

BAR – Backup ARchiver

Version 0.21

© 2007-2021 Torsten Rupp

Table of Contents

Overview.....	5
Features.....	5
Requirements.....	5
Supported entry types.....	6
Installation.....	7
Packages.....	7
Network connection key files.....	7
Usage.....	8
Creating archives.....	9
File archives.....	9
Image archives.....	10
List archives and directory content.....	10
Test archives.....	11
Compare archives.....	11
Extract archives.....	12
Convert archives.....	12
Incremental/Differential archives.....	13
Compress archives.....	14
Encrypt archives.....	15
Asymmetric encryption.....	16
Split archives.....	16
Store archives.....	17
File system.....	17
Remote server (FTP/SCP/SFTP/WebDAV/WebDAVS).....	17
CD/DVD/BD.....	18
Device.....	19
Server.....	20
Daemon.....	20
Ports.....	20
Firewall.....	20
Master and Slave.....	21
Database.....	22
bar-sqlite3.....	22
Graphical front end.....	24
Status tab.....	25
Jobs tab.....	26
Entries.....	26
Images.....	27
Filters & Mounts.....	28
Storage.....	29
Scripts.....	30
Schedule.....	31
Persistence.....	32
Comment.....	33
Restore tab.....	33
Server settings.....	36
General.....	36
Servers.....	37
Commands.....	38

Verbosity & Log.....	38
Use cases.....	39
a) Normal backup.....	39
b) Full daily backup with time schedule.....	40
c) Full weekly backup with time schedule and daily incremental backup.....	40
d) Full weekly backup with daily incremental and continuous backup.....	40
e) Restore single file or directory.....	41
f) Restore full backup.....	41
Appendix.....	42
Tools.....	42
Archive file name macros.....	42
BAR command line options summary.....	43
Basic commands.....	43
Include/Exclude files.....	45
Incremental/Differential archives.....	46
Compression.....	47
Encryption.....	50
Signatures.....	52
Splitting.....	52
Storage.....	53
File.....	53
FTP.....	53
SSH.....	53
SCP.....	54
SFTP.....	54
WebDAV/WebDAVS.....	54
CD/DVD/BD.....	55
Device.....	57
Restore.....	58
Server.....	59
Index database.....	60
Continuous database.....	61
Logging.....	62
Miscellaneous.....	63
BAR configuration and job files.....	68
Additional configuration entries.....	68
Block configuration entries.....	69
File server.....	69
FTP server.....	70
SSH/SCP/SFTP server.....	70
WebDAV/WebDAVS server.....	70
Device.....	71
Schedule.....	71
Persistence.....	72
Maintenance.....	72
Example configuration file.....	72
BARControl command line options summary.....	73
BARControl configuration file.....	75
Exit codes.....	76
Archive file format.....	77

Compilation.....	77
Development.....	81
Frequently asked questions.....	81
Contact and bug reports.....	89
License.....	89
References.....	90

Illustration Index

Figure 1: Graphical front end main window (status tab).....	25
Figure 2: Entries.....	27
Figure 3: Images.....	28
Figure 4: Filters and mounts.....	29
Figure 5: Storage.....	29
Figure 6: Archive name editor.....	30
Figure 7: Scripts.....	31
Figure 8: Scheduler.....	32
Figure 9: Persistence.....	32
Figure 10: Restoring files.....	34
Figure 11: Restore dialog.....	35
Figure 12: Server general settings.....	36
Figure 13: Servers settings.....	37
Figure 14: Server command settings.....	38
Figure 15: Server verbosity & log settings.....	39

Index of Tables

Table 1: Supported entries.....	6
Table 2: Certificates and key files.....	8
Table 3: Archive part and medium sizes.....	19
Table 4: Server ports.....	20
Table 5: Archive file name macros.....	43
Table 6: Pattern types.....	45
Table 7: Compression algorithms.....	50
Table 8: Encryption algorithms.....	51
Table 9: Log information types.....	62
Table 10: Log format macros.....	63
Table 11: Verbosity levels.....	67
Table 12: BAR exit codes.....	76
Table 13: Compilation packages.....	79
Table 14: Tests.....	80

Overview

BAR is backup archiver program. It can create compressed, encrypted and split archives of files and disk images which can be stored on a hard disk, CD, DVD, BD or directly on a server via FTP, SCP, SFTP, or WebDAV. *BAR* can create full and incremental/differential archives as well as continuously store created or changed files into archives. A server-mode and a scheduler is integrated for making automated backups in the background.

Features

- create, list, test, compare and extract archives of files and disk images,
- convert archives,
- fast access to entries in archives: can find and extract single files without decompressing or decryption of the whole archive,
- store extended file attributes (EAs),
- full and incremental/differential backup of files into archives,
- continuous storage of new created or changed files,
- support for raw and several file systems for disk images (ext, fat, reiserfs),
- can split archives into parts of selectable size, each part can be read independently,
- compress archives with zip, bzip2, lzma, lzo, lz4, Zstandard, and/or xdelta algorithms,
- encrypt archive content with gcrypt algorithms (BLOWFISH, TWOFISH, AES, a. o.),
- asymmetric encryption of content with RSA keys,
- store archives into file system and directly on external servers via FTP, SCP, SFTP, WebDAV, or WebDAVS
- with external tools: store archives on CD/DVD/BD including error correction codes, blanking medium,
- support storage on a generic device,
- restore archives from file system, external server, CD/DVD/BD and generic device,
- multi-core support for compression and encryption,
- server mode with included scheduler for doing backups regularly. Controlling the server can be done via network connection (plain & TLS/SSL),
- optional internal database with information to all stored files and disk images,
- graphical front end for server to check server status, create jobs, start jobs and stop jobs.

Requirements

To execute *BAR* you need:

- glibc 2.3.2 or higher
- *BAR* binary
- JRE 1.6 or newer [JRE, JDK] (for graphical font end only)

Optional external tools for creating CDs/DVDs/BDs with *BAR*:

- cdrecord [cdrtools]
- growisofs [growisofs]
- mkisofs, dvd+rw-format [dvd+rw-tools]
- dvdaster [dvdaster]

BAR is either available as a binary package for different systems or as source code for compilation. If you are not familiar with using a C compiler it is recommended to use one of the binary packages. Please check the website <http://www.kigen.de/projects/bar/index.html> for available binary packages.

To compile *BAR* you need:

- C/C++ development environment with gcc, g++
- GNU make
- m4
- xgettext
- perl
- tcsh
- sqlite3 3.24 or newer [SQLite]
- zlib library [zlib]
- icu library [ICU]
- JDK 1.6 [JRE, JDK]

Optional for compilation are:

- bzip2 library [bzip2]
- lzma library [LZMA]
- lzo library [LZO]
- lz4 library [LZ4]
- Zstandard library [Zstandard]
- xdelta library [xdelta]
- ssh2 library [libssh2]
- gcrypt library [libgcrypt]
- nettle library [nettle], gmp library [gmp], gnutls library [gnutls]
- libcurl 7.66 or newer [libcurl], c-ares [c-ares] and mxml [mxml]
- iconv library [libiconv] and cdio library 1.1 or newer [libcdio]
- pcre library [pcre]
- libbfd [binutils]
- ant, launch4j [ant, launch4j]
- epm [epm]

For more details to compile *BAR* by yourself please see chapter Compilation, page 77.

Supported entry types

BAR support the entry types listed in table 1: Supported entries.

Type	Comment
File	Regular file entries
Image	Disk block images
Directory	Regular directory entries
Link	Symbolic link entries
Hard link	Hard link entries
Special	Special files like character devices entries, block device entries, socket entries, fifo entries

Table 1: Supported entries

Installation

Packages

Installation of *BAR* can be done either from pre-compiled binary package files or from the sources. Pre-compiled binary packages are available from the web site

- <http://www.kigen.de/projects/bar/index.html>

as Debian or RPM package files for Linux distributions.

Note: the Debian and RPM package files are tested on some major Linux distributions, but they may be used on other Linux distributions, too which are not listed explicit when the package format is supported.

Installation of Debian packages:

```
sudo dpkg --install <BAR package file>.deb
```

Installation of RPM packages:

```
sudo rpm --install <BAR package file>.rpm
```

If there is no pre-compiled binary package available for your platform, you must compile *BAR* by yourself from the sources. The chapter Compilation (see page 77) give some instructions how to compile *BAR*.

Network connection key files

To use the SSL network connection to the *BAR* daemon a SSL key-pair must be generated first. The script *bar-keygen* can be used to create the public and private key files required for the front end. To generate a new key-pair call:

```
sudo bar-keygen
```

A key-pair is valid for one year. If the key-pair expires after one year a new key-pair must be generated with:

```
sudo bar-keygen --force
```

bar-keygen uses either the key generator tool *openssl* from the OpenSSL project [OpenSSL] or *certtool* from GNU TLS project [gnutls]. Thus to create the key files either OpenSSL or GNU TLS must be installed on the computer.

The generated and default installed key files in the appropriated directories are shown in table 2:Certificates and key files.

File	Description
/etc/ssl/cert/bar-ca.pem	<i>BAR</i> certificate
/etc/ssl/private/bar-key.pem	<i>BAR</i> private key
/etc/ssl/cert/bar-server-cert.pem	<i>BAR</i> server certificate
/etc/ssl/private/bar-server-key.pem	<i>BAR</i> server private key
/etc/bar/bar.jks	<i>BAR</i> Java front end key

Table 2: Certificates and key files

Note: keep the key files always in private!

Usage

BAR is a command line tool which can be called with several options to e. g. create, list, test, compare, convert and extract archives, include and exclude files, directories, links, hard links, special files and devices, enable compression and encryption and other things. The general syntax is:

bar [*<options>*] *<archive file name>* *<file>*...

BAR options are prefixed by either - (short name) or -- (long name) and may occur at any position. If an option is specified more than one time all values (for options which may occur multiple times) resp. the last value (for options which only can occur one time) are used.

The archive file name is the first non-option argument and specify the name of the archive to read or write. The archive file name can contain various macros which are expanded at run time (see Archive file name macros, page 42 for available macros).

The arguments following the archive name are names or patterns of entries which are stored, listed, tested or restored. The number of names or patterns is only limited by the max. command line length.

Note: For server mode see chapter Server, page 20. For the graphical front end see chapter Graphical front end, page 24.

The following chapters describe the different commands and the most used options. A full list of all options can be found in the appendix *BAR* command line options summary, page 43.

The following nomenclature is used in the following sections:

- *<name>* stay for a parameter, e. g. a name or some text,
- *<number>* stay for a number
- *<command|script>* stay for command or script
- [...] optional parameter,
- ... repeat the parameter one or more times,
- *<size>* specify a byte size. Sizes can be specified as numbers including the following optional units: P, T, G, M, K which multiply the given number by 1125899906842624 (1024⁵), 1099511627776 (1024⁴), 1073741824 (1024³), 1048576 (1024²), 1024 respectively.

Creating archives

BAR can create archives containing files, directories, links, hard links, special files, or block device images. Besides the file and image content data meta data like names, access rights, time stamps, user and group settings and extended attributes (EAs) are stored into the archive for each entry, too. Multiple entries to store into an archive can be specified on the command line. With the options *-#* or *--include* resp. *-!* or *--exclude* patterns can be specified to explicitly include or exclude entry or device patterns. By default include/exclude patterns can contain the meta characters *** and *?* to match multiple or single characters. With the option *--pattern-type* also regular expression or extended regular expression patterns can be enabled.

If a directory contain the file *.nobackup* or *.NOBACKUP* all files in this directory and all sub-directories are excluded. The *.nobackup/.NOBACKUP* file will be ignored (and included in the *BAR* archive, too, if not excluded otherwise) when the option *--ignore-no-backup-file* is specified.

If a file is marked with the file attribute “no dump” the file is excluded except when the option *--ignore-no-dump* is given.¹

A file, directory, link, hard link, special file or block device is included into an archive when:

- it is specified or match a pattern given with the option *--include* and
- is not specified and does not match a pattern given with *--exclude* and
- there does not exists the file *.nobackup* nor *.NOBACKUP* in the directory and the option *--ignore-no-backup-file* is not given and
- the file is not marked with the file attribute “no dump” and the option *--ignore-no-dump* is not given.

In all other cases the file is excluded from storage into an archive.

Note: If *** or *?* are used the pattern may be wrapped with *'...'* to avoid expansion of the pattern by the shell.

Note: When the option *--dry-run* is specified all operations are executed except compression, encryption and storage into archives, creating incremental file lists and update the index database. The option *--dry-run* can be used to do a dry-run to see what may be done.

File archives

To create an archive containing files, directories, links, hard links, or special files enter the command:

```
bar --create <archive file name> <file or directory>...  
bar -c <archive file name> <file or directory>...
```

Examples:

```
bar --create home.bar /home  
bar -c backup1.bar /home/foo/file1.txt /home/foo/file2.txt  
bar -c backup2.bar /home/foo/*.txt -# 'documents/*' -! '*/trash/*'  
bar -c backup3.bar /home --dry-run
```

Note: *BAR* try to preserve the file access time (atime) of the read files to the original value when creating archives.

¹ The no-dump attribute is not supported on all Linux versions and file systems.

Image archives

To create an archive containing the image of a disk or partition enter the command:

```
bar --image <archive file name> <device name>...  
bar -m <archive file name> <device name>...
```

Examples:

```
bar --image home.bar /dev/hda  
bar -m home.bar /dev/hda1
```

When you create an archive with disk images *BAR* tries to detect the used file system on the device. If the file system is known only blocks used by the file system are stored. Not used blocks are stored as “0”-blocks, thus when compression is used disk images become smaller. To disable detection of the file system and store all blocks (raw image) use the option *--raw-images*.

Currently supported file systems are:

- ext2
- ext3
- fat12
- fat16
- fat32
- ext4
- reiserFS (version 3.5/3.6)

Experimental supported file systems are:

- ext4

Experimental, but no longer supported files systems are:

- reiser4

List archives and directory content

To list the content of archives enter the command:

```
bar --list <bar archive file>...  
bar -l <bar archive file>...
```

or simply:

```
bar <bar archive file>...
```

Examples:

```
bar home.bar  
bar --list backup1.bar backup2.bar  
bar -L dvd://*.bar  
bar -H sftp://user@server  
bar -L -H webdav://user@server/backup3.bar
```

The content of the specified archives are listed in short list format with byte sizes. To get a long list format use the option *-L* or *--long-format*, to get human readable sizes use option *-H* or *--human-format*.

The directory content on a FTP, SFTP, WebDAV, or WebDAVS server or from CD/DVD/BD can be listed by specifying the URI and the directory or device name.

Examples:

```
bar /home
bar dvd://
bar sftp://user@server
bar webdav://user@server
```

Test archives

To test the integrity of an archive enter the following command:

```
bar --test <bar archive file>...
bar -t <bar archive file>...
```

Examples:

```
bar --test home.bar
bar -t backup1.bar backup2.bar
bar -t dvd://*.bar
bar -t webdav://user@server/*.bar
```

The specified archives are tested if all data can be read, decrypted and decompressed. If some data cannot be read, decrypted or decompressed an error is reported.

Compare archives

To compare the content of archives with the content in the file system, a device or a partition enter the following command:

```
bar --compare <bar archive file>...
bar -d <bar archive file>...
```

BAR compare the archive with the content in the file system, the device or the partition and report any entry or image which differs. The compare function can be used to check if an archive still contain the same entries or image data which are stored in the file system or device.

Examples:

```
bar --compare home.bar
bar -d backup1.bar backup2.bar
bar -d dvd://*.bar
bar -d webdav://user@server/*.bar
```

The content of the specified archives are compared with the existing file system entries or device image content. If some entry or image in an archive differ from the file system/image on the device a message is printed.

Extract archives

To extract (restore) the content of an archive enter the following command:

```
bar --extract <bar archive file>...  
bar -x <bar archive file>...
```

BAR extract the content of the archive and restore all files, directories, links, hard links, special files and device images which are stored in the specified archive files. Existing files, directories, links, hard links and special files will not be overwritten if not specified different by the option *--overwrite-files*. The destination of restored files, directories, links, hard links, special files, and device images can be modified with the option *--destination*.

Note: When the option *--dry-run* is specified all operations are executed except actually restoring archive entries. The *--dry-run* option can be used to do a dry-run to see what may be restored.

Examples:

```
bar --extract home.bar  
bar -x backup1.bar backup2.bar  
bar -x backup3.bar --dry-run
```

Convert archives

Archives can be converted to adapt one or all of the following parameters of existing archives:

- byte compress algorithm
- crypt algorithm
- crypt password
- crypt key
- comment
- signature

The archives are read and the content is re-encrypted and compressed with the new given settings. To set a new password or key see option *--crypt-new-password*.

Notes: If an archive is encrypted the old encryption password and old private key is required to change settings.

To change the signature of an existing archive the old public signature key is required or the option *--skip-verify-signatures* have to be used.

Examples:

```
bar --convert home.bar --compress-algorithm=lz4-l6  
bar --convert home.bar --crypt-algorithm=AES192 --crypt-password=secret  
bar --convert home.bar --crypt-algorithm=CAMELLIA256 --crypt-password=secret --crypt-  
new-password=new-secret  
bar --convert home.bar --comment='some text'  
bar --convert home.bar --signature-private-key=signature-key.private --signature-public-  
key=new-signature-key.public
```

Incremental/Differential archives

BAR can also create incremental and differential archives:

- Incremental archives: contain all changed files, directories, links, hard links and special files since the last full or incremental archive was created.
- Differential archives: contain all changed files, directories, links, hard links and special files since the last full archive was created.

Note: Incremental/Differential archives are not useful for storing device images.

To detect which files, directories, links, hard links, or special files are changed since the last incremental archive resp. the last full archive was created, an incremental data file is created when the first full archive is created resp. this file is updated when an incremental archive is created. With the information stored in this “incremental data file” it is detected which files, directories, links, hard links, or special files must be stored in an incremental or differential archive, because they were modified in the meantime.

The incremental data files have by default the file name extension *.bid*. The full name is derived from the created archive file name by the following pattern:

<path>/<archive name><date><number>.bar

will create an incremental data file with the name

<path>/<archive name>.bid

The derived default incremental data file name may be overwritten with the option *-I* or *--incremental-list-file*. The directory where the incremental data file is stored can be changed with the option *--incremental-data-directory*.

Note: Incremental data files are not encrypted, not compressed, and always stored on the local hard disk and are never transmitted to a remote server. The incremental data files are only used by *BAR* itself to create incremental or differential archives.

To create an incremental or differential archive first a full archive with all entries have to be created using the following command:

*bar --full --incremental-list-file=<incremental data file name> --create <archive file name>
<file or directory>...*

Note: Use the option *--no-storage* to create the incremental data file only.

Example:

bar --full --incremental-list-file=home.bid --create home.bar /home

This example command create the files *home.bar* and *home.bid*.

To create an incremental archive after some files, directories, links, hard links, or special files were modified use the command:

*bar --incremental --incremental-list-file=<incremental data file name> --create <archive file
name> <file or directory>...*
bar -i -I=<incremental data file name> --create <archive file name> <file or directory>...

To create a differential archive after some files, directories, links, hard links, or special files were modified use the command:

```
bar --differential --incremental-list-file=<incremental data file name> --create <archive file name> <file or directory>...
```

Examples:

```
bar --incremental --incremental-file-name=home.bid --create home-incremental.bar /home
bar --differential --incremental-file-name=home.bid --create home-differential.bar /home
```

These commands read in the incremental data file *home.bid* and store all modified files from */home* into the new archive *home-incremental.bar* resp. *home-differential.bar*. After the incremental archive is created the incremental data file *home.bid* is updated, too.

Compress archives

To save space in the created archive files it is highly recommended to compress the data. *BAR* offer several different compression algorithms of two different types:

- byte-compression and
- delta-compression.

Byte-compression encode data bytes in such a way that the result represent the same information, but usually require less space in the archive (isomorphic transformation). Delta-compression calculate a byte-difference of two data sets, e. g. the previous version of a file and the new version of a file and only store the difference information. Byte-compression can be used for any type of data, while delta-compression is recommended for pairs of data which only differ in a small number of data bytes like e. g. different versions of disk images or files of virtual machines. Delta-compression can also be used additionally to the byte-compression algorithms.

Note: Delta-compression is only useful for file, image or hard link archive entries and will be ignored for any other archive entry type.

If compression is enabled each entry in the archive is compressed separately. This result in a slightly bigger archive file than compressing the complete archive like e. g. *tar* do with the appropriated compression algorithm, but with single compressed entries *BAR* can list and search archive content faster, because it is not required to decompress the whole archive content before a single entry in the archive can be read. With separately compressed archive entries also not corrupted parts of a partially corrupted archive may still be readable.

To compress the archive content use the following option:

```
--compress-algorithm=<name>
--compress-algorithm=<name>+<name>
```

or

```
-z <name>
-z <name>+<name>
```

The supported compression algorithm names can be listed with the option *-h* or *--help*. All possible compression algorithm names are listed in table 7: Compression algorithms (see page 50).

When delta compression is enabled a delta source should be specified with the option *--delta-source=<archive|file>*. From the corresponding files in the existing archive and the files to store in the new archive only the difference (delta) is stored. If no delta source is found or no delta source is

specified with the option `--delta-source=<archive>` a warning is printed and delta compression is disabled for the archive entry. Delta compression can be combined with the byte compression algorithms by the separation character '+' to additionally compress the delta data.

Note: To force delta compression the option `--force-delta-compression` may be used. If a file, image or hard link archive entry cannot be delta compressed an error is reported.

The resulting total compression ratio depend on the used compression algorithms.

Examples:

```
bar --create home.bar /home --compress-algorithm=zip1
bar --create home.bar /home -z bzip5
bar --create home.bar /home -z lzma9
bar --create home.bar /home --compress-algorithm=xdelta9 --delta-source=home-previous.bar
bar --create home.bar /home -z=xdelta9+lzma9 --delta-source=home-previous.bar
```

Encrypt archives

BAR archive files can be encrypted. Encryption is done after compression and is highly recommended for archive files which are stored on external servers or CDs/DVDs/BDs to avoid non authorized access of the content.

To enable encryption use the following option:

```
--crypt-algorithm=<name>
```

or

```
-y <name>
```

When encryption is enabled, a pass phrase must be entered when the archive is created. The content of the archive may later only be accessed when the correct pass phrase is entered again. The pass phrase is read from the console if required² or can be stored in the *BAR* configuration file³.

Note: Because of the internal structure of the archive files, the general structure (how many entries, entry types) can be read without knowledge of the pass phrase. Nevertheless all data content including meta-data like file names are encrypted and can only be read with the correct pass phrase.

For encryption the crypt algorithms of libgcrypt [libgcrypt] are used. The supported crypt algorithm names can be listed with the option `-h` or `--help`. All possible crypt algorithm names are listed in table 8: Encryption algorithms (see page 51). The strength of encryption depend on the used algorithm and the used pass phrase. Currently e. g. AES256 may be a good choice for the encryption algorithm.

Do not weaken the encryption by weak pass phrases! If your pass phrase used for encryption is too weak and may be guessed easily then encryption become useless. Thus use a pass phrase of at least 8 characters and use not only characters, but digits and special characters, too. See also e. g. <http://en.wikipedia.org/wiki/Passphrase> how to select a good pass phrase.

2 If the environment variable `SSH_ASKPASS` is set to an executable this is called to enter the password.

3 If the pass phrase is stored in the configuration file, access to that file should be limited.

Note: There is no way to restore the archive content if you forgot the pass phrase!

Examples:

```
bar --create home.bar /home --crypt-algorithm=AES256 --crypt-password=secret
bar --create home.bar /home -z TWOFISH128
```

Asymmetric encryption

For asymmetric encryption the RSA algorithm⁴ with a public/private key-pair is used. A hybrid-encryption is used where the data is encrypted with the selected symmetric encryption algorithm (see option `--crypt-algorithm`) and a random generated password key. The random generated password key is RSA encrypted with the public key and stored into the archive. It can only be decrypted with the private key. For each new archive part a new random key is generated.

To use asymmetric RSA encryption first a public/private key-pair must be generated. The key-pair can be generated with *BAR* with the following command:

```
bar --generate-keys <key file name>
```

<key file name> is some file name where the created key-pair should be stored. The files *<key file name>.public* and *<key file name>.private* are created. The public key is used to encrypt a random data encryption key before storing the encrypted key into the archive. The private key file is required to decrypt the random generated encryption key and with that the data of the archive. Thus the public key must be stored on the computer where *BAR* is used to create archives while the private key must be available on the computer where archive files should be read or extracted.

Example:

```
bar --generate-keys key
```

This command ask for a pass phrase for the private key and then create a pair of files with the names *key.public* and *key.private* which contain a RSA key-pair.

Notes: Because the RSA encryption private key is a critical piece of data, always keep this key at a secret place!

The option `--generate-keys` always generate a new key-pair. There is no way to regenerate a lost key-pair. Thus take care to safely store the key files!

Split archives

Complete backup archives can grow to big files depending on the content to store and the used compression algorithm. *BAR* offer the possibility to split archives into parts of a specific size. Parts can usually be stored more easily e. g. on a server or on CD/DVD/BD. Each created part can be read independently. Even when a single part is missing or cannot be read anymore because it became destroyed by some reason, e. g. a read error on CD/DVD/BD, other parts can still be read by *BAR* and restored. Thus not the whole archive content is lost. Parts which cannot be read are restored as "0"-bytes in the resulting file or image.

To split an archive into pieces of a specific size use the following option:

```
--archive-part-size=<size>
```

⁴ RSA is a cryptographic algorithm of Ron Rivest, Adi Shamir and Leonard Adleman which uses separated key data for encryption and decryption (public-key-encryption). See <http://en.wikipedia.org/wiki/RSA> for more details.

or

-s <size>

Notes: The size of a single part may become a little bit larger than the specified size. Parts may oversized by max. 16kiB without compression. The reason is that data is written in 16kiB blocks to get a reasonable speed when archives are created.

When compression is used a single part may grow a few bytes more. The reason for this is that the compression algorithms usually use an internal buffer for some data which may be flushed and appended to the archive part when the part size reaches the specified size.

When splitting is enabled the archive file name should contain one or more #. A consecutive sequence of # is replaced by *BAR* by a unique number for each created archive part. If no # is found in the specified archive name, *BAR* append .##### to the archive name which result in an appended 6-digit part number.

Store archives

Created archives can be stored to several destinations by specifying an URI (Uniform Resource Identifier) at the beginning of an archive file name. The default is storing into the local file system when no URI is given. *BAR* can store archives directly to a remote server or with some external tools also on CD/DVD/BD or a generic device.

To create an archive which should be stored at a specific destination use the following command:

bar --create <uri><archive file name> <file or directory>...

<uri> can be one of the URIs described in the following sections.

File system

To store the archive into the file system of the local computer use the following URI:

file://<file name>

or simply

<file name>

Examples:

bar --create file:///backup/home.bar /home

bar --create /backup/home.bar /home

Remote server (FTP/SCP/SFTP/WebDAV/WebDAVS)

BAR can store archives directly on a server either via the FTP protocol when libcurl [libcurl] and [c-ares] is compiled-in or via the SSH/SCP/SFTP/WebDAV/WebDAVS protocols when the libssh2 [libssh2], libcurl [libcurl], libcares [c-ares], and mxml [mxml] are compiled-in.⁵

To store the archive directly on a remote server use the following URIs:

ftp://[<login name>[:<login password>]@]<server name>/<file name>

scp://[<login name>[:<login password>]@]<server name>[:<port>]/<file name>

⁵ The pre-compiled packages have the libraries libcurl, c-ares, mxml, and libssh2 compiled-in.

```
sftp://[<login name>[:<login password>]@]<server name>[:<port>]/<file name>
webdav://[<login name>[:<login password>]@]<server name>/<file name>
webdavs://[<login name>[:<login password>]@]<server name>/<file name>
```

The archive (part) is first created on the local hard disk in a temporary file and is then transmitted to the remote server. Thus the local hard disk should have enough free space to hold at least one archive part. See option *--max-tmp-size* to limit the space *BAR* may use for temporary files. Transmission of an archive part to a remote server is done with full speed of the network connection up link. To limited the used bandwidth use the option *--max-band-with*.

Note: A @ in a login name can be escaped by prefixing it with a backslash (\).

Examples:

```
bar --create ftp:///backup/home.bar /home
bar --create scp://foo@myserver/backup/home.bar /home
bar --create sftp://foo:mypassword@myserver/backup/system.bar /boot /etc /usr
bar --create webdav://foo:mypassword@myserver/backup/system.bar /boot /etc /usr
```

CD/DVD/BD

When the external tools *cdrecord* [cdrtools], *growisofs* [growisofs], *mkisofs*, *dvd+rw-format* [dvd+rw-tools], and optionally *dvdisaster* [dvdisaster] are installed, *BAR* can also create CDs, DVDs and BDs with optionally added error correction codes and blank a medium before writing. To store the archives on CD/DVD/BD use the following URIs:

```
cd://[device name:]<file name>
dvd://[device name:]<file name>
bd://[device name:]<file name>
```

Note: When the libiconv library [libiconv] and libcdio library [libcdio] is compiled-in *BAR* can read archive files written on CD/DVD/BD directly without mounting the CD/DVD/BD.⁶ Otherwise the CD/DVD/BD must be mounted first, then the archive files can be read as normal files.

To add error correction codes in the space not used on the CD/DVD/BD the external tool *dvdisaster* can be used to add Reed Solomon error correction codes⁷. With the error correction codes *dvdisaster* may be able to recover read errors on a CD/DVD/BD. *BAR* will call *dvdisaster* after creating the CD/DVD/BD image to add the error correction codes to the image when you specify the following option:

```
--ecc
```

If a CD/DVD/BD medium is used which may already contain some old data which should be overwritten it may be required first to blank the medium before it can be written. Blanking can be done with the option:

```
--blank
```

⁶ The pre-compiled packages have this library compiled-in.

⁷ http://en.wikipedia.org/wiki/Reed%E2%80%93Solomon_error_correction

When a CD/DVD/BD is created it is recommended to enable archive splitting (see Split archives, page 16). Otherwise a single archive file is created which may not fit on the CD/DVD/BD medium⁸. When additionally error correction codes are enabled, the archive part size should have a suitable setting depending on the medium size. Suitable settings are all combinations of archive part sizes and medium sizes which leave some not used space on the CD/DVD/BD medium which can be used by *dvdisaster* to store error correction codes. Table 3: Archive part and medium sizes give some useful settings for the archive part size and the medium size.

Medium type	Archive part size	Medium size
CD	140M	560M
CD	560M	560M
DVD, single layer	620M	3.6G
DVD, double layer	1.2G	7.2G
BD	2G	20G

Table 3: Archive part and medium sizes

Note: To create a CD/DVD/BD the archive files are first created in a temporary directory before they are written to the CD/DVD/BD. Thus the local hard disk should have enough free disk space to store the archive files and additionally the size of the CD/DVD/BD image. In general around 1.5GB for a CD, 9GB for a DVD (single layer), and up to 50G for a BD (single layer) of temporary disk space may be required. *BAR* will output a warning when the free disk space in the temporary directory may not be sufficient to create a CD/DVD/BD.

Examples:

```
bar --create cd://home.bar /home
bar --ecc --blank --create dvd:///dev/dvd:home.bar /home
```

Device

BAR can store archives also on some generic device. This could be a tape system or some other type of storage device which can be accessed via some device name.

To store the archive on a device use the following URI:

```
device://[<device name>:]<file name>
```

Example:

```
bar --create device:///device/backup/home.bar /home
```

With the options to execute a command before an archive is created or an entry is stored additional commands can be used to control the device. See options *--device-write-pre-command*, *--device-write-post-command*, *--device-write-command*, *--pre-command*, *--post-command*, *--file-write-pre-command*, *--file-write-post-command*.

⁸ Without using ISO9660 level 3 extension the size of a single file on a DVD is limited to 4GB. However it is recommended not to create files large than 2GB, because some non-standard DVD reading software may not be able to read files which are larger than 2GB.

Server

Daemon

BAR can be started as a server (daemon) which is running in the background and execute backup or restore jobs. To control the server you can connect to a running *BAR* server instance on some computer – including on a remote system⁹ – with a network connection. The network connection can either be a plain TCP/IP socket connection or it can be a protected and encrypted TLS (SSL)-socket connection (recommended).

To start *BAR* in server mode enter the following command:

```
bar --daemon
```

This start *BAR* and detach the process to run in the background. To prevent detaching the process use the option *--no-detach* (may be useful for debugging purposes).

Ports

BAR server use the default ports for accepting network connections from the graphical front end as listed in table 4: Server ports. The default port settings can be overwritten with the options shown in the table in the last column.

Port	Type	Option to overwrite default value
38523	plain connection or TLS (SSL) connection	<i>--server-port=<n></i>
38524	TLS (SSL) connection	<i>--tls-server-port=<n></i>

Table 4: Server ports

A SSL encrypted connection may also be accepted on the plain server port. Whenever possible a TLS connection is established.

Note: Specify 0 for the port number to disable the appropriated server.

Firewall

Please make sure the specified server ports can be reached from the network if there is a firewall program running. If required add some firewall rules, e. g. for *iptables* something like

```
iptables -I INPUT 4 -p tcp --dport 38523 -j ACCEPT
```

```
iptables -I INPUT 4 -p tcp --dport 38524 -j ACCEPT
```

The rule number (4 in this example) may be different. The correct number can be detected with the *iptables* command:

```
iptables -list
```

⁹ Accessing *BAR* on a remove system may require to open the *BAR* server control ports in routers and firewall settings. As an alternative a SSH tunnel may be used. See the FAQ, page 81 for this.

The new rules should be inserted before the general *iptables DROP* command.

Master and Slave

Several instances of a *BAR* server can run on different computers. A single instance can be declared as the “master”, all other instances become “slaves” and are controlled by the master server. With this structure all backup jobs can be defined on the master server. Jobs which should run on a slave are transferred to the slave on the fly and executed there. The resulting archives are optionally transported back to the master and stored to the final destination. To specify a job for a slave, set the host name and port number in the job setting accordingly to the host where the *BAR* server slave is running. The port number can be 0 for the default port number.

Before a job can be executed on the slave the slave and master have to be paired once. Pairing is done with of the following steps:

- define at least one job on the master for the slave which should be paired
- start the *BAR* server slave instance and
 - wait until pairing is done if no pairing was done ever before (automatic pairing) or
 - start *BARControl*, connect to the *BAR* server slave instance and select *master* in the menu or
 - run the command
barcontrol -pair-master <slave host name>
- or
- create an empty file with the name specified with the option *--pairing-master-file* (see Miscellaneous, page 63) on the computer where the *BAR* server slave instance is running.

The *BAR* server slave switch automatically to pairing mode at startup if there is still no master paired. The *BAR* server master has a time window of 10min to pair with the slave.

Pairing is done with the following procedure:

- The slave is in pairing mode. This is either done automatically at startup of the slave if there is still no master paired or by an explicit request to slave.
- The master check periodically every minute if a slave can be reached via the network. If the slave cannot be reached it is marked “offline”, if the slave can be reached, but it is still not paired the slave is marked “wait pairing”.
- If a slave is reachable the master login in to the slave and transmit a master pairing code.
- Slave accept the pairing code, store it and switch off pairing mode.

If this procedure is successfully done the *BAR* server master and slave are paired.

Database

When *BAR* is started as a server (daemon) and the optional internal database is configured (see configuration entries *index-database-...* in */etc/bar/bar.cfg*) *BAR* create for all stored entries and disk images an entry in an internal index database¹⁰. Also stored archives located in the file system and on external servers are scanned and an index is created and updated automatically. With the internal database searching for entries and disk images and restoring is much faster and it is possible to search for entries without having direct access to the storage files, e. g. stored on a remote server.

Archives in the database are organized in entities. An entity is an instance of a job run either by the scheduler or when executed manually. An entity may contain several storage files which are themselves contain all the entry meta information like names and sizes of the store entries.

Note: For special cases creating an index for archives can be disabled with the command line option *--no-index-database*.

Archives which are not already in the internal database are periodically checked and added automatically if

- the archive files are stored either in the file system, on a FTP server, SCP/SFTP server, or a WebDAV/WebDAVS server and
- the name suffix is *.bar*; and
- the archives can be decrypted which means the running *BAR* server has the required pass phrase stored in the internal pass phrase cache and the required private key file.

BAR checks periodically all entries in the index database. If an archive cannot be found anymore the index entry is deleted from the database. The time to keep an index entry can be specified with the option *--index-database-keep-time*. To limit the network band width used to update indices the option *--index-database-max-band-width* can be used. To disable automatic index creating and updates, specify the option *--index-database-no-auto-update*.

Note: The internal database file need some disk space. If the database is not required or a huge number of files are stored the database feature may be disabled by commenting out the configuration entry *index-database* in */etc/bar/bar.cfg*.

Storage files can be added, removed, or refreshed manually in the index database with the command line options *--index-database-add* *--index-database-remove* and *--index-database-refresh* of *BARControl* or via the *BARControl* front end in the tab restore in the context menu of the archive list.

Note: The index database is upgraded automatically when a new *BAR* version is installed and is run for the first time. Depending on the size of the database this may take some time. The upgrade is done in the background. The shown data in *BARControl* in the restore tab may be incomplete while an upgrade of the index database is running in the background.

Note: The index operations run much faster if the index database file is located on a storage medium with low random access latency, e. g. a SSD.

bar-sqlite3

Usually the index database is maintained and updated from *BAR* automatically. In case the database may be corrupt or invalid *BAR* try to create a new database and import the data from the old database file. For manual maintenance the tool *bar-sqlite3* can be used for some basic operations.

10 The sqlite SQL database engine is used [SQLite].

- create a new empty database: this is done with option *--create*.
Note: be careful with this option! An existing database is deleted when a new one is created
- *--create-triggers*: re-create all triggers of the database. It deletes all internal trigger functions and re-create them. The operation does not lose data in the index. This option is rarely used and may only be useful after an upgrade of BAR when the database internal structures changed and some trigger functions have to be updated.
- *--create-newest*: re-create all newest data in the database. It deletes all existing newest data in the database and re-creates it from the existing entries data.
- *--create-indizes*: re-create all indizes of the database. It deletes all internal indizes in the database and re-create them. The operation does not lose data in the index, but depending on the size of the database it takes some time to complete. This operation may solve problems when the index is incomplete or show wrong sizes or numbers of entries.
- *--create-aggregates*: re-create aggregate data in the database like storage sizes, number of entries in a entity. The operation does not lose data in the index, but depending on the size of the database it takes some time to complete. This operation may solve problems when the index is incomplete or show wrong sizes or numbers of entities.
- *--check*: check integrity of the database and report errors if something is corrupted.
- *--clean*: remove duplicated and orphaned entries. The operation does not lose data in the index, but depending on the size of the database it takes some time to complete.
- *--vacuum*: collect and remove not used space in the index database file. This may shrink the disk space used by the index database file. The operation does not lose data in the index, but depending on the size of the database it takes some time to complete.
- *--info*, *--info-entities*, *--info-storages*, *--info-entries*: print some information of the database content.
- execute some SQL command: execute an arbitrary SQL command. SQL commands may also be piped into *bar-sqlite3* with - as the SQL command or the option *--pipe*.

For more details to the available commands and options see *--help* of *bar-sqlite3*.

Graphical front end

The graphical front end *BARControl* for *BAR* can be used to control a *BAR* server instance (daemon) running on the local or some remote computer. Via the front end the status of a server can be displayed, backup or restore jobs can be created, edited, deleted, started, paused, suspended or canceled.

Note: To execute the graphical front end the Java Run time Environment 1.6 [JRE, JDK] must be installed. The front end use SWT [SWT]. The required SWT libraries for Linux (32- and 64bit), Solaris (32bit), Windows (32- and 64bit), and OS X (32- and 64bit) are included in *BARControl*.

The graphical front end is started with:

```
barcontrol [<server>]
```

resp. with:

```
barcontrol.bat [<server>]
```

If no server name is given or a connecting to the default server is not possible a login dialog is shown where a server name may be entered or selected from a known list as well as the server port number, SSL setting, the login password and the user role. With the option *--login-dialog* the login dialog may be forced to be shown.

To establish a TLS/SSL encrypted connection to the server the private key file *bar-key.pem* is required. The key files can be generated with the *BAR* tool *bar-keygen* (see chapter Network connection key files, page 7). The private key is read from one of the following locations:

- specified on the command line via the option *-key-file*
- *bar-key.pem* in the local directory
- *\$HOME/.bar/bar-key.pem*
- */etc/bar/bar-key.pem*
- */etc/ssl/bar-key.pem*

Note: If the connection fail because the password is wrong further connection tries from the same host may be delayed by the server to resist brute-force password attacks.

The main window look like in example figure 1: Graphical front end main window (status tab).

Notes: red numbers like ❶ in the images are only markers for this documentation and are not part of the graphical user interface.

Most input fields and buttons have a short online-help as tool tips. The tool tips are shown when you keep the mouse pointer over some field or button for a short time.

To hide the shell window which is opened on Windows, create a Windows link and select in the properties settings of the Windows link the option “minimized”.

The front end try to connect to the specified server or the local computer if no sever name is specified either via a TLS/SSL encrypted connection or via a plain connection.

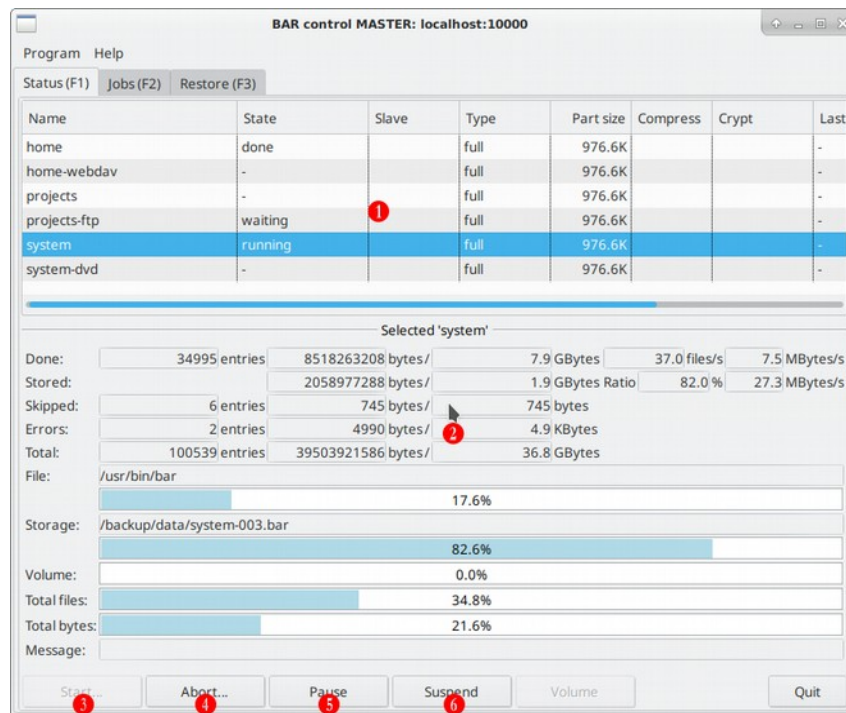


Figure 1: Graphical front end main window (status tab)

BARControl also have some useful command line options. See appendix *BARControl* command line options summary, page 73.

Status tab

In the status tab (see example in figure 1: Graphical front end main window (status tab)) the list in the upper part ① show the defined backup jobs and their current status. The status show if currently a job is executed, is waiting to be executed, execution is paused or suspended and in case of slave jobs if the slaves are online or offline.

The lower part ② show the detailed status of the currently select job in the list. Via the button ruler at the bottom a backup job can be started, aborted, paused or suspended. A job may also be started started, aborted, paused or suspended via the context menu which can be opened with the second mouse button.

After selecting a job in the list which is not already scheduled to run or is running, the job can be started with the *Start* button ③. In the appearing dialog you can select the specific operation:

- normal: create an archive with all entries as specified in the job. Do not read or create any incremental data files.
- full: create an archive with all entries as specified in the job. Create an incremental data file.
- Incremental: create an incremental archive with all entries as specified in the job which changed since the last incremental or full archive was created. Update the incremental data file.
- Differential: create an differential archive with all entries as specified in the job which changed since the last full archive was created. Do not update the incremental data file.
- Dry-run: do a dry-run of all operations except compression, encryption and storing data.

With the *Abort* button ④ the currently selected running job can be aborted, with the *Pause* and *Suspend* buttons ⑤ job execution can be suspended for some time or until the *Continue* button ⑤ is pressed.

When data is written to a CD, DVD, BD or a device and a new volume is required, a message appear that a new volume should be inserted. With the button *Volume* ⑥ *BAR* is informed that a new volume is loaded and can be used to store the next part of the data.

The message line also show error messages in case some job execution fail due to some error.

Note: The complete long text of the field “Message:” can be displayed in a tool tip box when the mouse is positioned over the text.

Jobs tab

In the jobs tab backup jobs can be created, copied, edited, renamed and deleted (see example figure 2). Each job have a unique name and is stored into a text file by the *BAR* server (default directory is */etc/bar/jobs*). A job file contain the following data:

- slave host to run the job ①
- included entry names
- included images
- filters and mounts
- storage information
- scripts
- schedule information
- persistence information
- comments

The jobs tab have some sub-tabs to edit these pieces of data.

Entries

With the entries tab (see example figure 2: Entries) file entries and directories can be included and excluded into/from a backup job. A double-click on a directory resp. the arrow symbol¹¹ ② open or close the directory view. With the buttons ③ at the bottom files, directories, links, hard links and special files can be included (entry become green) ④ or excluded (entry has a red cross) ⑤. Entries which are not either explicitly included or excluded will become included or excluded in the backup when the parent entry is included or excluded.

Note: Via the context menu the selected entry can be included or excluded in different ways: by its name, by setting the file attribute *NODUMP* or by creating the file *.nobackup*.

With the directory symbol ⑥ at the bottom all included directories are opened in the directory tree view. With the directory size checkbox ⑦ the directories sizes (sum of files in the directory and all included sub-directories) are calculated and shown.

Note: It is also possible via the context menu to explicitly ask for the size of the selected directory (sum of files in the directory and all included sub-directories).

¹¹ The symbol may different on different operating systems. It is part of SWT framework resp. the used window manager.

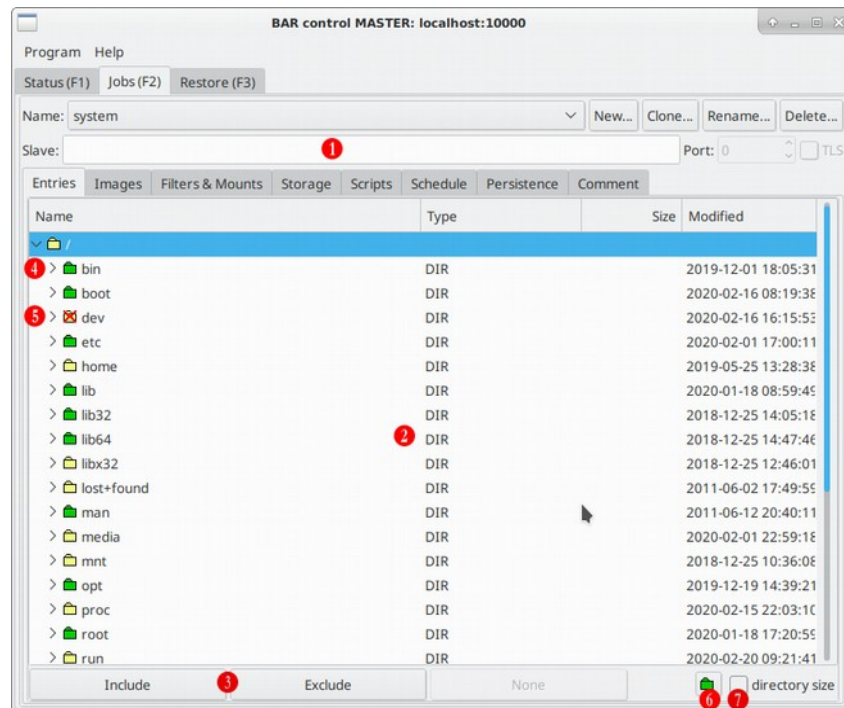


Figure 2: Entries

Images

Similar to file entries and directories in the files tab, in the images tab (see example figure 3: Images) devices can be selected to be stored into an archive as block device images.

Note: The block device should not be mounted while creating an image backup.

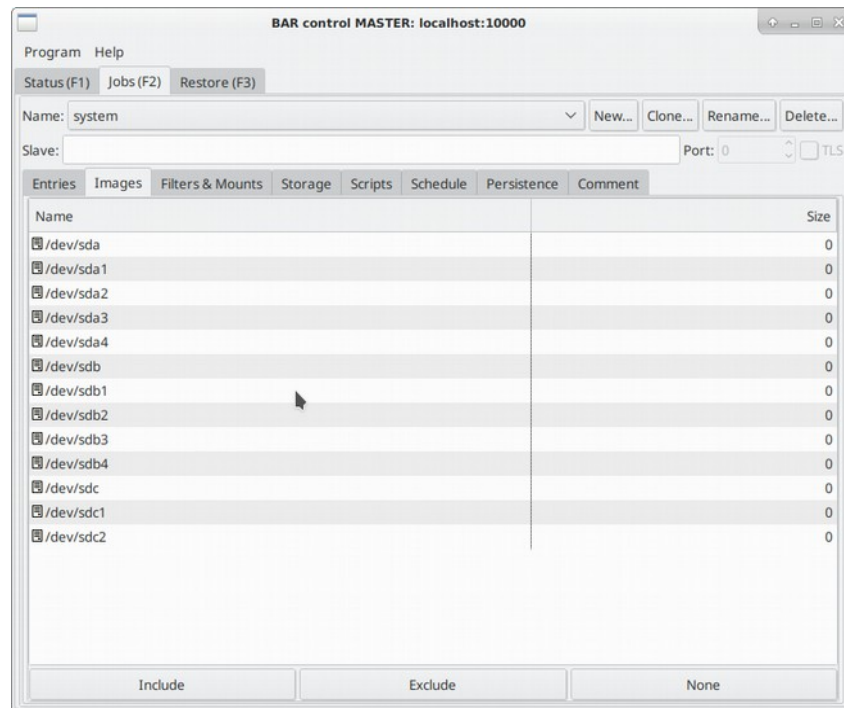


Figure 3: Images

Filters & Mounts

In the filters and mounts tab (see example figure 4: Filters and mounts) general include and exclude filters can be specified. Files, directories, links, hard links, special files, and device matching to an include filter ❶ and do not match to an exclude filter ❷ are included into the created archive. Entries which are explicitly excluded will never be stored into an archive.

Besides listing included and excluded entries also script commands can be specified which are executed to create a list of included ❸ or excluded entries ❹. The scripts should output in each line a single entry name on standard output.

To automatically mount and unmount file systems mounts ❺ may be defined. The mount name must match an entry in the system mount table¹² or a device name must be specified. When the job is executed all mounts are mounted if not already mounted and unmounted after the job is executed if the mount was executed before.

With the skip unreadable entries option ❻ not readable entries are skipped and no error is reported for those entries. With no stop on attribute error option ❼ errors when reading a entry attribute are ignored. With the raw image option ❽ the raw data of a disk image is stored for an image operation.

12 On Linux this is usually `/etc/fstab`.

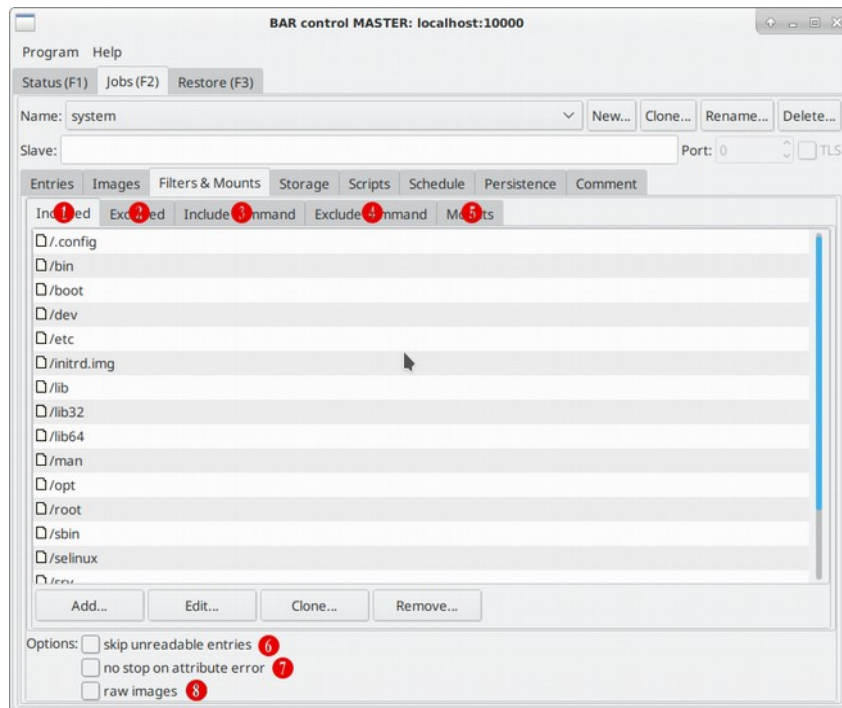


Figure 4: Filters and mounts

Storage

With the storage tab (see example figure 5: Storage) different settings for storing created archive files can be edited. In this tab the archive part size, compression, encryption and the storage name and location can be set.

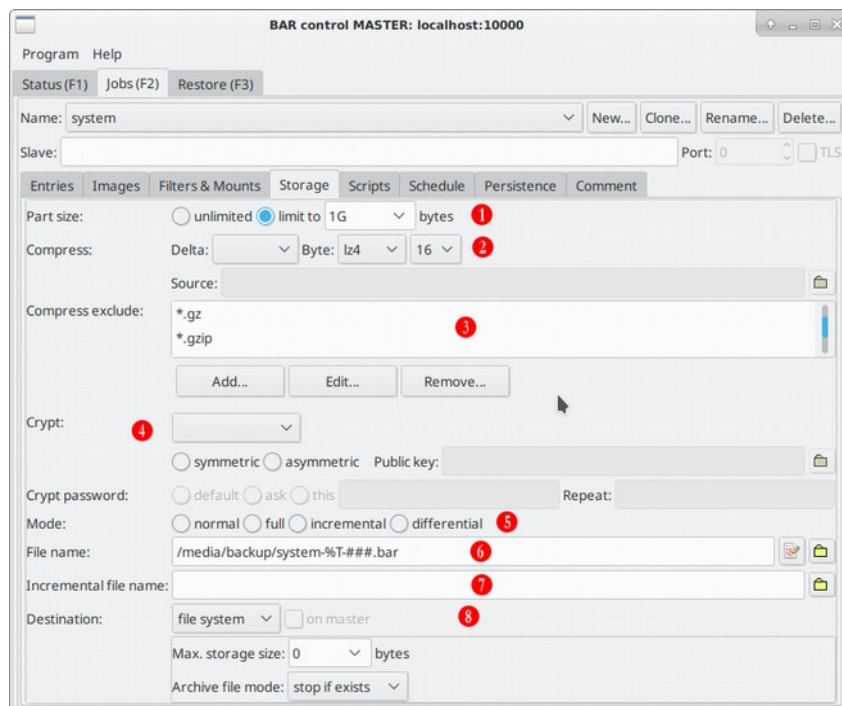




Figure 5: Storage

Created archives may be unlimited or split (1) and compressed (2). Entries should never be compressed, e. g. because they are already compressed, can be added to the compress exclude list

3. If the archives should be encrypted, too, the encryption algorithm, key and the password may be specified 4. Also the default archive mode 5 can be set here.

In the archive file name field 6 the button  open an editor dialog for the archive file name. With  button the system file selector dialog is opened. The archive name editor dialog include all possible variable patterns an archive name may include (see example figure 6: Archive name editor). The archive file name can simply be edited by drag and drop of the variable patterns into the name field. Parts can be removed by drag and drop onto the trashcan symbol or by pressing the DEL key on the keyboard.

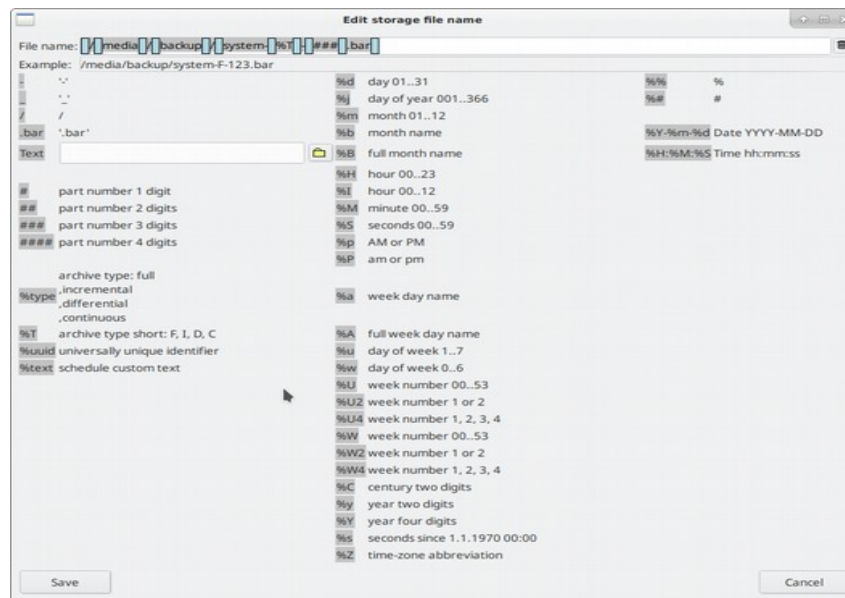


Figure 6: Archive name editor

In the incremental file name field 7 the incremental file name may be specified. In the destination settings 8 the storage destination of the created archives can be specified including storage or medium size limits, overwrite mode, host name, login credentials, device names, error correction code settings, and blanking a medium.

Scripts

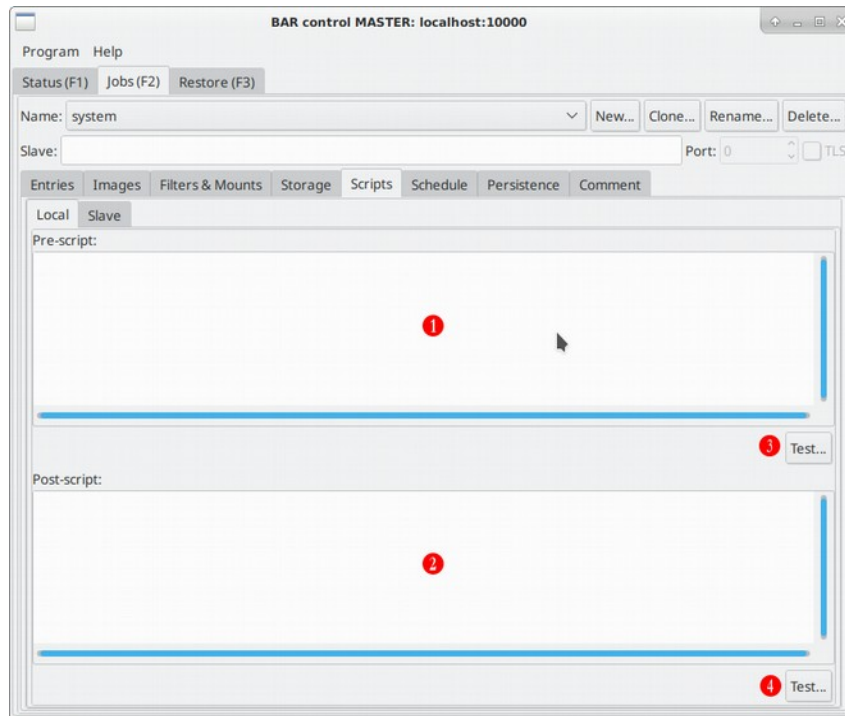


Figure 7: Scripts

In the scripts tab (see example figure 7: Scripts) pre- ❶ and post-scripts ❷ may be defined which are executed before the job starts and after the job has terminated. Scripts may contain special variables like %name, %archive, %type, %file, %directory which are expanded to the current job name, the full archive name, the archive file name and the archive directory part. With the test buttons ❸, ❹ the scripts can be executed for a test run.

Note: Scripts are executed with the current shell command defined with the environment variable SHELL or /bin/sh if that variable does not exists.

Schedule

In the schedule tab (see example figure 8: Scheduler) repeated schedule date ❶, weekdays ❷ and time ❸ for the backup job can be defined. Also the type of the created archive (default, normal, full, incremental, differential, or continuous) ❹, for continuous backups the time interval ❺. The time interval is the time window where changed files, directory, links, hard links or special files are collected for a continuous backup run. If no files are changed the backup job is skipped for the current time window. Optionally a custom text ❻ may be defined, too.

If only the incremental data should be collected the no storage option ❼ can be enabled. A schedule must be enabled to become active and executed. This can be done in the edit dialog or in the schedule list ❽.

Note: If a backup job was missed to execute because the *BAR* server was not running at the specified time, the missed job will executed immediately when the *BAR* server is restarted the next time. A job will be executed only if it was missed for max. 30 days.

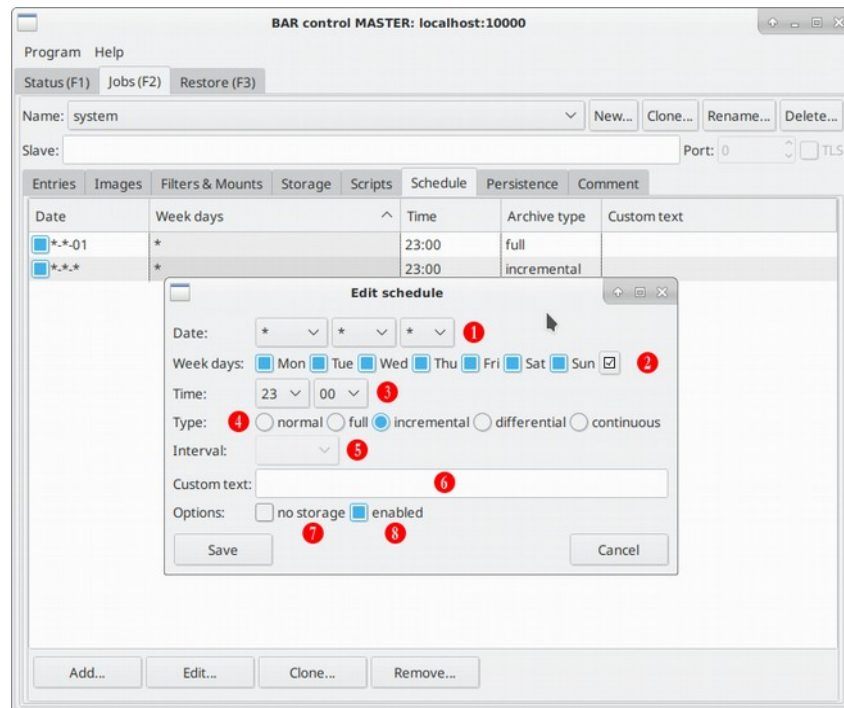


Figure 8: Scheduler

Persistence

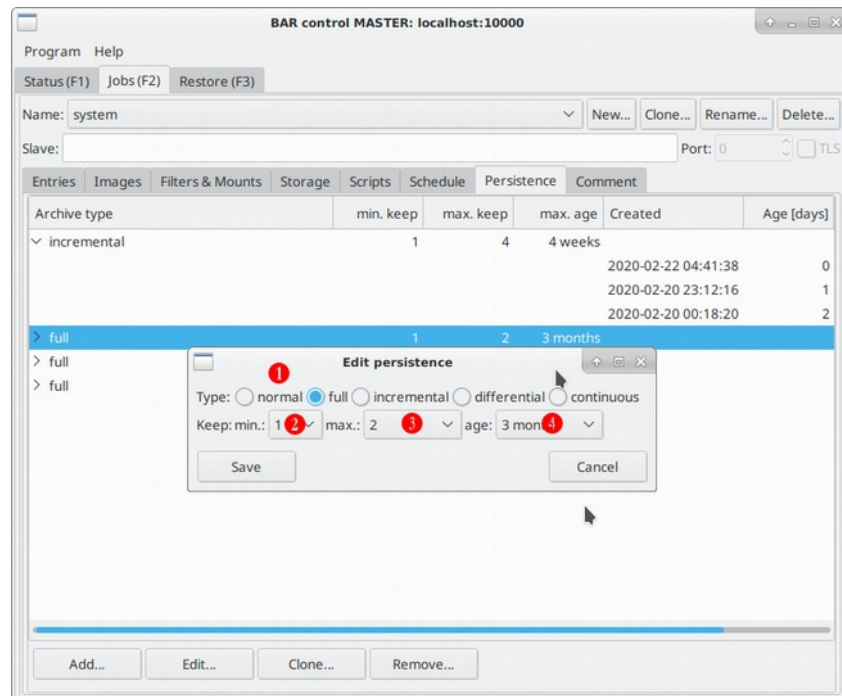


Figure 9: Persistence

In the persistence tab (see example figure 9: Persistence) the time periods to keep created archives can be defined. For each archive type **1** the minimal number of entities to keep **2**, the maximal number of entities to keep **3** and the maximal age of an entity **4** can be defined. If more entities exists or if an entity is older than the specified age then the entity and storage files of the entity are automatically deleted. With this feature a “ring buffer” behavior is supported, e. g. to hold the storage files for at least some specific number of backups, the maximal number of backups to hold and the max. age of a backup.

A typical persistence setup could be:

- keep incremental backups for four weeks
- keep one full backup for four weeks
- keep one full backup for six months
- keep one full backup for one year
- keep continuous backups for one week

Note: The oldest entity in a period may not be deleted if the entity is “in transit” to a following period. The condition when an archive is deleted is evaluated with:

- the archive is older than the max. age of the period
- the archive is not in transit to the next period
- the archive is the oldest in the period
- the number of existing archives in the period exceeds the max. number

Comment

In the comment tab a text comment can be entered which is included in the created archive files as a comment.

Restore tab

In the restore tab (see example figure 10: Restoring files) entries can be restored from archives.¹³ In the top tree view ① resp. list ② all known storage archives are listed. The tree view shows jobs¹⁴, entities and archives as a tree. The list shows just the archives of all jobs and entities. Via the context menu (click right mouse button) the existing archives may be added to the index database, removed or refreshed, archives or entities may be re-assigned to other jobs or entities and the entity type can be changed. It is also possible to remove archives from the index database as well as delete them from the storage location, too.

Note: Deleting storages is not reversible!

In the input field *Filter*: ③ parts of storage archive names can be entered (multiple words, at least 3 characters). Archive names which contain the filter words are listed immediately. The state filter ④ in the top list filters the archives by their current state. After selecting some storage archives with the checkboxes ⑤ the content of whole archives can be restored with the *Restore*-button ⑥.

The bottom list ⑦ shows entries which are stored in some archive. By entering parts of names (multiple words, at least 3 characters) in the input field *Filter*: ⑧ entries which contain the filter words are listed immediately¹⁵. After selecting some entries those entries can be restored with the *Restore*-button ⑨.

The *count* value ⑩, ⑪ in the upper left of the lists shows the number of total archives resp. entries.

If one or more archives are selected in the top list only entries in those archives are listed in the bottom list. With the ☒ button all entries in the lists can be selected resp. with the ☐ button the selection can be cleared.

The entry types shown in the bottom list can be filtered by files, images, directories, links, hard links and special entries with the type filter ⑫. Only entries with the matching type are shown in the list.

¹³ The internal database must be enabled to use the functions on the restore tab.

¹⁴ If a job is unknown, e. g. was deleted, only the internal UUID is shown instead of the name.

¹⁵ For technical reasons the total list length is limited. If more entries exist only the first 32000 may be viewed in the list.

If the check box *newest only* 13 only the newest entries are shown, otherwise all existing entries are shown.

After one of the restore buttons is selected a restore dialog opens (see example figure 11: Restore dialog) with a list of archives or single entries to restore 1. If a different restore destination is required, the directory 2 of the restore can be selected in the dialog. There also the check box *directory content* 3 can be selected to restore directories including their content. By default the signature of the archives are checked if there is a signature available. If the *skip signature check* box 4 is set the signatures are not checked. By selecting the check box *overwrite exiting files* 5 already existing files are overwritten. With the button *Start restore* 6 the restore of entries is started.

Note: Details for storage archives in the top list and entries in the lower list can be displayed by positioning the mouse over an entry in the right part of the list or via the context menu.

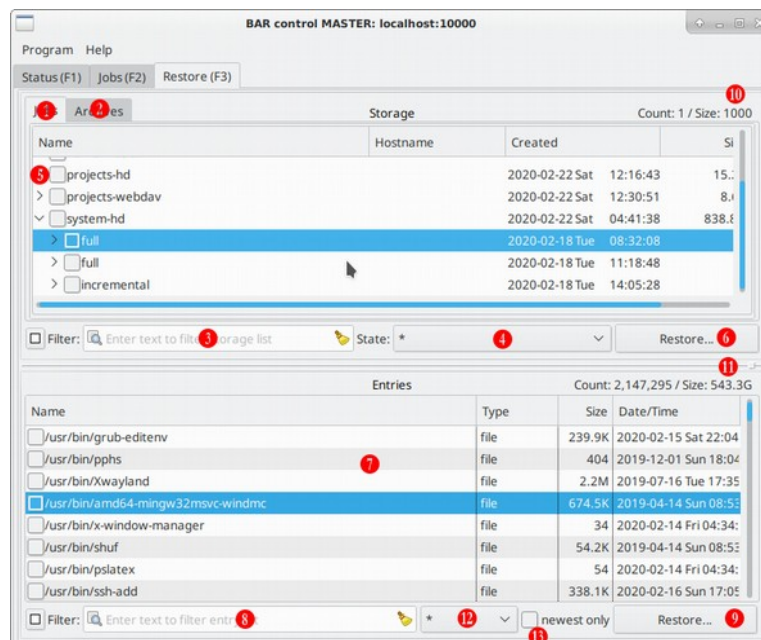


Figure 10: Restoring files

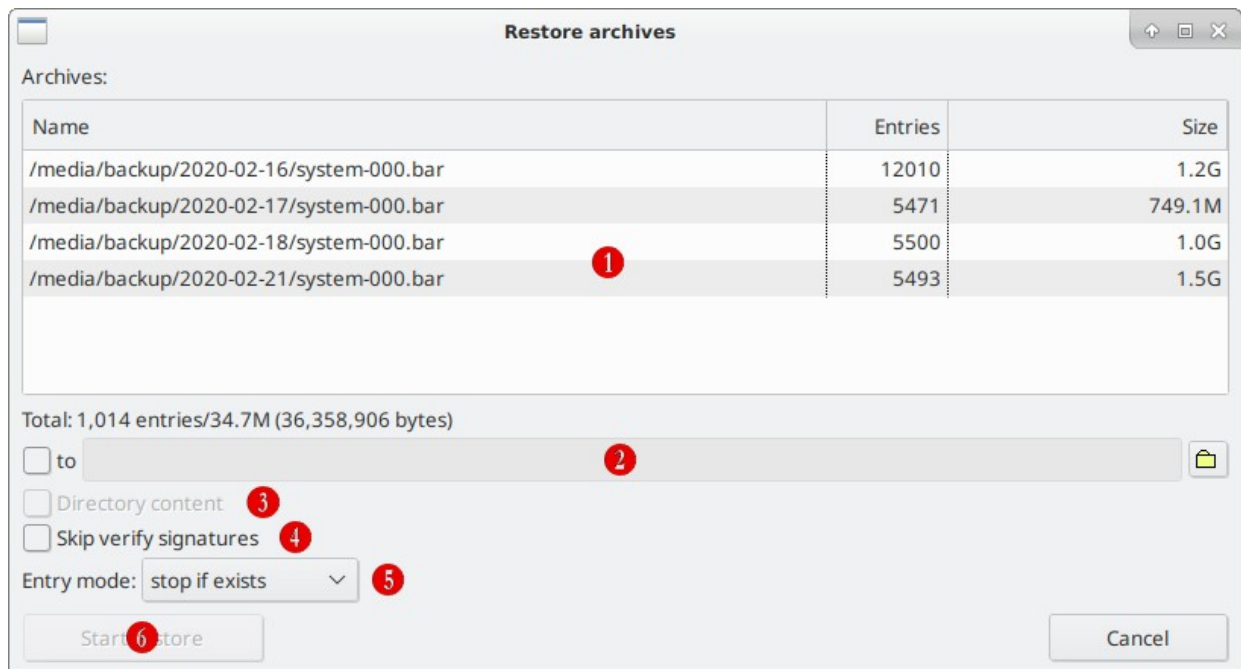


Figure 11: Restore dialog

Server settings

The server settings can be either modified by editing the text configuration file in `/etc/bar/bar.cfg` or with *BARControl* in the server settings dialog¹⁶ (see example figure 13: Servers settings). When the server configuration is changed with *BARControl*, the last writable configuration file in the list of configurations files is used to store the modifications.

Note: *BARControl* try to keep all information in the modified configuration file except the changed values. However there is no guarantee that manual made modifications in the file are not lost.

General

In the general sub-tab general settings of the *BAR* server can be modified (see example figure 12: Server general settings). This include the

- temporary directory ❶ and max. size ❷,
- the nice level ❸,
- the max. number of compress and encryption threads ❹,
- the minimal compress size ❺,
- the jobs directory ❻,
- the max. size of files included in a continuous backup ❼,
- the file name of the index database ❸ and if index auto-updates should be active ❹,
- the number of days archives which are not available anymore are kept in the index ❺.

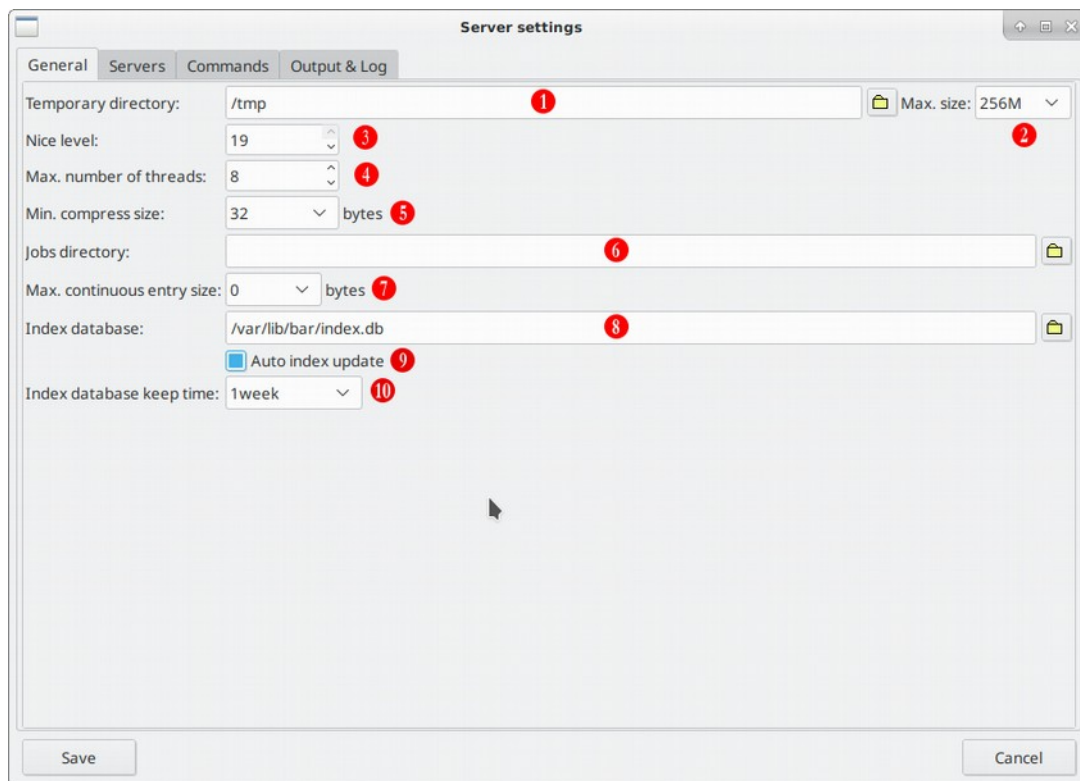


Figure 12: Server general settings

¹⁶ Several configuration files may exist which are concatenated in the ordering they are specified on the *BAR* command line.

Servers

In the server sub-tab (see example figure 13: Servers settings) the port number ❶ of the *BAR* server, the CA/Cert/key files ❷ and the server login password ❸ can be changed.

Here also names and login credentials can be set for a servers where archives can be stored ❹. There always exists default server settings for at least one FTP, SSH, and WebDAV server. Additional servers may be added.

Server settings

General Servers Commands Output & Log

Port: 38523 ❶

CA file: ❷

Cert file:

Key file: ❸

Password:

Storage servers:

Type	Name	#	Size
ftp	default	-	-
ftp	ftp.4shared.com	-	-
ftp	storage01.webcity.com.au	-	-
ssh	default	-	-
ssh	foo.org	-	-
ssh	localhost ❹	-	-
webdav	default	-	-
webdav	webdav.hidrive.strato.com	-	-
webdav	webdav.mc.gmx.net	-	-

Add... Edit... Clone... Remove...

Save Cancel

Figure 13: Servers settings

Commands

In the commands sub-tab (see example figure 14: Server command settings) the device name ❶ and external commands for writing CD/DVD/BD and devices ❷ can be defined.

The screenshot shows the 'Server settings' dialog box with the 'Commands' tab selected. The 'CD' sub-tab is active. The 'Device' field is set to '/dev/cdrw' (marked with ❶). The 'Image command' field is set to 'nice mkisofs -V Backup -volset %number -r -o %image %directory' (marked with ❷). Other fields include 'Request volumn command:', 'Unload command:', 'Load command:', 'Image pre-command:', 'Image post-command:', 'ECC pre-command:', 'ECC post-command:', 'ECC command:', 'Blank command:', 'Write pre-command:', 'Write post-command:', 'Write command:', and 'Image write command:'.

Figure 14: Server command settings

Verbosity & Log

In the verbosity and log sub-tab (see example figure 15: Server verbosity & log settings) the

- log level ❶,
- the log types ❷,
- the log file name ❸,
- the log format ❹ and
- the log post command ❺

can be modified.

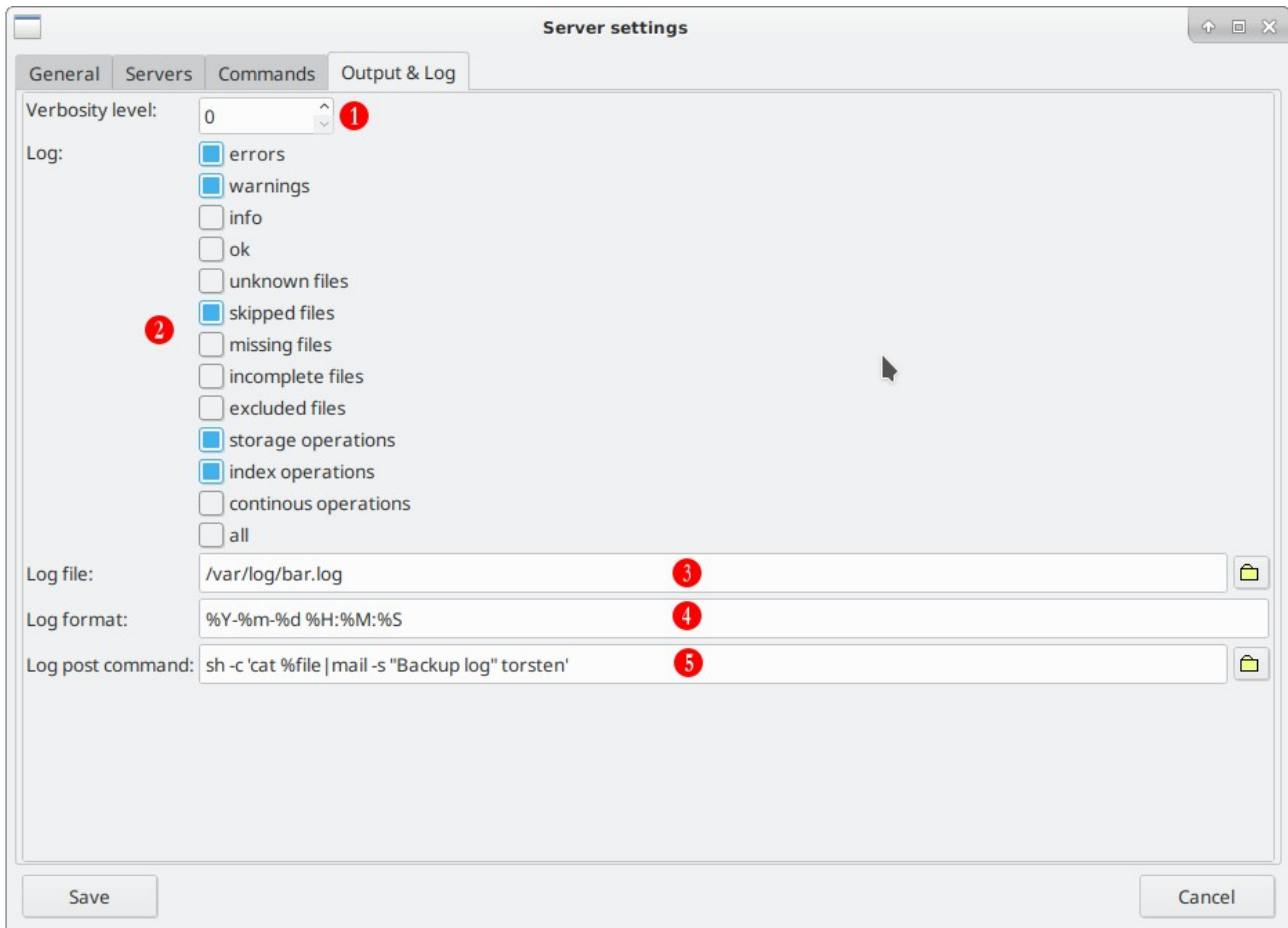


Figure 15: Server verbosity & log settings

Use cases

BAR can be used to create and restore backups in various ways and with various options. This chapter describes some common use cases of *BAR* in server mode.

a) Normal backup

A normal backup of some entries (files, directories, links, hard links, special files, or disks) can be done with the following settings in *BARControl*:

- in the *Jobs* tab create a new job with some appropriated name
- in the *Files* sub-tab below select some files or directories. For the whole disk select the root directory of the disk. Click then the *Include* button.
- switch to the *Storage* sub-tab. Select the name of the archive to create and select the type of storage. Most simply enter a archive name like *my-backup.bar* and *a* as the type *file system*. This will store the archive into the file system with the given name.

For a full backup this is all to do. The backup can be started in the *Status* tab by selecting the job and then click the button *Start*. In the dialog which opens click *Normal*. The job is immediately schedule for execution and will be run as soon as possible.

b) Full daily backup with time schedule

For a full backup of some entries with a regular scheduling first define some backup job as described in use case a) Normal backup. Then enter the following additional settings in *BARControl*:

- switch to the *Jobs* tab. In the *Schedule* sub-tab add a new schedule entry and define some time when to execute the job. Keep for the date * and select all week days.
- select for the type *full*
- verify that *enabled* is checked

The backup job is executed automatically at the specified time¹⁷ each day as a full backup.

c) Full weekly backup with time schedule and daily incremental backup

For a full weekly backup of some entries with a regular scheduling and daily incremental backups define some backup job as described in use case a) Normal backup. Then enter the following additional settings in *BARControl*:

- switch to the *Jobs* tab. In the *Schedule* sub-tab add a new schedule entry and define * for the date and time. Select the week day when to do execute the full backup.
- select for the type *full*
- add another new schedule entry and define * for the date and time. Select all week days where the incremental backup should be execute. Exclude the day when the full backup runs.
- select for the type *incremental*
- verify that *enabled* is checked

d) Full weekly backup with daily incremental and continuous backup

For a full weekly backup of some entries with with a regular scheduling, daily incremental backups and continuous storages of changes define some backup job as described in use case c) Full weekly backup with time schedule and daily incremental backup. Then enter the following additional settings in *BARControl*:

- switch to the *Jobs* tab. In the *Schedule* sub-tab add a new schedule entry and define * for the date, time and week day.
- select for the type *continuous*
- select some interval time when the continuous backup should be executed. A good choice maybe 30min which means every 30min a backup job is started storing all entries which were changed in the last 30min.

Note: continuous backups may require a lot of archive space depending on the size of the files to store, how often the are modified and time interval the backup is executed.

¹⁷ The exact time may be later than the specified time if some other jobs are running before.

e) Restore single file or directory

To restore a single entry do the following in *BARControl*:

- switch to the *Restore* tab
- in the filter of the bottom entry list enter parts of the path or name of the entry to restore. The list is then updated to show all matching entries available in some archives.
- filter the number of entries in the list by select the entry type or select *newest only* if needed
- click the button *Restore...* and in the dialog which opens select if you want to restore the entry into some different location and/or if you want that an existing entry should be overwritten.
- click *Start*. A restore job is scheduled and will be executed as soon as possible.

f) Restore full backup

To restore a a full backup including incremental backups do the following in *BARControl*:

- switch to the *Restore* tab
- in the top tree select a job and the entity to restore.
- click the button *Restore...* and in the dialog which opens select if you want that existing entries should be overwritten.
- click *Start*. A restore job is scheduled and will be executed as soon as possible.

Appendix

Tools

The following tools are part of *BAR*:

- bar: main program
- barcontrol: graphical front end
- bar-keygen: public and private key generator
- bar-sqlite3: index a maintenance and repair-tool

Archive file name macros


The following table list the macros which may be used in an archive file name. The macros are expanded on run time to create the final used archive file name.

Macro	Description	Example
#	single digit of part number	1
%type	archive type (normal, full, incremental, differential)	incremental
%T	archive type short (N, F, I, D)	D
%last	last for last part, empty otherwise	last
%uuid	unique id	76b5af97-690d-4418-bf9e-1051a6a10d48
%text	custom text (from schedule definition)	foo
%d	day of month (01..31)	04
%j	day of year (001..366)	134
%m	month (01..12)	06
%b	month name (Jan..Dec) ¹⁸	Jun
%B	full month name (January...December) ¹⁸	June
%H	hour (00..23)	20
%I	hour (00..12)	08
%M	minute (00..59)	25
%p	AM or PM	PM
%a	week day name (Mon..Sun) ¹⁸	Tue
%A	full week day name (Monday...Sunday) ¹⁸	Tuesday
%u	day of week (1..7)	2
%w	day of week (0..6)	1
%U	week number starting with first Sunday (00..53)	22
%U2	week number 1, 2 starting with first Sunday	1

¹⁸ The name depend on the current locale.

%U4	week number 1, 2, 3, 4 starting with first Sunday	3
%W	week number starting with first Monday (00..53)	22
%W2	week number 1, 2 starting with first Monday	1
%W4	week number 1, 2, 3, 4 starting with first Monday	3
%C	century two digits	20
%y	year two digits	11
%Y	year four digits	2011
%S	seconds since 1 st January 1970 00:00:00	12345678
%Z	time zone abbreviation	JST
%%	insert %	%
%#	insert #	#

Table 5: Archive file name macros

Note: *BARControl* has a nice archive file name editor. See the  at the archive file name field in tab *Jobs*, sub-tab *Storage* (see Storage, page 29).

BAR command line options summary

The following sections list the *BAR* command line options.

Basic commands

--create, -c

Create an archive with files, directories, links, hard links and special files like device-, pipes- and sockets-descriptors.

--image, -m

Create an archive with disk or partition images. *BAR* try to detect the file system on disks or partitions and will store not used data blocks with binary content “0”. To disable this feature use option *--raw-images*.

Note: When compression is enabled not used data blocks will almost use no space in the created archive.

--list, -l

List content of archives. Amount and type of data shown can be selected with options *--long-format*, *--human-format* and *--no-header-footer*.

--test, -t

Test content integrity of archives.

--compare, -d

Compare content of archives with content in file system.

--extract, -x

Extract content of archives.

--convert

Convert an archive with new compression, encryption, pass phrase or signature.

--generate-keys

Generate a public/private key pair for asymmetric encryption/decryption of archive content with RSA. The public key is used to encrypt the content of an archive, the private key is required to decrypt the content of an archive.

Note: access to the private key should be limited!

Note: RSA encryption is only used for the public/private key pair. The archive content itself is encrypted with the selected encryption algorithm and a random selected pass phrase key of 2048 bit. The random password is encrypted with the public key and stored into the archive. For each archive part a new random password is generated. This mechanism is known as “hybrid-encryption”. Using hybrid-encryption instead of plain RSA encryption is a common technique to avoid the expensive and usually slow RSA encryption for a huge amount of data.

--generate-signature-keys

Generate a public/private signature key pair. The private signature key is used to create a signature for the archive. With the public signature key the signature can be verified that the authenticity of the archive is valid and the archive was not modified.

Note: The signature is calculated with the SHA2 512 hash algorithm.

--generate-keys-bits=<n>

Specify number of bits to use for a public/private key pair. It can be 1024, 2048 or 3072 bits. Default value are 2048 bits.

--generate-keys-mode=secure|transient

Specify the mode to generate a public/private key pair. *secure* use the most secure random number generator and is the default (recommended). *transient* use a faster, but a less secure random number generator.

--storage-list-stdin,-T

Read storage names (glob pattern) from stdin.

--storage-list=<file name>

Define the name of a file from which storage names or simple patterns (glob pattern) are read.

--storage-command=<command|script>

Define a command or script to execute before the archive is created. The output (stdout) of the command or script define storage names or simple patterns (glob pattern).

--pre-command=<command>

Command to execute before a job is started.

Available macros: %name.

--post-command=<command>

Command to execute after a job terminated.

Available macros: %name.

--mount=<name>[,<device>]

Device to mount before a job is started and to unmount after the job terminated. If <device> is given mount that device.

--mount-command=<command>

Command to mount a device via a mount point.

Available macros: %directory

--mount-device-command=<command>

Command to mount a device.

Available macros: %directory, %device

--unmount-command=<command>

Command to unmount a device via a mount point.

Available macros: %directory

--comment=<text>

Add comment to archive.

--info

Show meta info of archive.

Include/Exclude files

--include=<pattern>, -#

Define a simple file pattern (glob pattern), a regular expression, or a extended regular expression which must match to files which are included into the created archive. The default pattern type is simple file pattern. To select regular expression or extended regular expression patterns use one of the prefixes listed in the table 6: Pattern types.

Prefix	Type	Example
g:	simple pattern (glob pattern)	g:/home/foo/*.txt
r:	regular expression	r:/home/foo/.*.txt
x:	extended regular expression	x:/home/foo/.*.txt

Table 6: Pattern types

--include-file-list=<file name>

Define the name of a file from which names or simple patterns (glob pattern) are read which

must match to entries which are included into the created archive.

`--include-file command=<command|script>`

Define a command or script to execute before the archive is created. The output (stdout) of the command or script define names or simple patterns (glob pattern) which must match to files which are included into the created archive.

`--include-image-list=<file name>`

Define the name of a file from which names or simple patterns (glob pattern) are read which must match to entries which are included into the created archive.

`--include-image-command=<command|script>`

Define a command or script to execute before the archive is created. The output (stdout) of the command or script define names or simple patterns (glob pattern) which must match to files which are included into the created archive.

`--exclude=<pattern>, -!`

Define a simple file pattern (glob pattern), a regular expression, or a extended regular expression which must match to files which are excluded from the created archive. Default pattern type is simple file pattern. See table 6 how to specify different pattern types.

Note: Exclusions have precedence over include patterns.

`--exclude-list=<file name>`

Define the name of a file from which names or simple patterns (glob pattern) are read which must not match to files and are excluded from the created archive.

`--exclude-command=<command|script>`

Define a command or script to execute before the archive is created. The output (stdout) of the command or script define names or simple patterns (glob pattern) which must not match to files and are excluded from the created archive.

Note: Exclusions have precedence over include patterns.

`--storage-list=<file name>`

Define the name of a file from which storage names or simple patterns (glob pattern) are read.

`--storage-command=<command|script>`

Define a command or script to execute to get a list of storages. The output (stdout) of the command or script define names or simple patterns (glob pattern) of storage names.

Incremental/Differential archives

`--normal`

Create a normal archive. No incremental data is used or stored.

`--full, -f`

Create a full archive and store incremental data.

`--incremental, -i`

Create an incremental archive by using stored incremental data. All files, directories, links, hard links or special files which changed since the incremental data was collected or updated are stored into the archive. The stored incremental data is updated after creating an incremental archive.

See option `--incremental-list-file` to specify the incremental data file to use.

`--incremental-list-file=<file name>`

Define the file to use for storing incremental data. If the file does not exist, it is created when the option `--full` is specified.

Note: Incremental data files are not compressed nor encrypted and are not stored on CD/DVD/BD or transmitted to a remote server. They are used only locally by *BAR*.

`--differential`

Create an differential archive by using stored differential data. All files, directories, links, hard links or special files which changed since the incremental data was collected are stored in the archive. The incremental data is not updated after creating an differential archive.

See option `--incremental-list-file` to specify the incremental data file to use.

Compression

`--compress-algorithm, -z`

Specify the compression algorithms to use for archive content data. Which compression algorithm can be used depend on the compiled-in libraries. If all supported compression algorithms are compiled into *BAR* then either none or one of the byte compression algorithms zip [zlib], bzip2 [bzip2], lzma [LZMA], lzo [LZO], lz4 [LZ4], Zstandard [Zstandard], and/or the delta compression algorithms xdelta [xdelta] algorithms can be used.¹⁹ Usually lzma have a better compression ratio than bzip2 than zip, but also need more resources (CPU time and memory). lzo, lz4, and Zstandard offer high speed compression at the expense of a limited compression ratio.

Xdelta compression can be used efficiently for pairs of huge files which only change in a small number of bytes. Additionally to xdelta compression the zip, bzip2, lzma, lzo, lz4, or Zstandard compression algorithms can be used to compress the delta-data.

The total ratio depend on the size and type of data which should be compressed. Small files and already compressed files can usually not further be compressed by zip, bzip2, lzma, lzo, lz4, or Zstandard. Files with changes in a large number of bytes between two versions can usually not further be compressed by the delta algorithms.

The different compression algorithms have different levels of compression. Usually a higher level result in a better compression, but require more CPU time or memory. Table 7: Compression algorithms list all supported compression algorithms. Table

¹⁹ Pre-compiled binary packages contain all supported compression libraries.

Name	Description
none	no compression
zip1	ZIP compression level 1
zip2	ZIP compression level 2
zip3	ZIP compression level 3
zip4	ZIP compression level 4
zip5	ZIP compression level 5
zip6	ZIP compression level 6
zip7	ZIP compression level 7
zip8	ZIP compression level 8
zip9	ZIP compression level 9
bzip1	BZIP2 compression level 1
bzip2	BZIP2 compression level 2
bzip3	BZIP2 compression level 3
bzip4	BZIP2 compression level 4
bzip5	BZIP2 compression level 5
bzip6	BZIP2 compression level 6
bzip7	BZIP2 compression level 7
bzip8	BZIP2 compression level 8
bzip9	BZIP2 compression level 9
lzma1	LZMA compression level 1
lzma2	LZMA compression level 2
lzma3	LZMA compression level 3
lzma4	LZMA compression level 4
lzma5	LZMA compression level 5
lzma6	LZMA compression level 6
lzma7	LZMA compression level 7
lzma8	LZMA compression level 8
lzma9	LZMA compression level 9
lzo1	LZO compression level 1: LZO1X-1(11)
lzo2	LZO compression level 2: LZO1X-1(12)
lzo3	LZO compression level 3: LZO1X-1(15)
lzo4	LZO compression level 4: LZO1X-1
lzo5	LZO compression level 5: LZO1X-999
lz4-0	LZ4
lz4-1	LZ4 HC compression level 1

Name	Description
lz4-2	LZ4 HC compression level 2
lz4-3	LZ4 HC compression level 3
lz4-4	LZ4 HC compression level 4
lz4-5	LZ4 HC compression level 5
lz4-6	LZ4 HC compression level 6
lz4-7	LZ4 HC compression level 7
lz4-8	LZ4 HC compression level 8
lz4-9	LZ4 HC compression level 9
lz4-10	LZ4 HC compression level 10
lz4-11	LZ4 HC compression level 11
lz4-12	LZ4 HC compression level 12
lz4-13	LZ4 HC compression level 13
lz4-14	LZ4 HC compression level 15
lz4-15	LZ4 HC compression level 15
lz4-16	LZ4 HC compression level 16
zstd0	Zstandard compression level 0
zstd1	Zstandard compression level 1
zstd2	Zstandard compression level 2
zstd3	Zstandard compression level 3
zstd4	Zstandard compression level 4
zstd5	Zstandard compression level 5
zstd6	Zstandard compression level 6
zstd7	Zstandard compression level 7
zstd8	Zstandard compression level 8
zstd9	Zstandard compression level 9
zstd10	Zstandard compression level 10
zstd11	Zstandard compression level 11
zstd12	Zstandard compression level 12
zstd13	Zstandard compression level 13
zstd14	Zstandard compression level 14
zstd15	Zstandard compression level 15
zstd16	Zstandard compression level 16
zstd17	Zstandard compression level 17
zstd18	Zstandard compression level 18
zstd19	Zstandard compression level 19

Name	Description
xdelta1	xdelta compression level 1
xdelta2	xdelta compression level 2
xdelta3	xdelta compression level 3
xdelta4	xdelta compression level 4
xdelta5	xdelta compression level 5
xdelta6	xdelta compression level 6
xdelta7	xdelta compression level 7
xdelta8	xdelta compression level 8
xdelta9	xdelta compression level 9

Table 7: Compression algorithms

Note: In the archive files created by *BAR* each entry is compressed separately. This may result in slightly bigger files than archives which are compressed at whole, because meta data like the archive structuring data is not compressed as well compression of the separated files may be not as efficient than a single block of files. Nevertheless when each entry is compressed separately, each entry can be read and extracted separately as well, thus result in faster listing and access of single entries in an archive file.

`--compress-min-size=<size>`

Specify the minimal size in bytes a file must have to become compressed. Usually small files cannot be compressed efficiently. Thus it is better to store them directly into the archive. By default the minimal size is 32bytes.

`--compress-exclude=<pattern>`

Define a simple file pattern (glob pattern), a regular expression, or a extended regular expression which must match to files, images or hard links which are excluded from compression in the created archive. This option may be useful to suppress compression for entries which are already compressed, e. g. JPEG files, zip files a. o., and which may not be compressed further. Default pattern type is simple file pattern. See table 6: Pattern types how to specify a different pattern type to use.

`--delta-source=<pattern>`

Delta source pattern to find delta source files for delta compression.

Encryption

`--crypt-algorithm, -y`

Specify the encryption algorithm to use for archive content data. Which encryption algorithm can be used depend on the compiled-in libgcrypt [libgcrypt] library.²⁰ Table 8: Encryption algorithms list all supported encryption algorithms.

²⁰ Pre-compiled binary packages contain all supported libraries.

Name	Description	Block size [bytes]
none	no encryption; store as clear-text data	1
3DES	3DES cipher	8
CAST5	CAST5 cipher	8
BLOWFISH	Blowfish cipher	8
AES128	AES cipher 128bit	16
AES192	AES cipher 192bit	16
AES256	AES cipher 256bit	16
TWOFISH128	Twofish cipher 128bit	16
TWOFISH256	Twofish cipher 256bit	16
SERPENT128	Serpent cipher 128bit	16
SERPENT192	Serpent cipher 192bit	16
SERPENT256	Serpent cipher 256bit	16
CAMELLIA128	Camellia 128bit	16
CAMELLIA192	Camellia 192bit	16
CAMELLIA256	Camellia 256bit	16

Table 8: Encryption algorithms

Usually one of the AES encryption algorithms, e. g. AES 256, is a good choice. For details about the encryption algorithms see the documentation of *libgcrpt* [libgcrpt] or http://en.wikipedia.org/wiki/Block_cipher_security_summary.

--crypt-type=symmetric|asymmetric

Use symmetric or asymmetric encryption (RSA).

Note: For asymmetric encryption a hybrid-encryption is used. See option --generate-keys for details.

--crypt-password=<password>

Default crypt password phrase to use.

Note: Use this option with care! Password phrases specified on the command line may be stored in the history of your shell, thus can be recovered by displaying the command history.

--crypt-new-password=<password>

New crypt password phrase to use when converting an archive.

Note: Use this option with care! Password phrases specified on the command line may be stored in the history of your shell, thus can be recovered by displaying the command history.

--crypt-public-key=<file name>

Name of file with the RSA encryption public key. The encryption public key is use to encrypt an archive.

--crypt-private-key=<file name>

Name of file with the RSA encryption private key. The encryption public is used to decrypt an archive.

Note: Because the RSA encryption private key is a critical piece of data, always keep this key at a secret place!

Signatures

--signatures-public-key=<file name>

Name of file with the RSA signature public key. The signature public key is used to verify the data integrity of an archive.

--signature-private-key=<file name>

Name of file with the RSA encryption private key. The signature public key is used to add a signature to an archive.

Note: Because the RSA signature private key is a critical piece of data, always keep this key at a secret place!

--force-verify-signatures

Force verification of signatures. If the signature public key is not available or the signature is invalid an error is reported. If this option is not given only a warning may be shown.

--skip-verify-signatures

Skip verification of signatures and do not show an error or a warning.

--no-signatures

Do not add a signature in created archives .

Splitting

--archive-part-size=<size>, -s=<size>

Create archive parts of the specified size.

Note: The size of a single created part may be slightly bigger than the specified number of bytes. The reason for this are internal buffers used in the compression algorithms which must be stored, too, when an archive part is completed.

--fragment-size=<size>

Define the size of a fragment for splitting huge files. Each fragment is processed in a separated thread to compress and encrypt the fragment.

Storage

File

`--file-write-pre-command=<command>`

Command to execute before an archive is written to the file system.

Available macros: %file, %number.

`--file-write-post-command=<command>`

Command to execute after an archive is written to the file system

Available macros: %file, %number.

FTP

`--ftp-login-name=<name>`

Specify the general FTP login name.

`--ftp-password=<password>`

Specify the general FTP login password.

Note: Use this option with care! Passwords specified on the command line may be stored in the history of your shell, thus can be recovered by displaying the command history.

`--ftp-max-connections=<n>`

Specify the max. number of concurrent connections for this FTP server.

`--ftp-write-pre-command=<command>`

Command to execute before file is transmitted to FTP server.

Available macros: %file, %number.

`--ftp-write-post-command=<command>`

Command to execute after file is transmitted to FTP server.

Available macros: %file, %number.

SSH

`--ssh-port=<n>`

Specify ssh port to use. Default is 22.

`--ssh-login-name=<name>`

Specify the default ssh login name.

`--ssh-password=<password>`

Specify the default ssh login password.

Note: Use this option with care! Passwords specified on the command line may be stored in the

history of your shell, thus can be recovered by displaying the command history.

--ssh-public-key=<file name>

Specify the file name with the ssh public key.

--ssh-private-key=<file name>

Specify the file name with the ssh private key.

--ssh-max-connections=<n>

Specify the max. number of concurrent connections for this SSH/SCP/SFTP server.

SCP

--scp-write-pre-command=<command>, --ftp-write-post-command=<command>

Command to execute before file is transmitted with SCP protocol.

Available macros: %file, %number.

--scp-write-post-command=<command>

Command to execute after file is transmitted with SCP protocol.

Available macros: %file, %number.

SFTP

--sftp-write-pre-command=<command>

Command to execute before file is transmitted to SFTP server.

Available macros: %file, %number.

--sftp-write-post-command=<command>

Command to execute after file is transmitted to SFTP server.

Available macros: %file, %number.

WebDAV/WebDAVS

--webdav-login=<name>

Specify the default WebDAV login name.

--webdav-password=<password>

Specify the default WebDAV login password.

Note: Use this option with care! Passwords specified on the command line may be stored in the history of your shell, thus can be recovered by displaying the command history.

--webdav-max-connections=<n>

Specify the max. number of concurrent connections for this WebDAV server.

`--webdav-write-pre-command=<command>`

Command to execute before file is transmitted to WebDAV server.

Available macros: %file, %number.

`--webdav-write-post-command=<command>`

Command to execute after file is transmitted to WebDAV server.

Available macros: %file, %number.

CD/DVD/BD

`--cd-device=<device name>, --dvd-device=<device name>, --bd-device=<device name>`

Default device name for creating CDs/BDs/DVDs.

`--cd-request-volume-command=<command>, --dvd-request-volume-command=<command>, --bd-request-volume-command=<command>`

Command to execute when a new CD/DVD/BD is required and should be placed into the CD/DVD/BD drive. Usually this command open some interactive dialog for the user. If a CD/DVD/BD changer unit is available the appropriated command to change the CD/DVD/BD may be send to this unit as well.

Available macros: %device, %number.

`--cd-unload-volume-command=<command>, --dvd-unload-volume-command=<command>, --bd-unload-volume-command=<command>`

Command to unload a CD/DVD/BD volume. The default value is the *eject* command.

Available macros: %device, %number.

`--cd-load-volume-command=<command>, --dvd-load-volume-command=<command>, --bd-load-volume-command=<command>`

Command to load a CD/DVD/BD volume. The default value is the *eject* command.

Available macros: %device, %number.

`--cd-volume-size=<n>[G|M|K], --dvd-volume-size=<n>[G|M|K], --bd-volume-size=<n>[G|M|K]`

CD/DVD/BD volume size.

Default values are 700MB, 4.4GB, and 25GB for CD/DVD/BD without error correction codes and 560MB, 3.4GB, and 20GB for CD/DVD/BD with error correction codes

Note: Of course CDs/DVDs/BDs can be completely filled. Nevertheless if *dvdaster* is used as a post processing tool when creating a CDs/DVDs/BDs the size of 560MB/3.4GB/20GB is a useful value for CDs/DVDs/BDs. The rest of the available space is then used by *dvdaster* to store Reed-Solomon error correction codes. These error correction codes may be used to recover physically destroyed parts on the CD/DVD/BD. For more information to *dvdaster* see [dvdaster].

`--cd-image-pre-command=<command>, --dvd-image-pre-command=<command>, --bd-image-pre-command=<command>`

Pre-process command to execute before creating a CD/DVD/BD image. This can be some

arbitrary command which should always be executed before a new CD/DVD/BD image is created.

--cd-image-post-command=<command>, --dvd-image-post-command=<command>, --bd-image-post-command=<command>

Post-process command to execute after creating a CD/DVD/BD image. This can be some arbitrary command which should always be executed after a new CD/DVD/BD image is created.

--cd-image-command=<command>, --dvd-image-command=<command>, --bd-image-command=<command>

Command to create a CD/DVD/BD image. Default is *mkisofs*.

Available macros: %device, %file, %image, %sectors, %number, %j, %j1.

Note: %i is the number of available CPU cores, %j1 is the number of available CPU cores minus 1, but at least 1.

--cd-ecc-pre-command=<command>, --dvd-ecc-pre-command=<command>, --bd-ecc-pre-command=<command>

Command to execute before error correction codes (ecc) are added to a CD/DVD/BD image.

--cd-ecc-post-command=<command>, --dvd-ecc-post-command=<command>, --bd-ecc-post-command=<command>

Command to execute after error correction codes (ecc) are added to a CD/DVD/BD image.

--cd-ecc-command=<command>, --dvd-ecc-command=<command>, --bd-ecc-command=<command>

Command to add error correction codes (ecc) to a CD/DVD/BD image. A useful command is *dvdisaster* to add Reed-Solomon codes to the CD/DVD/BD image.

Available macros: %device, %file, %image, %sectors, %number, %j, %j1.

Note: %i is the number of available CPU cores, %j1 is the number of available CPU cores minus 1, but at least 1.

--cd-blank-command=<command>, --dvd-blank-command=<command>, --bd-blank-command=<command>

Command to blank a CD/DVD/BD medium before writing.

--cd-write-pre-command=<command>, --dvd-write-pre-command=<command>, --bd-write-pre-command=<command>

Command to execute before CD/DVD/BD image is written.

--cd-write-post-command=<command>, --dvd-write-post-command=<command>, --bd-write-post-command=<command>

Command to execute after CD/DVD/BD image is written.

--cd-write-command=<command>, --dvd-write-command=<command>, --bd-write-command=<command>

Command to write on CD/DVD/BD.

Available macros: %device, %file, %image, %sectors, %number, %j, %jl.

Note: %i is the number of available CPU cores, %jl is the number of available CPU cores minus 1, but at least 1.

--cd-write-image-command=<command>, --dvd-write-image-command=<command>, --bd-write-image-command=<command>

Command to write a CD/DVD/BD image.

Available macros: %device, %file, %image, %sectors, %number, %j, %jl.

Note: %i is the number of available CPU cores, %jl is the number of available CPU cores minus 1, but at least 1.

Device

--device=<device name>

Specify default device name.

--device-request-volume-command=<command>

Command to request new volume for device.

--device-load-volume-command=<command>

Command to load volume for device.

--device-unload-volume-command=<command>

Command to unload volume from device.

--device-volume-size=<size>

Device volume size.

--device-image-pre-command=<command>

Pre-process command to execute before creating a device image. This can be some arbitrary command which should always be executed before a new device image is created.

Available macros: %device, %file, %image, %number, %j, %jl.

Note: %i is the number of available CPU cores, %jl is the number of available CPU cores minus 1, but at least 1.

--device-image-post-command=<command>

Post-process command to execute after creating a device image. This can be some arbitrary command which should always be executed after a new device image is created.

Available macros: %device, %file, %image, %number, %j, %jl.

Note: %i is the number of available CPU cores, %jl is the number of available CPU cores minus 1, but at least 1.

--device-image-command=<command>

Command to create a device image.

Available macros: %device, %file, %image, %number, %j, %jl.

Note: %i is the number of available CPU cores, %jl is the number of available CPU cores minus 1, but at least 1.

--device-ecc-pre-command=<command>

Command to execute before error correction codes (ecc) are added to a device image.

Note: Still not implemented!

--device-ecc-post-command=<command>

Command to execute after error correction codes (ecc) are added to a device image.

Note: Still not implemented!

--device-ecc-command=<command>

Command to added error correction codes (ecc) to a device image.

Note: Still not implemented!

--device-blank-command=<command>

Command to blank a device medium before writing.

--device-write-pre-command=<command>

Command to execute before device image is written.

Available macros: %device, %file, %image, %number, %j, %jl.

Note: %i is the number of available CPU cores, %jl is the number of available CPU cores minus 1, but at least 1.

--device-write-post-command=<command>

Command to execute after device image is written.

Available macros: %device, %file, %image, %number, %j, %jl.

Note: %i is the number of available CPU cores, %jl is the number of available CPU cores minus 1, but at least 1.

--device-write-command=<command>

Command to write a device image.

Available macros: %device, %file, %image, %number, %j, %jl.

Note: %i is the number of available CPU cores, %jl is the number of available CPU cores minus 1, but at least 1.

Restore

--directory-strip=<n>, -p=<n>

Strip specified number of directory levels from file names of restored files.

`--destination=<path>`

Restore files into specified path.

`--owner=<user>:<group>`

Set specified user and group for restored entries. `<user>` is a user name which must exist on the system. `<group>` is a group name which must exist on the system.

`--permissions=<owner>[:<group>[:<world>]]`

Set permission RWX of owner, group, or world when restoring entries.

Server

`--daemon`

Run *BAR* in daemon mode (as a server).

`--no-detach, -D`

Do not detach process when running in daemon mode.

`--server-port=<n>21`

Specify server port to use. Default is: 38523.

`--server-tls-port=<n>21`

Specify TLS (SSL) server port to use. Default is: 38524.

`--server-ca-file=<file name>`

Specify TLS (SSL) server certificate authority file (CA file) to use. Default is */etc/ssl/certs/bar-ca.pem*.

`--server-cert-file=<file name>`

Specify TLS (SSL) server certificate file to use. Default is: */etc/ssl/certs/bar-server-cert.pem*.

`--server-key-file=<file name>`

Specify TLS (SSL) server key file to use. Default is: */etc/ssl/private/bar-server-key.pem*.

`--server-password=<password>`

Specify server password.

Note: Use this option with care! Passwords specified on the command line may be stored in the history of your shell, thus can be recovered by displaying the command history.

Note: To prevent brute-force attacks each time the authorization fails an increasing delay time is added for connections coming from that client which effectively slow down incoming connections of that client to defeat the attack. The delay is reset after a successful authorization or automatically after around 30s.

²¹ Connections on the server port may also accept SSL encrypted connections.

`--server-max-connections=<n>`

Specify the max. concurrent connections to the server. The default is 8 concurrent connections.

`--server-mode=master|slave`

Specify the server mode. The server operate either as the master (*master*) or is a slave (*slave*) and is controlled by a master instance.

`--jobs-directory=<path name>`

Specify jobs directory. Default is: */etc/bar/jobs*.

`--nice-level=<n>`

Specify the nice-level the server should run. On Linux the nice level may be 0 (not nice) to 19 (very nice).

`--max-band-width=(<n>[M|K])<file name> [<time/date string>]`

Specify the max. band width in bits per second (bits/s) to use to transmit archive files to/from a remote server. If the measured band width is above the specified limit *BAR* will throttle the transmission.

The limit can either be specified directly as a number (with an optional unit) or as an external file. If the name of an external file is specified the first non-empty and non-commented line in that file is read and parsed (with an optional unit) as the band width limit to use. Commented lines can have the prefix *//*, *#* or *;*.

The optional *<date/time string>* define the start date and time for the band width limitation. The date/time string must have the following structure:

<year|>-<month|*>-<day|*> [<week day|*>] <hour|*>:<minute|*>*

with

- year: four digit year
- month: 1..12 or names Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sept, Oct, Nov, Dec
- day: 1..31
- week day names: Mon, Tue, Wed, Thu, Fri, Sat, Sun
- hour: 00..23
- minute: 00..59

A *** mean “do not care” or “always”. If multiple band width limitations are specified the band width limit with the closest matching date/time string before the current date/time will be used.

`--remote-bar-executable=<file name>`

Specify remote executable of *BAR* binary.

Index database

`--index-database=<file name>`

Specify the file name of the internal index database. Default is: */var/lib/bar/index.db*.

--index-database-auto-update

Enable automatically update index database with archives found on the file system or some remote server.

--index-database-max-band-width=(*<n>*[M|K])*<file name>* [*<date/time string>*]

Specify the max. band width in bits per second (bits/s) to use to update the index database via network. If the measured band width is above the specified limit *BAR* will throttle the index update via network.

The limit can either be specified directly as a number (with an optional unit) or as an external file. If the name of an external file is specified the first non-empty and non-commented line in that file is read and parsed (with an optional unit) as the band width limit to use. Commented lines can have the prefix *//*, *#* or *;*.

The limit can either be specified directly as a number (with an optional unit) or as an external file. If the name of an external file is specified the first non-empty and non-commented line in that file is read and parsed (with an optional unit) as the band width limit to use. Commented lines can have the prefix *//*, *#* or *;*.

The optional *<date/time string>* define the start date and time for the band width limitation. The date/time string must have the following structure:

<year|>-<month|*>-<day|*> [*<week day|*>*] *<hour|*>:<minute|*>**

with

- year: four digit year
- month: 1..12 or names Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sept, Oct, Nov, Dec
- day: 1..31
- week day names: Mon, Tue, Wed, Thu, Fri, Sat, Sun
- hour: 00..23
- minute: 00..59

A * mean “do not care” or “always”. If multiple band width limitations are specified the band width limit with the closest matching date/time string before the current date/time will be used.

--index-database-keep-time=*<time>*[weeks|days|h|m|s]

Time to keep index database for not existing storage. If a storage entry does not exists for the specified time, e. g. cannot be found on a remote server, the storage entry is removed from the internal index database. The specified time may have post fixes *weeks*, *days*, *h* (hours), *m* (minutes), or *s* (seconds).

Note: Automatic update index database must be enabled for this option.

--no-index-database

Do not store index database for created archives.

Continuous database

--continuous-database=*<file name>*

Specify the file name of the internal continuous database. Default is empty to keep the database file in memory only.

--continuous-max-size=<size>

Specify the max. size of an entry which should be stored during a continuous job. Entries (files, images, hard links) which size exceed the limit are skipped.

--continuous-min-time-delta=<time>[weeks|days|h|m|s]

Min. time delta for storage of continuous entries. Single entries will only be stored with this time delta when changed. A time of 0 mean store always when changes.

Logging

--log=<name>[,<name>...]

Specify which information is stored in the log file. One or more log information types can specified separated by a comma (.). Table 9: Log information types show the available information log types.

Log type	Description
none	no logging (default)
errors	log errors
warnings	log warnings
OK	log stored/restored files
unknown	log unknown files
skipped	log skipped files
missing	log missing files
incomplete	log incomplete files
excluded	log excluded files
storage	log storage
index	log index operations
all	log everything

Table 9: Log information types

--log-file=<file name>

Specify the log file name.

Note: The log file is only written in daemon mode. See option **--daemon**.

--log-format=<format>

Define the format of a log entry. The log format may contain the macros listed in table 10: Log format macros.

%Y	4 digits year
%m	month 1..12
%d	day 1..31
%H	hours 0..23
%M	minutes 0..59
%S	seconds 0..59
%a	day name
%A	full day name
%b	month name
%B	full month name
%I	hours 0..12
%p	AM or PM
%P	am or pm
%s	number of seconds since 1970-01-01, 00:00:00 (Epoch)
%Z	time zone

Table 10: Log format macros

--log-post-command=<command>

Command to execute as a post-processing command for a log file. This command is executed when a archive was created or restored and can e. g. be a mail command to send the log file via mail.

Available macros: %file.

Miscellaneous

--config=<file name>

Read configuration from the specified file.

--tmp-directory=<path>

Use the specified directory for temporary files.

--max-tmp-size=<size>

Use max. specified size for temporary files. If the space is occupied pause creating new storage files.

Note: only temporary storage files are taken into account!

--max-storage-size=<size>

Specify the max. size in bytes to use for storage. If the storage size is exceeded older archives are deleted.

--max-threads=<n>

Max. number of threads for compress/decompress and encrypt/decrypt entries. Default is number of available CPU cores.

Note: if archives are read via NFS then increasing the number of worker threads may decrease the execution speed due to the NFS read ahead algorithm. Thus use this option with care when running BAR in an NFS environment.

--directory=<path>, -C=<path>

Change to directory.

--jobs-directory=<path name>

Specify jobs directory. Default is: */etc/bar/jobs*.

--incremental-data-directory=<path name>

Specify directory to write incremental data in server mode. Default is: */var/lib/bar*.

--mount=<mount point>[,yes|no]

Define a device to mount resp. unmount before resp. after creating archives. The device is only mounted if it is not already mounted and only unmounted if it was mounted by *BAR* before. If the optional flag “yes” is set then device is unmounted, even it was not mounted by *BAR* before.

--pid-file=<file name>

BAR process identifier file to use. *BAR* write his own process id into this text file.

Note: the file is only created when *BAR* is running as a daemon (server).

--pairing-master-file=<file name>

File to enable or clear pairing of a master with slave. Write *start* into this file or creating an empty will start new pairing of the master, the content *clear* will clear the current paired master.

--pattern-type=glob|regex|extended

Define the pattern type for patterns. glob are simple patterns with * and ? wildcards, regex are regular expressions, extended are extended regular expressions.

--group, -g

Group files when listing archive content.

--all

Show all files, not only the newest version of a file when listing archive content.

--long-format, -L

Use long output format when listing archive content.

--human-format, -H

Print sizes in human readable format (number and unit) instead of bytes.

`--numeric-uid-gid`

Print user/group ids as numbers instead resolved names.

`--numeric-permission`

Print file/directory permissions as octal numbers instead of permission text flags.

`--no-header-footer`

Suppress output of header/footer when listing archive content.

`--delete-old-archive-files`

Delete all old archive files after creating new archive files.

`--ignore-no-backup-file`

Ignore *.nobackup/.NOBACKUP* file and store directories containing this file, too.

`--ignore-no-dump`

Ignore “no dump” file attribute. The “no dump” file attribute can be set for files to prevent them included into a backup. To include these files into the created archive, too, set this option.

Note: on Unix systems see the commands *chattr* and *lsattr* to set and list file attributes.

`--skip-unreadable`

Skip unreadable files instead of reporting an error.

`--force-delta-compression`

Enable forced delta compression. If no delta compression can be used an error is reported instead of a warning only.

`--raw-images`

Store all blocks of a disk in an archive instead of storing only used blocks by the file system.

Note: *BAR* always store all blocks of a disk into the archive. Not used blocks are stored with content “0”. When compression is enabled those blocks almost use no space in the created archive.

`--no-fragments-check`

Do not check if a fragmented file is complete.

`--archive-file-mode=<mode>`

Select archive file write mode. If an archive already exists either stop with an error (*stop*), append to the existing archive (*append*), or overwrite the existing archive (*overwrite*). See also option `--overwrite-archive-files`.

`--overwrite-archive-files, -o`

Overwrite existing archive files instead of reporting an error when an archive file already exists. See also option `--archive-file-mode`.

`--restore-entry-mode=stop|rename|overwrite`

Select mode when restoring entries which already exists. Restore is either stopped with an error (*stop*), an existing entry is renamed (*rename*), on an existing entries is overwritten (*overwrite*).

`--overwrite-files`

Overwrite existing files when restoring files instead of reporting an error.

`--wait-first-volume`

Wait for first volume (CD/DVD/BD or device). If not specified *BAR* assume the first CD/DVD/BD or device volume is already available.

`--dry-run`

Do not compress, encrypt and store archives, restoring files, create/update incremental file lists, or update index database. This option can be used to do a dry-run of a job to see what may be done without actually create archive files or restore something.

`--no-storage`

Do not store archives (skip storage) and skip updating the index database. This option may be useful to create e. g. incremental file lists only with the option *--create* and *--full*.

`--no-bar-on-medium`

Do not store a copy of *BAR* on a CD/DVD/BD medium. By default a copy of the *BAR* executable is stored on a created CD/DVD/BD, too, to be able to restore the files in the archive easily.

`--no-stop-on-error`

Do not store operation when an error occur. Try to continue with remaining restore.

`--no-stop-on-attribute-error`

Do not store operation when an attribute error occur. Try to continue with remaining restore.

Note: some file systems may not support file attributes. Restoring in those file system may fail if this option is not set.

`--no-default-config`

Do not read personal configuration file *~/.bar/bar.cfg*.

`--volume-size=<n>[G|M|K]`

CD/DVD/BD/device volume size. Specified size may have post fixes *G* (1024³bytes), *M* (1024²bytes), or *K* (1024bytes).

Default values are 700MB, 4.4GB, and 25GB for CD/DVD/BD without error correction codes and 560MB, 3.4GB, and 20GB for CD/DVD/BD with error correction codes

`--ecc`

Enable generation of error correction codes when creating CD/DVD/BD or device images. For CDs/DVDs/BDs the external tool *dvdaster* is required.

--always-create-image

Enable always generate an image file for CDs/DVDs/BDs or devices. Usually image files are only generated when error correction codes should be added.

--blank

Blank CD/DVD/BD medium before writing.

--quiet

Be quiet and suppress any output.

--verbose=<n>, -v=<n>

Specify verbosity level. A range of 0..6 can be specified. Default is 1. A higher verbosity level produce more informational output. Table 11: Verbosity levels give an overview what information is printed for the different verbosity levels.

Level	Information printed
0	No additional information is printed
1	Fatal errors
2	All above and some processing information
3	All above and external executed program command lines
4	All above and output (stdout, stderr) of external executed programs
5	All above and output some SSH debug information
6	All above and output all SSH/FTP debug information

Table 11: Verbosity levels

Note: In interactive mode the verbose level is independent from the setting in the BAR configuration files and by default 1.

--version

Output version of *BAR*.

--help, -h

Output help to options.

--xhelp

Output help to extended options.

--help-internal

Output help to internal commands.

BAR configuration and job files

Almost any option setting can be stored in text configuration files. A personal configuration file is read from the users home directory from *.bar/bar.cfg*. If *BAR* is started by root in server mode the configuration file */etc/bar/bar.cfg* is read by *BAR*.

Note: Because the configuration files may contain sensitive data like passwords read access to these files should be limited!

The corresponding configuration file entries to the options

--foo

--foo=<value>

are in general

foo

foo=<value>

If a configuration file entry have a corresponding command line option then this configuration file entry is not described here. Please check the corresponding description in the chapter *BAR* command line options summary, page 43. In the following sections only configuration file entries which does not have a corresponding command line option are listed.

Empty lines and lines starting with # (comments) are ignored.

Additional configuration entries

The following settings may be set additionally in the *BAR* configuration and job files:

UUID=<uuid>

An unique universal identifier for the job. The UUID is created once when the job is created.

archive-name=<name>

Specify the archive name. Usually this configuration entry is only found in job files.

archive-type=normal|full|incremental|differential

Specify the archive type to create. Usually this configuration entry is only found in job files.

crypt-password-mode=default|ask|config

Specify the password mode. Usually this configuration entry is only found in job files.

include-file=<pattern>

Include matching entries into the created archive.

include-file-list=<file name>

Define the name of a file from which names or simple patterns (glob pattern) are read which must match to entries which are included into the created archive.

include-file-command=<command|script>

Define a command or script to execute before the archive is created. The output (stdout) of the command or script define names or simple patterns (glob pattern) which must match to entries

which are included into the created archive.

`include-image=<pattern>`

Include matching devices as disk images into the created archive.

`include-image-list=<file name>`

Define the name of a file from which names or simple patterns (glob pattern) are read which must match to devices which are included into the created archive as disk images.

`include-image-command=<command|script>`

Define a command or script to execute before the archive is created. The output (stdout) of the command or script define names or simple patterns (glob pattern) which must match to devices which are included into the created archive as disk images.

`exclude=<pattern>`

Define a simple file pattern (glob pattern), a regular expression, or a extended regular expression which must match to files which are excluded from the created archive. Default pattern type is simple file pattern. See table 6 how to specify different pattern types.

Note: Exclusions have precedence over include patterns.

`exclude-list=<file name>`

Define the name of a file from which names or simple patterns (glob pattern) are read which must not match to files and are excluded from the created archive.

`exclude-command=<command|script>`

Define a command or script to execute before the archive is created. The output (stdout) of the command or script define names or simple patterns (glob pattern) which must not match to files and are excluded from the created archive.

Note: Exclusions have precedence over include patterns.

Block configuration entries

Some settings, like server, schedule, persistence, or maintenance settings, can be defined multiple times. These settings are prefixed by a block descriptor of the form [*<type> <name>*].

If a server or device is specified by <name> all not explicitly in the URI specified values are read from the configuration entries from a matching block (type and name must match). It is recommended for security reasons and convenience to specify server passwords only in the block configuration entries and limit the read access to the configuration file.

The following sections describe the possible block descriptors and list also the possible configuration entries.

Note: The [...] do not mean “optional”. Instead they are part of the syntax!

Note: The global settings block can be prefixed with [*global*].

File server

File server configuration blocks define the settings for a file server.

```
[file-server <name>]
file-max-storage-size=<size>
file-write-pre-command=<command>
file-write-post-command=<command>
```

Please see chapter File, page 53 for details to the configuration entries.

FTP server

FTP server configuration blocks define the login data for a FTP server.

```
[ftp-server <name>]
ftp-login-name=<name>
ftp-password=<password>
ftp-max-connections=<n>
ftp-max-storage-size=<size>
ftp-write-pre-command=<command>
ftp-write-post-command=<command>
```

Please see chapter FTP, page 53 for details to the configuration entries.

SSH/SCP/SFTP server

SSH/SCP/SFTP server configuration blocks define the login data for a SSH/SCP/SFTP server.

```
[ssh-server<name>]
ssh-port=<n>
ssh-login-name=<name>
ssh-password=<password>
ssh-public-key=<file name>
ssh-private-key=<file name>
ssh-max-connections=<n>
ssh-max-storage-size=<size>
ssh-write-pre-command=<command>
ssh-write-post-command=<command>
```

Please see chapter SSH, page 53 for details to the configuration entries.

WebDAV/WebDAVS server

WebDAV server configuration blocks define the login data for a WebDAV server.

```
[webdav-server<name>]
webdav-login-name=<name>
webdav-password=<password>
webdav-max-connections=<n>
webdav-max-storage-size=<size>
webdav-write-pre-command=<command>
webdav-write-post-command=<command>
```

Please see chapter WebDAV/WebDAVS, page 54 for details to the configuration entries.

Device

Device configuration blocks define the data required to access a generic device.

```
[device <name>]
device-request-volume-command=<command>
device-load-volume-command=<command>
device-unload-volume-command=<command>
device-volume-size=<size>[G|M|K]
device-image-pre-command=<command>
device-image-post-command=<command>
device-image-command=<command>
device-ecc-pre-command=<command>
device-ecc-post-command=<command>
device-ecc-command=<command>
device-blank-command=<command>
device-write-pre-command=<command>
device-write-post-command=<command>
device-write-command=<command>
```

Please see chapter Device, page 57 for details to the configuration entries.

Schedule

Schedule configuration blocks define the data to automate scheduling of jobs.

```
[schedule <name>]
UUID=<uuid>
date = <date>
weekdays = <week days>
time = <time>
archive-type = normal, full, incremental, differential
interval = <n>
text = <text>
no-storage = yes|no
enabled = yes|no
```

The date string must have the following structure:

```
<year|*>-<month|*>-<day|*>
```

The week days string must have the following structure:

```
<day>,...|*
```

The time string must have the following structure:

```
<hour|*>:<minute|*>
```

with

- month names: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sept, Oct, Nov, Dec
- week day names: Mon, Tue, Wed, Thu, Fri, Sat, Sun

A * mean “do not care” or “always”.

Persistence

Persistence configuration blocks define how entities of a job are persisted.

```
[persistence <type>]
min-keep = <n>
max-keep = <n>
max-age = <*|n>
```

Maintenance

Maintenance configuration blocks define maintenance time slots where some maintenance work, like indexing archives or database clean operations are done. These operations may take some time, thus they can be scheduled in time slots. If not maintenance time slots are defined, maintenance work may be done any time.

```
[maintenance]
date = <date>
weekdays = <week days>
begin = <time>
end = <time>
```

The date string must have the following structure:

```
<year|*>.-<month|*>.-<day|*>
```

The week days string must have the following structure:

```
<day>,...|*
```

The time string must have the following structure:

```
<hour|*>.:<minute|*>
```

with

- month names: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sept, Oct, Nov, Dec
- week day names: Mon, Tue, Wed, Thu, Fri, Sat, Sun

A * mean “do not care” or “always”.

Example configuration file

```
# -----
tmp-directory=/home/tmp
max-tmp-size=1G
nice-level = 19
log = errors,warnings,skipped
log-file = /var/log/bar.log
# -----
compress-algorithm = bzip9
compress-min-size = 128
# -----
crypt-algorithm = AES256
crypt-password = "foo"
# -----
server-port = 0
server-password = "foo"
server-jobs-directory = /etc/bar/jobs
# -----
index-database = /var/lib/bar/index.db
```



```

index-database-keep-time = 3days
# -----
ecc = yes
blank = yes
skip-unreadable = yes
overwrite-archive-files = yes
quiet = yes
verbose = 1
# -----
[ftp-server ftp1.org]
ftp-login-name = boell
ftp-password = "Ansichten eines Clowns"
[ftp-server ftp2.org]
ftp-login-name = hesse
ftp-password = "Unterm Rad"
# -----
[ssh-server gegen-das-vergessen.org]
ssh-port = 1234
ssh-login-name = anne
ssh-public-key = /home/foo/.ssh/id_rsa.pub
ssh-private-key = /home/foo/.ssh/id_rsa
ssh-password = "Anne Frank Tagebuch"
[ssh-server franz.kafka.org]
ssh-port = 1122
ssh-login-name = kafka
ssh-public-key = /home/kafka/.ssh/id_rsa.pub
ssh-private-key = /home/kafka/.ssh/id_rsa
ssh-password = "Die Verwandlung"
# -----
[webdav-server heinrich.boell.org]
webdav-port = 1122
webdav-login-name = boell
webdav-password = "Ansichten eines Clowns"
# -----
[device /dev/tape]
volume-size = 2T
request-volume-command = "echo Please insert new tape!|mail -s 'Tape control' root"
unload-volume-command = eject %device
load-volume-command = eject -t %device
device-write-command = dd if=%file of=%device

```

BARControl command line options summary

The following sections list the *BARControl* command line options.

--port=<n>, -p=<n>²²

Server port. Default is:: 38523.

--tls-port=<n>

TLS server port. Default is: 38524.

--password=<password>

Note: Use this option with care! Passwords specified on the command line may be stored in the history of your shell, thus can be recovered by displaying the command history.

--login-dialog

Force to open the login dialog.

²² A SSL encrypted connection may also be established on the plain server port.

--ca-file=<file name>

Server certificate authority file (PEM format, CA file). Default is: *./bar-ca.pem* or */home/<HOME>/bar/bar-ca.pem* or */etc/bar/bar-ca.pem*.

--cert-file=<file name>

Server certificate file (PEM format). Default is: *./bar-server-cert.pem* or */home/<HOME>/bar/bar-server-cert.pem* or */etc/bar/bar-server-cert.pem*.

--key-file=<file name>

Key file name (PEM format). Default is: *./bar.pem* or */home/<HOME>/bar/bar.pem* or */etc/bar/bar.pem*.

--force-ssl

Force SSL connection

--select-job=<name>

Select a job in the status tab.

--job=<name>, -j=<name>

Start execution of job <name> and terminate *BARControl*.

--archive-type=<mode>

Set job mode: normal, full, incremental, differential. Default is: normal.

--abort=<name>

Abort execution of job <name> and terminate *BARControl*.

--ping, -i

Check connection to server and terminate *BARControl*.

--pause=<n>, -t=<n>

Pause job execution for <n> seconds and terminate *BARControl*.

--suspend, -s

Suspend job execution and terminate *BARControl*.

--continue, -c

Continue job execution and terminate *BARControl*.

--list, -l

List jobs and terminate *BARControl*.

--index-database-add=<pattern>

Add storage archive to the index database and terminate *BARControl*. If <pattern> is a directory all containing files with the suffix *.bar* are added to the index database.

--index-database-remove=<pattern>

Remove storage archive from the index database and terminate *BARControl*.

--index-database-refresh=<pattern>

Refresh storage archives in the index database and terminate *BARControl*.

--index-database-entities-list=<name>, -n=<name>

List entities which match to the names <names> and terminate *BARControl*.

--index-database-storages-list=<name>, -a=<names>

List storage archives which match to the names <names> and terminate *BARControl*.

--index-database-entries-list=<pattern>, -e=<name>

List index entries which match to the names <name> and terminate *BARControl*.

--maintenance=<n>

Intermediate maintenance time for <n> minutes.

--restore=<name>

Restore storage with name <name>

--destination=<directory>

Selected destination directory for restore.

--overwrite-entries

Enable overwrite existing entries on restore.

--role=basic|normal|expert

Select role of user. A *basic* user may only see the most important values, a *normal* user see all important values, and a *expert* user see all values.

--version

Output version of *BARControl*.

--help, -h

Output help to options.

***BARControl* configuration file**

BARControl also store a few settings into a personal configuration file in *~/bar/barcontrol.cfg*. Some entries in the file are updated by *BARControl*. Entries which may be edited are the server settings.

The file may contain the following configuration entries:

pause-create=yes|no

Pause creating new archive files when pause-button is pressed.

pause-storage=yes|no

Pause storage of archive files when pause-button is pressed.

pause-restore=yes|no

Pause restoring archive files when pause-button is pressed.

pause-index-database-update=yes|no

Pause automatic update index database when pause-button is pressed.

server=<name>[:<port>[:<password>]]

Name, optional port number and optional password of computer running a *BAR* server daemon.

server-ca-file=<filename>

BAR server certify authority key file name.

server-cert-file=<filename>

BAR server certificate key file name.

server-key-file=<filename>

BAR server key file name.

force-ssl=yes|no

Force TLS protected connection to BAR server.

role=basic|normal|export

User role to use.

Exit codes

BAR generate the exit codes shown in table 12: BAR exit codes when terminating under some conditions.

Exit code	Description
0	Normal termination. No error.
1	Program execution fail.
5	Invalid or wrong number of arguments.
6	Configuration error.
125	Initialization fail.
126	Fatal error occurred.
127	Function still not supported.
≥ 128	System error. Error number is <code>-128.

Table 12: BAR exit codes

Archive file format

The *BAR* archive format is a file format only related to *BAR*. It is organized in so called *chunks*. Each archive file consists of a sequences of chunks. Chunks can contain sub-chunks which have the same structure. All numerical non-data entries are stored in big endian format. Encrypted data is padded with 0-bytes to a multiple-size of the used encryption block length (see table 8: Encryption algorithms, page 51) for the block length of encryption algorithms).

The following data types are used:

uint8	unsigned 8 bit integer
uint16	unsigned 16 bit integer
uint32	unsigned 32 bit integer
uint64	unsigned 64 bit integer
int8	signed 8 bit integer
int16	signed 16 bit integer
int32	signed 32 bit integer
int64	signed 64 bit integer
string	string: uint16 length field, following string character data
<type>[]	array: uint16 length field, following data of specified type
crc32	unsigned 32 bit containing the CRC32 sum of the entry
data	arbitrary data

A single chunk have the structure:

- id (4 characters)
- size (uint64)
- data (n bytes)

The chunk id is unique and specify how the data of the chunk must be interpreted. The size value give the size of the chunk in bytes excluding the header (id and size). All chunk data except the chunk id, and size may be encrypted. Only data entries may become compressed.

Note: The file *bar/archive_format.def* contain the formal specification of all archive chunk types. This specification is compiled in to C source code with the perl-script *bar/archive_format.pl*.

The chunk format allow extension of the archive format in future development and is more robust when parts of an archive are destroyed. E. g. if a chunk is destroyed *BAR* can skip that chunk, search for the next one and can read the rest of the archive.

Compilation

To compile *BAR* by yourself you need a *gcc* compilation environment, *make* and some libraries. The following list the required packages to compile *BAR* including all features. If some package is not available, support of some features may be disabled by the *configure* script. If a mandatory package is missing *configure* will output an error:

Package	Requirement	Comment
---------	-------------	---------

gcc	mandatory	C and C++ compiler
openjdk-6-jdk ²³	mandatory	Java development kit
make	mandatory	make program (GNU compatible)
m4	mandatory	m4 macro language
xgettext	mandatory	translation tool
tclsh	mandatory	tcl shell
zlib1g	mandatory	zip compression library
zlib1g-dev	mandatory	zip compression library
libbz2-1.0	optional	bzip compression library
libbz2-dev	optional	bzip compression library
liblzma	optional	lzma compression library
liblzma-dev	optional	lzma compression library
liblzo2-2	optional	lzo compression library
liblzo2-dev	optional	lzo compression library
lz4 ²⁴	optional	lz4 compression library
zstd ²⁵	optional	Zstandard compression library
libxdelta2	optional	xdelta compression library
libxdelta2-dev	optional	xdelta compression library
libssl0.9.8	optional	SSL/TLS library
libssl-dev	optional	SSL/TLS library
openssl	optional	SSL library
openssl-devel	optional	SSL library
libssh2-1	optional	ssh library
libssh2-1-dev	optional	ssh library
libgcrypt11	optional	encryption library
libgcrypt11-dev	optional	encryption library
nettle	optional	nettle library
gmp	optional	GMP library
gnutls-bin	optional	TLS tools
libgnutls26	optional	TLS library
libgnutls-dev	optional	TLS library
libcurl	optional	curl library
c-ares	optional	asynchronous DNS request library

²³ Instead of the OpenJDK package a standard JDK package may be usable, too.

²⁴ lz4 may only be available from the source repository. See [LZ4].

²⁵ Zstandard may only be available from the source repository. See [Zstandard].

mxml ²⁶	optional	mini XML library
libiso9660	optional	ISO 9660 read library
libiso9660-dev	optional	ISO 9660 read library
libicu	optional	i18n library
epm	optional	package creating tool (only needed when rpm/deb/pkg package should be created)
ant, launch4j	optional	tool to pack <i>BARControl</i> into a single-binary (only needed when a single-binary exe-file should be created on Windows) ²⁷

Table 13: Compilation packages

To compile *BAR* first run *configure* with the command sequence:

./configure

If all features are available with the default settings *configure* may show on a Linux 64bit system the following output at the end:

```
Configuration 0.21:
Host           : x86_64-unknown-linux-gnu
Dynamic linked : no
SQLite         : yes (3.21.0)
zip compression : yes (1.2.11)
bzip2 compression : yes (1.0.6, 6-Sept-2010)
lzma compression : yes (5.2.4)
lzo compression  : yes (2.09)
lz4 compression  : yes (1.7.1)
zstd compression : yes (1.3.2)
xdelta compression: yes
crypto support   : yes (1.8.4)
FTP support      : yes
curl support     : yes (7.62.0)
SCP/SFTP support : yes (1.8.0)
TLS/SSL server   : yes (OpenSSL 1.1.1 11 Sep 2018, GMP 6.1.2, GnuTLS 3.5.19)
ISO 9660 support : yes (2.0.0 x86_64-pc-linux-gnu)
PCRE support     : yes (8.40 2017-01-11)
ICU support      : yes (58.2)
GUI support      : yes
GUI i18n support : yes
SWT directory    :

Debug version    : no
Crash dump support: no
Stacktrace support: yes

Packages support : no

Install directory : /usr
Binary directory  : /usr/bin
Config directory  : /etc/bar
Runtime directory : /var/lib/bar
TLS directory     : /etc/ssl

Heapview diretory :
```

Note: you can download missing additional packages with the script '*./download-third-party-packages.sh*'. Please rerun *configure* after downloading.

²⁶ mxml is required when curl is used for WebDAV access.

²⁷ Currently ant/launch4j is not fully supported in the *BAR* make file.

To compile *BAR* enter the command:

make

This will create the *BAR* binaries on your system.

It is highly recommended to run the test suite included in the *BAR* sources. To execute all standard tests enter the command:

make test

This will execute several tests to create, list, test, compare, and extract archives with some test data and compression, encryption, and split settings. The following test runs are available:

test1	basic tests
test2	compress tests
test3	encryption tests
test4	asymmetric encryption tests
test5	split tests
test6	image tests
test7	storage tests
test8	large file tests
test9	DVD tests
test10	server tests
test_combined	combined compress, encryption, and split tests
test_parameters	test command line parameters
test_misc	miscellaneous tests
test	execute <i>test1..test6</i> , <i>test8</i> , <i>test_combined</i> , <i>test_parameters</i> and <i>test_misc</i>

Table 14: Tests

It is also possible to execute single test sets. Then also some additional output is visible.

For the image tests the Linux loop device system is required and some commands must be executed as *root*. This is done via *sudo*²⁸. Therefore to execute the image tests successfully some *sudo* settings are required. Please see the *README* file for an example.

For the FTP storage tests a FTP server with login credentials and write access for the current user must run on the local computer.

For the SCP and SFTP storage tests a SSH server with login credentials and write access for the current user must run on the local computer.

For the WebDAV storage tests a https server with WebDAV support, login credentials and write access for the current user must run on the local computer.

For the DVD storage tests a DVD writer and an empty DVD or an empty DVD RW must be available.

Note: A complete test run may take some time. There are a lot of tests done. Please be patient.

²⁸ *sudo*: Unix command to execute a command as a different user. See e. g. <http://en.wikipedia.org/wiki/Sudo>.

After a successful test run you can install *BAR* on your system with the command:

make install

Note: This will **not** use the package manager of your system. The installation is done under */usr/local* and */etc/bar* in your system (if not configured different with the *configure* script).

For additional make targets enter the command:

make help

Development

The *BAR* sources are currently available as a TAR archive only on:

<http://www.kigen.de/projects/bar>

If you want to contribute to *BAR*, you can send me patches via mail.

Frequently asked questions

If you use *BAR* or recompiled *BAR* by yourself and you run into some problem, please check the following questions and answers list before creating a bug report. Some problems are known and are not caused by *BAR*.

1. *Why is BAR linked static by default?*

BAR is not fully linked static, but partially. All needed libraries are linked static except *libc*, *libm*, *libdl*, *librt* and *libpthread*. The advantage of this type of linkage is that *BAR* can be executed on many systems, even when some specific libraries like libraries for compression or encryption are not available on the target system. *BAR* can also be started from a CD or DVD without installation. If a dynamic linked *BAR* version is required set the configure option *--enable-link-dynamic*.

2. *Can I build a dynamic linked version of BAR?*

Yes. Use *--enable-link-dynamic* when calling configure. The resulting binary is linked dynamic.

7. *Can I create incremental archives even the file system does not have an "archive" bit?*

Yes. *BAR* create his own incremental list files (*.bid* files). These list files contain information which are used to check which files changed.

8. *Which compression algorithm should I use?*

A good choice for the compression algorithm depend on the type of data to compress and how many time and memory you may spend to let *BAR* compress the data. In general *lzma* compress better than *bzip2* than *zip*. *lzo*, *lz4*, and *zstd* are faster compression algorithms with the expense of reduced compression results. A higher compression level result in general in a better compression result. On the other side in general a better compression require more time and memory. Though if time and memory consumption is not critical *lzma9* is a good choice for not already compressed data. If time and memory is critical choose e. g. *lz4-0*, *zstd-0* or *zip1*.

9. *Can I just create the incremental list file?*

Yes. Use option `--no-storage`. The incremental list file is created, but no archive files are neither created nor stored.

10. *I cannot connect with BARControl to the running BAR server.*

If there is a firewall program running the server ports must be opened to be able to access the *BAR* server from outside. See chapter Firewall, p. 20.

11. *I cannot establish a TLS/SSL connection with BARControl.*

For the TLS/SSL connection in *BARControl* a TLS/SSL key is required. Create that key once with "make keys" and copy the keys with "make install_keys" in the appropriated directories of your system or use the command *bar-keygen*.

12. *I only want to allow TLS/SSL-connection to the BAR server. How can I forbid non-encrypted connections?*

If you want to prohibit non-encrypted connections to the *BAR* server, specify the port number 0 for the option *server-port* in the configuration file *bar.cfg*. This disable the plain server connection port in *BAR*.

13. *I tried to list a *.bid-file, but it does not show anything useful.*

The *.bid*-files are not archive files, instead these are files with lists of file names needed for an incremental backup. Do not modify, delete or copy those files. The content is not encrypted. *BAR* does not store them on an external server or CD/DVD/BD.

14. *When I specify -# /foo/ the list of included files is different to -# /foo. Why?*

The option values for `-#` and `-!` are patterns, not file or directory names. `-# /foo/` match to everything which starts with */foo/* including the trailing `/`. If the directory */foo* contain sub-directories only those sub-directories are included. If you specify `-# /foo` the directory "foo" including all sub-directories and all files in `/` starting with "foo" are included.

15. *I get the error "configure: error: function zlibVersion() is not available".*

The zlib library [zlib] is mandatory for compilation. Please install the zlib package (binary and development files) with your distribution package tools or use the script *download-third-party-packages.sh* included in the *BAR* sources to download a suitable zlib library source package.

16. *Instead of the SWT JAR included in BAR I like to use another one I already have. BARControl is starting, but some of the buttons does not work properly. What is wrong?*

This is probably a bug in the SWT JAR you are using in combination with the installed GTK libraries on your system. Probably SWT 3.5.x and GTK 2.18 or newer versions seems not to work together anymore. I could avoid this problem by using SWT 3.6 or newer.

17. *Why is BAR using his own archive format and not some already existing one?*

BAR is using his own (chunk based) archive format, because features like encryption and splitting cannot be handled very well by other formats. Another design decision for a new format is that it should be more robust when there is an read-error.

18. *Is the BAR archive format robust when there are read-errors or something is lost?*

Yes. The archive format of *BAR* is based on chunks (see also Archive file format, page 77). If some part of an archive is destroyed and cannot be read anymore, e. g. read-error on a CD, the rest of the archive which is following and which is still readable can be processed by *BAR*. Also if some part of a split archive is missing the other still existing parts can be read independently. *BAR* can partially restore files, even when archives are broken or parts got lost.

19. *When I create an archive, BAR append a 6-digit number to the archive name. Why is BAR doing this and how can I avoid it?*

When you enabled to create split archives and there is not a single # in the specified archive name, *BAR* append internally ##### to the archive name for the part number. The result is a continuous increasing 6-digit-number appended to the name.

To avoid appending the number either disable splitting archives (see Splitting, page 52) or place some # into the archive file name where you want to have the generated part number.

20. *When I compile BAR with the libgcrypt library installed my system, the private key file cannot be read. I get the “Unable to initialize private key file from file”. Whats wrong?*

I also saw this problem and I assume it is a bug in some libgcrypt versions. When I use libssh2 with the compiled-in libgcrypt-functions and OpenSSL (configure options `--without-libgcrypt` and `--with-openssl`) this error disappears.

21. *When I try to use the 3DES encryption algorithm, I always get the error message “init cipher fail”. What is going wrong?*

When the 3DES encryption algorithm is used the pass phrase must have a minimal length. If a shorter pass phrase is used the *gcrypt* library report the error “init cipher fail”. To solve this problem, just specify a pass phrase which has a length of at least 17 characters.

22. *When I try to create an archive on a remote server by using the scp protocol I always get the error message “ERROR: Cannot store file 'scp://.../<path>/<file name>' (error: failed to send file)”. Why can the file not be stored?*

Please make sure the specified path *<path>* exists on the remote server and can be accessed by *BAR*. The scp protocol cannot create remote directories automatically and if a file cannot be written because either the path does not exists or there are insufficient permissions to create the file, scp return the error “failed to send file”.

23. *When I try to create an archive on a remote server by using the SCP or SFTP protocol BAR try to open a public key file even I use not asymmetric encryption. I see the error message “ERROR: Cannot initialize storage '...' (error: SSH authentication fail: Unable to open public key file)”. What key file is BAR asking for?*

To establish a ssh connection for either SCP or SFTP protocol *BAR* may need the private and public ssh keys to login on the ssh server. These ssh keys are usually stored in some file in *\$HOME/.ssh*. The files may be specified with the command line options `--ssh-public-key` and `--ssh-private-key` or in */etc/bar/bar.cfg* with the entries *ssh-public-key* and *ssh-private-key*.

These key files are different to the public/private key file pair used for asymmetric encryption of the archives. The ssh-key files are used by the ssh protocol to encrypt the network connection data stream while the asymmetric encryption keys are used to encrypt the archive file content.

24. *When I try to create an archive on a remote server by using the SCP or SFTP protocol BAR report the error message “ERROR: SSH authentication fail: Callback returned error” even I have specified the required ssh-key files. Why does ssh authentication still do not work?*

The error message is coming from the ssh protocol stack and probably indicate that the key files cannot be read e. g. because there is a password required. If your ssh-keys are password protected (which is recommended!), please make sure you have the password specified in `/etc/bar/bar.cfg` with the entry `ssh-password` besides the key configuration entries `ssh-public-key` and `ssh-private-key`.

Note: make sure the access rights for the configuration file `/etc/bar/bar.cfg` is limited when you add passwords there!

25. *When I compile a static linked version of BAR, I get linker errors for some libraries. What is wrong?*

Maybe there is no static version of some external library available on your system. Try to download the required libraries with the script `download-third-party-packages.sh`. These libraries are linked static.

26. *When I create the key files with bar-keygen it takes a very long time. Why is this so slow?*

If some specific version of `certtool` is used to generate the key files it may use `/dev/random` instead of `/dev/urandom` to get random data for generating the keys. Reading `/dev/random` may block when there is not enough entropy to generate new random numbers. This slow down the key generation. Try to install OpenSSL²⁹ and use the option `--openssl` when calling `bar-keygen`.

27. *I like to use BAR on my system, but I don't know if it is compatible. How can I verify this?*

Check if there is a pre-compiled package available for your system. Usually pre-compiled packages for an older system version will work on newer versions, too. If you are using a system which is currently not listed at all on the web-site, send me an email with information about your system (operating system, version, etc.). Maybe I can compile a package for your system.

28. *I like to use BAR to backup some remote folder, but I do not like to use the full client-server-mode. How can this be done?*

You can execute `BAR` on the remote computer to store the folder into an archive and then let `BAR` transmit this archive to your local computer. For this you must have either a FTP server (not recommended, because the connection is not encrypted!) or a SSH server (recommended) running on your local system. The user which executes `BAR` on the remote system must be able to login on your local system. If this is working, you can specify the local system as the archive destination when execution `BAR` on the remote system, e. g. with via FTP `“ftp://NAME@LOCAL_MACHINE_NAME/foo.bar”` or via SSH `“scp://NAME@LOCAL_MACHINE_NAME/foo.bar”`.

Note: you may store FTP or SSH server data like login names and passwords as well as key file names in your configuration file `~/bar/bar.cfg` or `/etc/bar/bar.cfg` to avoid to reveal sensitive information on the command line.

29 See <http://www.openssl.org>

29. *Can I store BAR archives on a Samba³⁰ server?*

Yes. Just mount the destination directory of the Samba server where you want to store the archive files and use file system storage of *BAR* (see File system, page 17).

30. *BARControl does not start on my system. I get the error “No more handles [gtk_init_check() failed]”. What is wrong?*

Probably *BARControl* cannot connect to the running X11 server. Either the environment variable *DISPLAY* is not set correctly or connections to the X11 server are not allowed. Please check the variable and execute *xhost <host name>* to allow an X11 server connection for *<host name>*. Further details to this problem can be found here:

http://www.eclipse.org/tptp/home/downloads/releasenotes/releasenotes4_3_0.html

31. *BARControl ask me for a password I do not know? What is that password for and were do I define it?*

There are no default passwords in *BAR*. Depending on the operation which require the password, the password *BAR* is asking for may be either a login password needed for FTP or SSH or it is the encryption password for an archive.

The FTP or SSH passwords are defined in the FTP or SSH server resp. SSH keys you are using. *BAR* may ask for this password with the text output “FTP login password” resp. “SSH login password”. Please check the configuration of the FTP or SSH server and the password you have used when you created the SSH key files.

The encryption password for an archive can be selected by you when you create the archive. The same password is required to read the created archive later again. There you see the text output “Crypt password for”.

If you start *BARControl* then the password to login on a (remote) running *BAR* is required. This password is defined in *bar.cfg* under “server-password”.

32. *Can I automate the login with BARControl?*

Yes. You can store the login name and password in your personal *BARControl* configuration file *~/.bar/barcontrol.cfg* under “server-password”. By default *BARControl* first try the password stored in *~/.bar/barcontrol.cfg*. If this is failing, the login dialog is presented.

Note: Because *~/.bar/barcontrol.cfg* may contain sensitive information, it is recommended to limit read access to this file.

33. *BAR server is running with the Linux root account on my system. Is it required that BARControl is also running as root?*

No, this is not required. Any user account can be used for starting *BARControl* and to connect to the running *BAR* daemon. Authorization is done via the *BAR* server login password. If access to the *BAR* server should be limited, then set a login password in */etc/bar/bar.cfg* via the entry *server-password*.

Note: If no server password is set, *BAR* print a warning on startup.

34. *When I created an encrypted archive and I restore it via BARControl, BARControl is not asking for the encryption pass phrase, but BAR can read the archive. Why is BARControl not asking for the pass phrase?*

When *BAR* (in daemon mode) is requested by *BARControl* to create an encrypted archive the

30 See <http://samba.org>.

pass phrase is cached internally in *BAR*³¹. If you immediately restore the encrypted archive the required pass phrase for reading the archive is read from the internal cache, thus *BARControl* is not asking for the pass phrase. If the pass phrase is unknown, *BARControl* will ask for the it.

Note: The cache is cleared when the *BAR* daemon is restarted.

35. *Sometimes when I create an archive I see in the progress bars of BARControl numbers over 100%. How can this happen?*

The numbers shown in the progress bars of *BARControl* can become higher than 100% when the number of files or the file sizes which are currently stored into the archive changes in the meantime. *BAR* first collect in a pre-processing step information how many files and how many bytes will be stored into the archive. If files are modified while the archive is finally created, the number of files or the sizes may be more than detected in the pre-processing step, thus result in more than 100% of stored files or bytes.

Note: This is not a bug. The progress bars can only give a estimation if files are changing while stored into an archive.

36. *When I try to create a backup on DVD with BAR I always get the error message "Is a directory: ..." in barcontrol. What is going wrong?*

Please check the settings for the DVD device to use in the "Jobs" tab under "Storage". You can leave this field empty for using the default DVD device of your system or you can specify the DVD device, e. g. `/dev/dvd`, to use for writing on the DVD.

37. *I tried to list the content of CD/DVD/BD with a specific device name, but BAR always report the error "file not found".*

Make sure the device name is terminated by a ":", e. g.

bar dvd:///dev/sr1:

38. *How can I display the content of my SSL/TLS server certificate?*

This can be done with e. g. the *openssl* command:

openssl x509 -text -in /etc/ssl/certs/bar-server-cert.pem

39. *How can I renew a SSL/TLS server certificate when it is expired?*

This can be done by creating new key and certificate files with the command

bar-keygen

Just execute it and copy the generated files on the server where BAR is running.

40. *Is the graphical front end available as a separated package?*

Yes. Download *bar-gui-<version>.zip*. The archives contain the compiled *BARControl* graphical front end for Linux, Windows, Solaris, and MacOSX.

31 When the gcrypt library support secure memory, such memory is used for storing pass phrases in memory at run time. This make sure pass phrases are not unintended written to the hard disk e. g. into a virtual memory swap file or when a memory dump is created. If secure memory is not available, *BAR* use a least a simple XOR obfuscation to avoid plain text pass phrases in RAM.

41. *I cannot establish a direct connection to the BAR server running on a remote computer, because of firewall restrictions. Can I use a SSH tunnel to connect to the BAR server?*

Yes. You can use a SSH tunnel to forward the network ports used to communicate with the BAR server on the remote computer. You may create a SSH tunnel for the default ports like this:

```
ssh -L38523:localhost:38523 <user>@<host>
```

```
ssh -L38524:localhost:38524 <user>@<host>
```

where <user> is the SSH login user name and <host> is the remote computer running the BAR server.

42. *Sometimes when I want to connect the BAR server with BARcontrol I just get ERROR communication: Network error on...: read time out.*

If the connection to the BAR server stop with this error during the authorization it may happen that before from the same client several authorization tries already failed. Each time an authorization fail, BAR add an increasing short delay to slow down incoming connections from that client to prevent brute-force attacks. The delay is reseted after a successful authorization or around 30s automatically.

43. *I want to start BAR as a daemon on Linux at boot time automatically. How can this be done?*

You can use the supplied script `/etc/init.d/barserver` to start BAR as a server at boot time. Edit in the run-level editor of your Linux distribution the entry for barserver to start barserver e. g. in run-levels 3 and 5. Then BAR is started as a daemon when the computer is booted.

You can also start and stop the BAR daemon manually by calling `/etc/init.d/barserver` with the arguments *start* or *stop*.

44. *A full backup is expired and was purged. Is it useful to keep the following incremental backups?*

When a full backup was purged because its lifetime expired the following incremental backup archives may still be used as partial backups if they are not compressed by the delta-algorithm. Each backup archive can be read independently and does contain specific versions of the files. Though you may keep the incremental backup archives until they expire, too, and become purged.

Note: if the delta-compress-algorithm is used the original file for the delta-compression must still exist to extract the delta-compressed files, because the archive only contains the differences to the original files.

45. *How can I use BAR on computer with tightly limited resources? I always get the error insufficient disk space when BAR try to create archives.*

If you want to run BAR on a computer with tightly limited resources, e. g. a Live Linux system where no hard disk for creating the temporary files is available you can instruct BAR to use a RAM disk for the temporary files and limit the used temporary space.

Create a RAM disk, e. g.:

```
mount -t tmpfs -o size=2G none /media/ramdisk
```

Call BAR with the options:

```
--max-threads=2 -s 256M --tmp-directory=/media/ramdisk--max-tmp-size=1G
```

46. *Using more threads when testing or comparing an archive reduce the execution speed instead of increasing it. Why is parallel testing/comparing slower than doing the same with a single thread only?*

The effect may be related to several issues of the system environment. It is not related that parallel processing is slower. Instead it is faster, but there are conditions:

if an archive is read for testing or comparing via NFS the random read operations of the multi-threaded run may cause a slow down due to caching and read ahead issues of NFS and the used algorithm. E. g. with a Linux kernel 2.6 using `--max-threads=<number of cores>` may be much slower than using `--max-threads=1`. See this paper for detailed information <https://www.fsl.cs.sunysb.edu/docs/nfs4perf/nfs4perf-microscope.pdf>.

Contact and bug reports

If you find a bug in *BAR*, please try to reproduce the bug with *bar-debug*. This binary include some additional debug code. Please send me a bug report including the following information:

- version number of *BAR*
- description of your system environment (Linux distribution and version, CPU information)
- if you used a pre-compiled version and which package or a self-compiled source-version
- steps to trigger the bug
- what is going wrong

With *bar-debug* you can also get some information of the internal state while *BAR* is running by pressing `Ctrl-\` in the shell. Alternatively you can send the signal `SIGQUIT` to the running *BAR* instance with the command

```
killall -QUIT bar-debug
```

This output stack traces of all running threads and locking information to semaphores. Please send me this information, too.

Note: It is recommended to run *bar-debug* not in the background (use command line options `-D` and `-daemon`).

In case of a fatal error where *BAR* is crashing, please send me the created *BAR* crash dump file *bar.dump*³² and the printed C³³ or a Java stack trace. To get a C stack trace you must use *bar-debug* or start *BAR* inside a debugger like *gdb*. With *gdb* you can get a C stack trace after the crash with the command *bt*.

In case of a fatal error where *BARControl* is crashing, please start the program with the option `--debug`. With this option a Java stack trace should be printed with detailed information of the crash.

Please send the report and files to:

- torsten.rupp@gmx.net

I will try to reproduce the problem and fix it as soon as possible.

License

BAR and all included files are under the GPL version 2. The full GPL version 2 license text can be found here:

- <http://www.gnu.org/licenses/gpl-2.0.html>

BARControl is using SWT [SWT]. SWT is under the "Eclipse Public License" which can be found here:

- <http://www.eclipse.org/org/documents/epl-v10.php>

³² The file *bar.dump* is a simple TAR file containing a crash mini-dump, a symbol table file, the operation system information and the CPU information.

³³ If a dump file and a C stack trace is created depend on the system. This feature is not available on all systems.

References

ant	http://ant.apache.org/manual
binutils	http://ftp.gnu.org/gnu/binutils
bzip2	http://www.bzip.org
c-ares	http://c-ares.haxx.se
cdrtools	http://cdrecord.berlios.de/old/private/mkisofs.html
dvd+rw-tools	http://fy.chalmers.se/~appro/linux/DVD+RW
dvdisaster	http://dvdisaster.net/en/index.html
epm	http://www.epmhome.org
nettle	http://www.lysator.liu.se/~nisse/nettle
gmp	http://www.gmpilib.org
gnutls	http://www.gnu.org/software/gnutls
growisofs	http://fy.chalmers.se/~appro/linux/DVD+RW
ICU	http://site.icu-project.org/
JRE, JDK	http://www.java.com
launch4j	http://launch4j.sourceforge.net
libcurl	http://curl.haxx.se
libcdio	www.gnu.org/software/libcdio/libcdio.html
libgcrypt	http://directory.fsf.org/project/libgcrypt
libiconv	http://www.gnu.org/software/libiconv
libssh2	http://www.libssh2.org
LZMA	http://tukaani.org/lzma
LZO	http://www.oberhumer.com/opensource/lzo
LZ4	https://code.google.com/p/lz4
Zstandard	https://facebook.github.io/zstd/
mxml	http://www.minixml.org
OpenSSL	http://www.openssl.org
pcre	http://www.pcre.org
SQLite	http://www.sqlite.org
SWT	http://www.eclipse.org/swt
xdelta	http://xdelta.org
zlib	http://www.zlib.net

Alphabetical Index

--abort.....	74	--continue.....	74
--all.....	64	--continuous-database.....	61
--always-create-image.....	67	--continuous-max-size.....	62
--archive-file-mode.....	65	--continuous-min-time-delta.....	62
--archive-part-size.....	52	--convert.....	44
--archive-type.....	74	--create.....	9, 43
--bd-blank-command.....	56	--crypt-algorithm.....	50
--bd-device.....	55	--crypt-new-password.....	51
--bd-ecc-command.....	56	--crypt-password.....	51
--bd-ecc-post-command.....	56	--crypt-private-key.....	52
--bd-ecc-pre-command.....	56	--crypt-public-key.....	51
--bd-image-command.....	56	--crypt-type.....	51
--bd-image-post-command.....	56	--daemon.....	59
--bd-image-pre-command.....	55	--delete-old-archive-files.....	65
--bd-load-volume-command.....	55	--delta-source.....	50
--bd-request-volume-command.....	55	--destination.....	59, 75
--bd-unload-volume-command.....	55	--device.....	57
--bd-volume-size.....	55	--device-blank-command.....	58
--bd-write-command.....	56	--device-ecc-command.....	58
--bd-write-image-command.....	57	--device-ecc-post-command.....	58
--bd-write-post-command.....	56	--device-ecc-pre-command.....	58
--bd-write-pre-command.....	56	--device-image-command.....	57
--blank.....	67	--device-image-post-command.....	57
--ca-file.....	74	--device-image-pre-command.....	57
--cd-device.....	55	--device-load-volume-command.....	57
--cd-ecc-command.....	56	--device-request-volume-command.....	57
--cd-ecc-post-command.....	56	--device-unload-volume-command.....	57
--cd-ecc-pre-command.....	56	--device-volume-size.....	57
--cd-image-command.....	56	--device-write-command.....	58
--cd-image-post-command.....	56	--device-write-post-command.....	58
--cd-image-pre-command.....	55	--device-write-pre-command.....	58
--cd-load-volume-command.....	55	--differential.....	47
--cd-request-volume-command.....	55	--directory.....	64
--cd-unload-volume-command.....	55	--directory-strip.....	58
--cd-volume-size.....	55	--dry-run.....	66
--cd-write-command.....	56	--dvd-blank-command.....	56
--cd-write-image-command.....	57	--dvd-device.....	55
--cd-write-post-command.....	56	--dvd-ecc-command.....	56
--cd-write-pre-command.....	56	--dvd-ecc-post-command.....	56
--cd-blank-command.....	56	--dvd-ecc-pre-command.....	56
--cert-file.....	74	--dvd-image-command.....	56
--comment.....	45	--dvd-image-post-command.....	56
--compare.....	11, 43	--dvd-image-pre-command.....	55
--compress-algorithm.....	14, 47	--dvd-load-volume-command.....	55
--compress-exclude.....	50	--dvd-request-volume-command.....	55
--compress-min-size.....	50	--dvd-unload-volume-command.....	55
--config.....	63	--dvd-volume-size.....	55

--dvd-write-command.....	56	--info.....	45
--dvd-write-image-command.....	57	--job.....	74
--dvd-write-post-command.....	56	--jobs-directory.....	60, 64
--dvd-write-pre-command.....	56	--key-file.....	74
--ecc.....	66	--list.....	10, 43, 74
--exclude.....	46	--log.....	62
--exclude-command.....	46	--log-file.....	62
--exclude-list.....	46	--log-format.....	62
--extract.....	12, 44	--log-post-command.....	63
--file-write-post-command.....	53	--login-dialog.....	73
--file-write-pre-command.....	53	--long-format.....	64
--force-delta-compression.....	65	--maintenance.....	75
--force-ssl.....	74	--max-band-width.....	60
--force-verify-signatures.....	52	--max-storage-size.....	63
--fragment-size.....	52	--max-threads.....	64
--ftp-login-name.....	53	--max-tmp-size.....	63
--ftp-max-connections.....	53	--mount.....	45, 64
--ftp-password.....	53	--mount-command.....	45
--ftp-write-post-command.....	53	--mount-device-command.....	45
--ftp-write-pre-command.....	53	--nice-level.....	60
--full.....	46	--no-bar-on-medium.....	66
--generate-keys.....	44	--no-default-config.....	66
--generate-keys-bits.....	44	--no-detach.....	59
--generate-keys-mode.....	44	--no-fragments-check.....	65
--generate-signature-keys.....	44	--no-header-footer.....	65
--group.....	64	--no-index-database.....	61
--help.....	67, 75	--no-signatures.....	52
--help-internal.....	67	--no-stop-on-attribute-error.....	66
--human-format.....	64	--no-stop-on-error.....	66
--ignore-no-backup-file.....	65	--no-storage.....	66
--ignore-no-dump.....	65	--normal.....	46
--image.....	43	--numeric-permission.....	65
--include.....	45	--numeric-uid-gid.....	65
--include-file command.....	46	--overwrite-archive-files.....	65
--include-file-list.....	45	--overwrite-entries.....	75
--include-image-command.....	46	--overwrite-files.....	66
--include-image-list.....	46	--owner.....	59
--incremental.....	47	--pairing-master-file.....	64
--incremental-data-directory.....	64	--password.....	73
--incremental-list-file.....	47	--pattern-type.....	64
--index-database.....	60	--pause.....	74
--index-database-add.....	74	--permissions.....	59
--index-database-auto-update.....	61	--pid-file.....	64
--index-database-entities-list.....	75	--ping.....	74
--index-database-entries-list.....	75	--port.....	73
--index-database-keep-time.....	61	--post-command.....	45
--index-database-max-band-width.....	61	--pre-command.....	44
--index-database-refresh.....	75	--quiet.....	67
--index-database-remove.....	75	--raw-images.....	65
--index-database-storages-list.....	75	--remote-bar-executable.....	60

--restore.....	75	-D.....	59
--restore-entry-mode.....	66	-e.....	75
--role.....	75	-f.....	46
--scp-write-post-command.....	54	-g.....	64
--scp-write-pre-command.....	54	-h.....	67, 75
--select-job.....	74	-H.....	64
--server-ca-file.....	59	-i.....	47, 74
--server-cert-file.....	59	-j.....	74
--server-key-file.....	59	-l.....	43, 74
--server-max-connections.....	60	-L.....	64
--server-mode.....	60	-m.....	43
--server-password.....	59	-n.....	75
--server-port.....	59	-o.....	65
--server-tls-port.....	59	-p.....	58, 73
--sftp-write-post-command.....	54	-s.....	16, 52, 74
--sftp-write-pre-command.....	54	-t.....	43, 74
--signature-private-key.....	52	-T.....	44
--signatures-public-key.....	52	-x.....	44
--skip-unreadable.....	65	-y.....	50
--skip-verify-signatures.....	52	-z.....	14, 47
--ssh-login-name.....	53	A	
--ssh-max-connections.....	54	archive-name.....	68
--ssh-password.....	53	archive-type.....	68
--ssh-port.....	53	atime.....	9
--ssh-private-key.....	54	B	
--ssh-public-key.....	54	bar-keygen.....	7
--storage-command.....	44, 46	basic commands.....	43
--storage-list.....	44, 46	C	
--storage-list-stdin.....	44	chunks.....	77
--suspend.....	74	compare archive.....	11
--test.....	11, 43	compress archive.....	14
--tls-port.....	73	compression.....	47
--tmp-directory.....	63	creating archive.....	9
--unmount-command.....	45	crypt-password-mode.....	68
--verbose.....	67	D	
--version.....	67, 75	device configuration.....	71
--volume-size.....	66	differential archive.....	13
--wait-first-volume.....	66	E	
--webdav-login.....	54	encryption.....	50
--webdav-max-connections.....	54	exclude.....	69
--webdav-password.....	54	exclude-command.....	69
--webdav-write-post-command.....	55	exclude-list.....	69
--webdav-write-pre-command.....	55	extract archive.....	12
--xhelp.....	67	F	
-!.....	46	file format.....	77
-#.....	45	force-ssl.....	76
-a.....	75	ftp server configuration.....	70
-c.....	43, 74	G	
-C.....	64	graphical front end.....	24
-d.....	43		

I		R	
include-file.....	68	role.....	76
include-file-command.....	68	S	
include-file-list.....	68	schedule configuration.....	71
include-image.....	69	server.....	20, 59, 76
include-image-command.....	69	server-ca-file.....	76
include-image-list.....	69	server-cert-file.....	76
include/exclude files.....	45	server-key-file.....	76
incremental archive.....	13	split archives.....	16, 52
L		ssh/scp/sftp server configuration.....	70
list archive.....	10	store archives.....	17, 53
P		T	
pause-create.....	75	test archive.....	11
pause-index-database-update.....	76	U	
pause-restore.....	76	URI.....	17
pause-storage.....	75	UUID.....	68
persistence configuration.....	72		