]
   [OK]
   ```
   
   - Fill in the form (be sure to enter a valid email-address!)
     - Choose the license type:
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`:
     - adapt 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 `SERVER_ROOT`)
   - 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 your 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 [Admin] in the users’ menu in the right upper corner of the interface.

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

```
[Admin > Upgrade license]
[OK]
```

Fill in the form (be sure to enter a valid email-address!) and choose the license type *Change license for new server (royalty free)*. Please print and sign the shown license agreement and fax or send document (scanned) by email to us.

4. As soon your license is granted you will receive an email notification:
   - follow the mentioned URL
   - perform a garbage collection and
   - restart the BSCW database server.

**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 Changing the “SERVER_ROOT” without service interruption

The BSCW license will become invalid whenever the `SERVER_ROOT` or the secure prefix path within the `SCRIPTS` dictionary is changed! This applies for example when the `SERVER_ROOT` is changed from HTTP to HTTPS.

To change your license without service interruption see the BSCW Admin Manual 7.3 (http://www.bscw.de/files/Download/AdminManual72.pdf) chapter 9 *BSCW license*.

Keywords: BSCW-license, license change

### 10.3.13 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 in `<bscw-runtime-path>/var/log` the log files `bscw.log` and `sys.log`):

1. The BSCW server reports the a *System error* to a client and the `<bscw-runtime-path>/var/log/sys.log` file contains a traceback like:

```
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:
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 `<bscw-runtime-path>/var/data/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@orbitteam.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\Tables
   > copy var\data\Backup var\data\Store
   > bin\bsadmin start
   ```

If your backup is outdated, or the backup 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`
```
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

### 10.3.14 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 license.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 https://bscw.orbiteam.de.

2. Open the previously stored "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, scan and send 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 license.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 granted.html).

5. The last step again needs connection to your BSCW server (the one behind the firewall):

6. Open the stored license (URL file:granted.html) on a system with access to your BSCW server and select [Upload license].

**Keywords:** Upgrade license, connect problems, firewall

### 10.3.15 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:** Backup your database storage files!

   ```
   $ bin/bsadmin getconfig STORE
   <bscw-runtime-path>/var/data/StoreA
   # active database store
   $ mkdir var/data/backups
   $ cp var/data/Backup var/data/backups
   $ cp var/data/StoreA var/data/backups
   $ cp var/data/StoreB var/data/backups
   ```

   **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):

```
$ bin/bsadmin dbcheck list
$ bin/bsadmin dbcheck repair
```

• 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

### 10.3.16 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 2.4 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 2.4 for Unix and Windows [http://httpd.apache.org]). You also require the interpreter and standard libraries for the Python 3 programming language. The Python implementation is copyrighted, but is freely usable and can be downloaded from [http://www.python.org].

See also:
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`. Usually the BSCW software is installed in the home directory of the BSCW system user at `/home/bscw` or at `/opt/bscw` (Unix) or at `C:\BSCW` (Windows).

It is necessary that your Web server have access to the file system where BSCW is installed. For best performance use a local file system of the host where your Web server runs.

**Keywords:** BSCW server installation, operating system

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

```plaintext
https://<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 - 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/python3`) and the Python libraries (usually `/usr/lib/python3.?/`) 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

- In `<bscw-runtime-path>/conf/config.py` change (or add a line):
  ```python
  GSMOD_CAN_FLUSH = 0
  ```

- Stop the running database server:
  ```bash
  $ bin/start_servers -k
  ```

- Check if stopping the database server was successful:
  ```bash
  $ ps -ef | grep bsadmin
  ```

- There should be no process `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 a BSCW user interface in different languages?

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-7.3.2-<rev>-py3?/bscw/msg (e.g. msg/(de|en|es|fr) come with the server distribution).

To add support for additional languages see section 8.1.2 *Translation instructions* of this manual. Please note that up to date information on available languages can be found at https://www.bscw.de/en/social/#languages.

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 "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

```python
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 *Installation* of this manual for correct BSCW user and group setup.

Execute as BSCW user `bscw` with the group `bscw` the `bsadmin chkconfig` script:

```python
# 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)

### 10.4. BSCW Installation
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). On Debian Linux ensure the following ownership/permissions

```bash
$ su -
$ id
uid=0(root) gid=0(root)
# cd <bscw-runtime-path>
# chown www-data ./var/data/expired_users
# chmod 660 ./var/data/expired_users
# chown www-data ./var/data/htpasswd
# chmod 660 ./var/data/htpasswd
# chown www-data ./var/data/registered_users
# chmod 660 ./var/data/registered_users
# chown www-data ./var/data/removed_users
# chmod 660 ./var/data/removed_users
# chown -Rh www-data ./var/cache/preview
# find ./var/cache/preview -type d | xargs chmod 2770
# chown -Rh www-data ./var/data/Temp
# find ./var/data/Temp -type d | xargs chmod 2770
# chown -Rh www-data ./var/log
# chmod 2770 ./var/log
```

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 webserver user.

```bash
$ su -
$ id
(continues on next page)
```
BSCW Administrator Documentation, Release 7.3.2

(continued from previous page)

uid=0(root) gid=0(root)

# chown -Rh www-data ./var/data/Files
# ./bin/start_servers

Keywords: Permission denied, HTTP server, OSError, Unix

10.4.8 Why do I get a "RuntimeError: var/www/bscw.cgi: No setgid"?

If your operating system does not support set-group-id scripts (such as Linux) a binary wrapper program is used to allow set-group-id operation of the bscw.cgi script. If your operating system supports set-group-id scripts, 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-xr-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 10.4.8 Why do I get a "Permission denied" error? (Unix).

Keywords: Python traceback, RuntimeError, CGI scripts, operating system

© 1995-2020 Fraunhofer FIT and OrbiTeam Software GmbH & Co. KG
a
airdesktop, 127
approval, 127

b
blog, 128


c
case, 128
chat, 129

e
expire, 129
exportpdf, 129

f
flow-folder, 130

h
http, 130

i
incognito, 132

l
ldap, 124

m
metaprofiles, 133

o
office, 133

p
PyLucIndex, 117

s
sync, 140

t
tasks, 141

w
WebFolder, 141
INDEX

A
airdesktop (module), 127
approval (module), 127

B
blog (module), 128

C
case (module), 128
chat (module), 129

E
expire (module), 129
exportpdf (module), 129

F
flow-folder (module), 130

H
http (module), 130

I
incognito (module), 132

L
ldap (module), 124

M
metaproses (module), 133

O
office (module), 133

P
preview Unix, 25
preview Windows, 54
PyLucIndex (module), 117

S
sync (module), 140

T
tasks (module), 141

W
WebFolder (module), 141