AutoArchive

Release 1.1.0

March 06, 2014

CHAPTER 1

Contents

1.1 Program Description

AutoArchive is a simple utility to help create backups more easily. The idea of the program is that all essential information for creating a single backup—such as list of directories that should be archived, the archive name, etc.—is stored in a single file – the *archive specification file*. It can use **tar** for creating archives, it has a command line interface and supports incremental backups.

Archive specification files, also called ".aa files" are normally stored in a predefined location from where they are processed by the `aa` command which results to creating of a corresponding *backup* for each.

Command `autoarchive` is alias for `aa`; these commands are equivalent.

1.1.1 Usage

aa [*options*] [*command*] [*AA_SPEC*]...
autoarchive [*options*] [*command*] [*AA_SPEC*]...

Most of the options can be specified also in *configuration files* and in the *archive specification file* (by using the long option form and leaving out leading dashes) – see *Configuration File* and *Archive Specification File* for complete list of options that can be specified there. Command line options has higher priority than options in configuration files but lower priority than the ones in the archive specification file. `--force-*` options are available for the purpose of overriding some of the options specified in the *.aa file*.

Boolean options can also have a negation form defined. It has the "no-" prefix before the option name. For example: `--incremental` vs. `--no-incremental`. The negation form has always higher priority than the normal form.

List of command line options

Commands:

Commands for program's operations. The default operation is the *backup* creation if no command is specified.

--list	Show all *configured* or *orphaned archives*.
--purge	Purge stored data for an orphaned archive.
--version	Show program's version number and exit.
-h, --help	Show this help message and exit.

Archiving options:

-a ARCHIVER, --archiver=ARCHIVER Specify archiver type. Supported types are: 'tar', 'targz', 'tarbz2', 'tarxz', 'tar_internal', 'targz_internal', 'tarbz2_internal' (default: targz).

-c NUM, --compression-level=NUM Compression strength level. If not specified, default behaviour of underlying compression program will be used.

-d DIR_PATH, --dest-dir=DIR_PATH Directory where the *backup* will be created (default: <current directory>).

Incremental archiving options:

-i, --incremental Perform incremental backup.

-l LEVEL, --level=LEVEL Specify backup level which should be created. All information about higher levels—if any exists—will be erased. If not present, the next level in a row will be created.

--restarting Turn on backup level restarting. See other `*restart-*` options to configure the restarting behaviour.

--restart-after-level=LEVEL Maximal backup level. If reached, it will be restarted back to a lower level (which is typically level 1 but it depends on `--max-restart-level-size`) (default: 10).

--restart-after-age=DAYS Number of days after which the backup level is restarted. Similarly to `--restart-after-level` it will be restarted to level 1 or higher.

--full-restart-after-count=COUNT Number of backup level restarts after which the level is restarted to 0.

--full-restart-after-age=DAYS Number of days after which the backup level is restarted to 0.

--max-restart-level-size=PERCENTAGE Maximal percentage size of a *backup* (of level > 0) to which level is allowed restart to. The size is percentage of size of the level 0 backup file. If a backup of particular level has its size bigger than defined percentage, restart to that level will not be allowed.

--remove-obsolete-backups Turn on removing backups of levels that are no longer valid due to the backup level restart. All backups of the backup level higher than the one currently being created will be removed.

Options for keeping old backups

-k, --keep-old-backups Turn on backup keeping. When a *backup* is about to be overwritten, it is renamed instead. If `--incremental` is enabled it applies to all corresponding *increments*. The new name is created by inserting a *keeping ID* in front of backup file(s) extension. The keeping ID is a string from interval 'aa', 'ab', ..., 'zy', 'zz' where 'aa' represents most recent kept backup.

--number-of-old-backups=NUM Number of old backups to keep when `--keep-old-backups` is enabled (default: 1).

General options:

-v, --verbose Turn on verbose output.

-q, --quiet Turn on quiet output. Only errors will be shown. If `--quiet` is turned on at the same level as `--verbose` (e. g. both are specified on the command line) then `--quiet` has higher priority than `--verbose`.

--all Operate on all *configured archives*. See also `--archive-specs-dir`.

--archive-specs-dir=DIR_PATH Directory where *archive specification files* will be searched for (default: ~/.config/aa/archive_specs).

--user-config-file=FILE_PATH Alternate user configuration file (default: ~/.config/aa/aa.conf).

--user-config-dir=DIR_PATH Alternate user configuration directory (default: ~/.config/aa).

Force options:

Options to override standard options defined in archive specification files.

--force-archiver=ARCHIVER Force archiver type. Supported types are: 'tar', 'targz', 'tarbz2', 'tarxz', 'tar_internal', 'targz_internal', 'tarbz2_internal'.

--force-incremental Force incremental backup.

--force-restarting Force backup level restarting.

--force-compression-level=NUM Force compression strength level.

--force-dest-dir=DIR_PATH Force the directory where the backup will be created.

Negation options:

Negative variants of standard boolean options.

--no-incremental Disable incremental backup.

--no-restarting Turn off backup level restarting.

--no-keep-old-backups Turn off backup keeping.

--no-all Do not operate on all *configured archives*.

AA_SPEC is the *archive specification file argument*. It determines the *archive specification file* that shall be processed. None, single or multiple *AA_SPEC* arguments are allowed. If option `--all` or command `--list` is specified then no *AA_SPEC* argument is required. Otherwise at least single *AA_SPEC* argument is required. If it contains the ".aa" extension then it is taken as the path to an archive specification file. Otherwise, if specified without the extension, the corresponding *.aa file* is searched in the *archive specifications directory*.

1.1.2 Exit Codes

AutoArchive can return following exit codes:

- 0: The operation finished successfully.
- 1: The operation finished with minor (warnings) or major (errors) issues.

1.1.3 Files

~/.config/aa/aa.conf User configuration file. See *Configuration File* for its description.

~/.config/aa/archive_specs/ Default directory that contains *archive specification files*. See *Archive Specification File* for description of the *.aa file* format.

~/.config/aa/snapshots/*.snar Files that stores information about incremental backup. They are created by **GNU tar** archiver.

~/.config/aa/storage/*.realm Application internal persistent storage. It stores various data needed to be preserved between program runs. For example: last backup level restart, number of backup level restart, etc.

/etc/aa/aa.conf System configuration file. See *Configuration File* for its description.

1.2 Operations Explained

AutoArchive can execute several commands. Besides the *backup* creation—its main function—it can show list of *configured* and *orphaned archives* displaying various information about them, or perform a cleaning action that wipes the orphaned archive data. The operation is chosen by specifying the corresponding command as a program's argument. For list of all commands see the *Usage* section.

1.2.1 Configuring the Archive

One of the actions that is actually not handled by the AutoArchive is the configuration of the *archive*. In order to be able to create a *backup* AutoArchive has to be provided by an *archive specification file*. It needs to be created manually and placed to *archive specifications directory* or path to it passed as the program's argument. Archive specification file is a plain text file with simple structure which is described in the *Configuration File* section. Sample files are distributed with the program and an example is provided also in the *Configuring the Archive Example* section.

1.2.2 Backup Creation

Main AutoArchive's function is the *backup* creation. It is the default operation so no command needs to be specified in order to create one. Name or path to an *archive specification file* is required unless `--all` option is given. By default non-incremental *tar.gz* backup is created in the current directory. This can be changed with options on the command line, configuration files or the archive specification file itself. A simple example of the backup creation is shown in the *Backup Creation Example* section. See also *Usage*, *Configuration File* and *Archive Specification File* sections for all possible configuration options.

Incremental Backup Creation

Passing `-i` option on the commandline or specifying corresponding configuration option in a configuration file causes creation of incremental backups. In this case a single full backup is created upon first execution. Subsequent executions will create diff backups with increasing *backup level*. To restore a backup the full backup plus all *increments* (or all increment up to the desired restoration point) are required. Options for manual or automatic restarting to a particular lower level are available. When restarting is applied option `--remove-obsolete-backups` can be specified to remove backups that becomes obsolete due to the restart.

Keeping old backups

In order to reduce risk of losing a valuable older backup AutoArchive can keep backups which are going to be removed or overwritten during a new backup creation. This feature makes possible to have desired number of older backups always available with or without using the incremental archiving. To enable it use `-k` option and to specify desired number of kept backups use the `--number-of-old-backups=NUM` option. The option

`--remove-obsolete-backups` can be used to automatically remove *kept backups* which may become obsolete due to lowering the `--number-of-old-backups=NUM` value.

Each kept backup (or series of kept backup *increments* in case of incremental archiving) has its own *keeping ID*. The most recent kept backup gets keeping ID 'aa', second most recent gets 'ab' and so on up to maximal value 'zz' (which is by default further limited by `--number-of-old-backups=NUM`). When a new backup is going to be kept back all existing kept backup are shifted so that they get higher keeping ID. Backups with keeping ID 'aa' will get 'ab', those with 'ab' gets 'ac' and so on. When number of kept backups would exceed value of the `--number-of-old-backups=NUM` option the last kept backup (with highest keeping ID) is removed.

Refer to *Backup Keeping* section for an example.

1.2.3 Listing Archives

In order to list all archives and show information about them the `--list` command is provided. By default it shows all *archives* that are known to AutoArchive and *orphaned archives*. Note that "archive" here means the "archive configuration", which is represented by the *archive specification file*, not the result of the *backup* creation (the *.tar.gz file). If one or more names or paths to archive specification files are passed as arguments it lists only those.

The output has two forms: normal and verbose.

Normal output

The structure of the normal `--list` output is following:

```
<Name> <Root> <Destination directory> <Current backup level/next/max.>
```

An *archive* per line is listed.

Verbose output

If `--verbose` option is specified alongside with `--list` the verbose form is printed. It shows following information:

```
Name:
Root:
Archiver type:
Destination directory:
Current backup level/next/max.:
Target backup level for non-full restart:
Upcoming restart reason:
Restart count/max.:
Days since last restart/max.:
Days since last full restart/max.:
```

The meaning of the particular fields is:

Name *Archive* name as determined from *archive specification file* name or the `name` option.

Root Archive's root path as configured with `path` option.

Archiver type Type of the archiver as configured with the `archiver` option.

Destination directory Directory where the *backup* will be created as configured with the `dest-dir` option.

Current backup level/next/max. Corresponds to "Current backup level/Next backup level/Maximal backup level". Current backup level is the backup level that was created in last backup creation. Next backup level is the backup

level that will be created in next backup creation (if restarting is enabled it will not be always the next level in a row). Maximal backup level is the value configured with the `restart-after-level` option.

Target backup level for non-full restart Backup level to which will be restarted to in case of non-full backup level restart (for example if `restart-after-level` value is reached. It is typically 1 but can be higher due to `max-restart-level-size` option.

Upcoming restart reason Show the reason of following backup level restart.

Restart count/max. Number of non-full backup level restarts and maximal number of restarts as configured with the `full-restart-after-count` option.

Days since last restart/max. Number of days since last non-full backup level restart occurred and maximal number of days without a restart as configured with the `restart-after-age` option.

Days since last full restart/max. Number of days since last full backup level restart occurred and maximal number of days without a full restart as configured with the `full-restart-after-age` option.

Value format

If the value is enclosed in square brackets ([]) it means that it is not relevant to the current *archive* configuration. For example if an archive was previously configured as incremental and some incremental *backups* were already created, and its configuration was changed to non-incremental later, then the actual backup levels are shown but they are enclosed in square brackets. **In case of** *orphaned archives* **the** *name* **is enclosed in square brackets.**

If the value is not applicable or not available a dash (-) is printed instead.

Sometimes a question mark (?) is printed instead of the value which means that the value could not be determined while it is expected to be available. This happens mostly for orphaned archives where only a limited number of information is available.

1.2.4 Cleaning Orphaned Information

Orphaned archives shown in the `--list` output with their names enclosed in square brackets does not have a corresponding *archive specification file*. It is just leftover information saved in a previous backup creation operation (it is not the *backup* itself). This information can be removed with the `--purge` command. It may be provided with the orphaned archive name in order to remove information about that particular *archive* or with the `--all` option in order to remove information about all orphaned archives.

Note that the `--purge` command does not remove created backups.

1.2.5 Restoration of the Backup

AutoArchive does not handle backup restoration by itself. *Backups* can be restored by using standard **GNU tar** archiver or any other tar-compatible archiver. Please see the **GNU tar** documentation for more information or the *Backup Restoration Example* section for examples on restoring backups.

1.3 Archive Specification File

Archive specification file contains all information needed for creation of a single archive.

Options in the *.aa file* are divided into sections. A section begins with the section name enclosed in square brackets. It contains variables which represents the options.

Variables are written in the `option-name = value` form, one variable per line. Boolean values are written as `yes` and `no`. For path values, tilde (`~`) is expanded to the user's home. Form `option-name =` can be used to specify a variable with undefined value.

Values of `include-files` and `exclude-files` options that contains spaces has to be enclosed in double quotes (`""`). They may contain standard shell wildcards.

When specifying the value it is possible to refer to other variables in the form `%(variable-name)s`.

Lines beginning with `#` or `;` are ignored and may be used for comments.

Two sections are valid: `[Archive]` (optional) and `[Content]`.

1.3.1 Section `[Archive]`

This section can contain configuration options which are, when specified, overriding the ones specified in configuration files and command line.

Options valid in the [Archive] section:

- archiver
- incremental
- restarting
- restart-after-level
- restart-after-age
- full-restart-after-count
- full-restart-after-age
- max-restart-level-size
- remove-obsolete-backups
- keep-old-backups
- number-of-old-backups
- compression-level
- dest-dir

See *Usage* for their description.

1.3.2 Section `[Content]`

This section contains options specific to an archive. All options except `name` are required.

Options valid in the [Content] section:

- name

  Archive name. Created *backup* will be named according value of this variable plus appropriate extension. It is optional; default value is the name of the *.aa file* without the extension.

- path

  Path to archive root. All paths and file names specified in the same archive specification file will be relative to this path. It will be also the root of the created archive.

- include-files

  List of space separated file or directory names to backup. Paths here are relative to the path specified in `path` variable above. Starting forward slash (`/`) from absolute paths as well as parent directory tokens (`..`) will be ignored.

- exclude-files

  List of space separated file or directory names to be excluded from the backup. Use `exclude-files =` to specify that no files should be excluded. Similarly to `include-files` these paths are relative to `path`.

1.4 Configuration File

There are two configuration files for AutoArchive – *system-* and *user-*. *System configuration file's* location is: "/etc/aa/aa.conf". *User configuration file's* location is by default: "~/.config/aa/aa.conf". Options in the *user file* have higher priority. Note that some options can only be specified in the *system file* (see the list of the options below).

Structure is similar to the *archive specification file* – options are divided into sections. A section begins with the section name enclosed in square brackets. Sections contains variables which represents the options.

Variables are written in the `option-name = value` form, one variable per line. Boolean values are written as `yes` and `no`. For path values, tilde (`~`) is expanded to the user's home. Form `option-name =` can be used to specify a variable with undefined value.

Lines beginning with `#` or `;` are ignored and may be used for comments.

Two sections are valid: `[General]` and `[Archive]`. Both are optional although a configuration file without any section at all is considered invalid.

1.4.1 Section `[General]`

Contains configuration options for the program itself, which do not alter the backup creation.

Options valid in the [General] section:

- verbose
- quiet
- archive-specs-dir
- user-config-file

  This option can not be specified in the *user configuration file*.

- user-config-dir

  This option can not be specified in the *user configuration file*.

See *Usage* for their description.

1.4.2 Section `[Archive]`

This section contains configuration options which are specific to the backup creation.

Options valid in the [Archive] section:

- archiver
- incremental

- restarting
- restart-after-level
- restart-after-age
- full-restart-after-count
- full-restart-after-age
- max-restart-level-size
- remove-obsolete-backups
- keep-old-backups
- number-of-old-backups
- compression-level
- dest-dir
- force-archiver
- force-incremental
- force-restarting
- force-compression-level
- force-dest-dir

See *Usage* for their description.

1.5 Examples

Let's make a *backup* of configuration files of all users except the user "foo". Let's assume that our system has unix-like style of home directories (directory "/home" contains directories of all users; configuration files begins with dot). Name of this backup will be "user-configs".

1.5.1 Configuring the Archive Example

First, we need to create the file "user-configs.aa" under the "~/.config/aa/archive_specs/" directory - this is the *archive specification file*. The file doesn't need to have the same name as the *backup*. If it does however, the option `name` can be left out (in this example we specified it anyway, even it is not needed).

In the `path` variable we specify the archive root which is the **the base directory which content we want to backup**.

Variables `include-files` and `exclude-files` contains list of files and directories that we want to be included or excluded respectively. In this example we specify `*/.*` pattern because we want to include home directories of all users (such as /home/**bob**, /home/**joe**, etc.), what the first `*` is for. And from within those user home directories we want to include everything that begins with `.` (for example /home/bob/**.bashrc**), what the `.*` pattern is for. Paths specified in these variables are relative to `path`.

Although, yet we do not want to include *all* user home directories as we specified in `include-files`. Those directories that should not be included we put in `exclude-files` ("foo" in this example, which makes /home/foo excluded). If we would not want to exclude any file then the corresponding variable would be specified as `exclude-files =`.

Content of the "user-configs.aa" file:

```
# ------ begin of user-configs.aa ------
# AutoArchive's archive specification file for users configuration files
[Content]
name = user-configs
path = /home
include-files = */.*
exclude-files = foo

[Archive]
dest-dir = /mnt/backup
# ------ end of user-configs.aa ------
```

1.5.2 Backup Creation Example

Once we configured the *archive* we can create the *backup* easily with command:

```
aa user-configs
```

and in the "/mnt/backup" directory the file "user-configs.tar.gz" will be created.

Given the "user-configs.aa" example file above, the command:

```
aa -i user-configs
```

will create **level 0** incremental backup – "user-configs.tar.gz" which is essentially the same as a non-incremental backup. Another execution of the same command will create **level 1** backup named "user-configs.1.tar.gz" which contains only a differences from level 0. Each subsequent call will create a **next level** which will contain only a differences from previous.

In order to restart to level 0 again, thus create a **fresh full backup**, the following command can be used:

```
aa -i -l 0 user-configs
```

Note that you **should remove** all previously created "user-configs" backups with *level* higher than 0 because they are no longer valid in regards to the newly created level 0 backup. You may pass `--remove-obsolete-backups` option to the command above and they will be removed automatically.

Backup Keeping

We assume that all previously created backups were removed in order to demonstrate the backup keeping.

First we create a standard backup:

```
aa user-configs
```

This creates "user-configs.tar.gz" backup. Some days later let's say, we want to create the same backup again. However we do not want to overwrite the original one. The option `-k` can be used to keep the original backup:

```
aa -k user-configs
```

This will rename the original backup to "user-configs.**aa**.tar.gz" and create the new one "user-configs.tar.gz". If we create the same backup for the third time (still using the `-k`) option, "user-configs.aa.tar.gz" will be removed, "user-configs.tar.gz" will be renamed to "user-configs.aa.tar.gz" and the new "user-configs.tar.gz" will be created. So AutoArchive by default keeps single old backup when `-k` options is specified. To keep more, e.g. four backups we would specify `--number-of-old-backups=4` alongside with `-k`.

Incremental backups can be kept as well. Again, we assume that all previously created backups were removed. Let's create a few levels of incremental "user-configs" archive:

```
aa -i -l 0 user-configs
aa -i user-configs
aa -i user-configs
aa -i user-configs
```

This will create following files:

```
user-configs.tar.gz
user-configs.1.tar.gz
user-configs.2.tar.gz
user-configs.3.tar.gz
```

Then we (manually) restart to level 2 while asking to keep old backups:

```
aa -i -l 2 -k user-configs
```

After this command following files will be present:

```
user-configs.tar.gz
user-configs.1.tar.gz
user-configs.2.tar.gz
user-configs.2.aa.tar.gz
user-configs.3.aa.tar.gz
```

Let's explain what happened. The original file "user-configs.2.tar.gz" was going to be overwritten therefore it was renamed to "user-configs.2.aa.tar.gz". As all backup levels higher than the renamed one depends on it they have to be renamed as well. In this example "user-configs.3.tar.gz" depends on "user-configs.2.tar.gz" therefore it was renamed to "user-configs.3.aa.tar.gz". Finally the new *increment* "user-configs.2.tar.gz" was created.

1.5.3 Listing Archives Example

Our "user-configs" *archive* can be listed by following command:

```
aa --list
```

Which results to the following output:

```
user-configs /home                    /mnt/backups               [0]/[1]/[10]
```

If we pass `--verbose` option then it shows:

```
Name: user-configs
Root: /home
Archiver type: targz
Destination directory: /mnt/backups
Current backup level/next/max.: [0]/[1]/[10]
Target backup level for non-full restart: [1]
Upcoming restart reason: [No restart scheduled for the next backup.]
Restart count/max.: [-]/[-]
Days since last restart/max.: [-]/[-]
Days since last full restart/max.: [-]/[-]
```

The archive *Name* is "user-configs" as configured with the `name` variable in the *Configuring the Archive Example* section. *Root* corresponds to the value configured with the `path` variable. *Archiver type* is "targz" which is the default. *Destination directory* "/mnt/backup" is configured with `dest-dir` variable. *Current backup level/next/max.* shows [0]/[1]/[10] because in the section *Backup Creation Example* we have created an incremental backup of level 0, so current level is 0. Next level is 1 (restarting is not enabled). Both the current and the next levels are enclosed in square brackets because incremental archiving is not enabled (it was enabled only temporarily with the `-i` option).

Finally, the maximal backup level is 10 as it is the default. It is also shown in square brackets because restarting is not enabled; this also applies for all following values. Since no `max-restart-level-size` is specified the *Target backup level for non-full restart* is and always be 1. Obviously, no restart is scheduled as the *Upcoming restart reason* value is showing. Since no restart ever occurred and no value is specified for the rest of restarting options the values *Restart count/max.*, *Days since last restart/max.* and *Days since last full restart/max.* shows only dashes.

1.5.4 Cleaning Orphaned Information Example

If we remove the "user-configs.aa" *archive specification file* then the `--list` will still be showing the *archive* with its name enclosed in square brackets (it becomes the *orphaned archive*):

```
[user-configs] ?                    .                                [0]/[?]/[10]
```

This is because some information is still stored in the AutoArchive's configuration directory. It is the snapshot file created by **tar** when incremental *backup* was created. There may be more information left behind if restarting would be enabled. All of this orphaned information can be deleted with the `--purge` command:

```
aa --purge user-configs
```

or:

```
aa --purge --all
```

which would remove all orphaned archives.

1.5.5 Backup Restoration Example

Restoring Non-Incremental Backup

Let's say we have created simple (non-incremental) backup as in the *Backup Creation Example*. Thus we have a file called "user-configs.tar.gz" in the "/mnt/backup" directory. As the AutoArchive does not handle restoration we will use standard **GNU tar** archiver.

To restore the backup to its original destination and thus **replace all existing files with the ones from the backup** we can use following command:

```
tar -xf /mnt/backup/user-configs.tar.gz -C /home
```

The value of the `-C` option (/home) is the same as the value of `path` variable in the "user-configs.aa". The `-C` option can be left out if the destination is the *current working directory* (in other words you did "cd /home" earlier).

Of course the backup can be restored to any arbitrary location by replacing "/home" with some other path in the command above. This may be more safe and convenient as it does not replaces original files. The extracted backup files can be reviewed and copied to the original destination afterwards. You may also use a graphical file manager or an archive manager to browse content of the backup interactively.

Restoring Incremental Backup

Suppose we have several increments of the "user-configs" archive in the /mnt/backup directory. The content of the directory is following:

```
$ ls -1 /mnt/backup
-rw-r--r-- 1 root root  10M Apr 20 17:07 user-configs.tar.gz
-rw-r--r-- 1 root root   1M May 11 12:21 user-configs.1.tar.gz
-rw-r--r-- 1 root root 1.5M Jun 26 16:43 user-configs.2.tar.gz
```

Which means we have backup level 0, 1 and 2. To restore entire backup to the latest possible date (in this case Jun 26) we have to restore all backup levels. Similarly to the previous example the following series of commands will restore the backup to the original location **replacing** the original files there:

```
tar -xf /mnt/backup/user-configs.tar.gz -G -C /home
tar -xf /mnt/backup/user-configs.1.tar.gz -G -C /home
tar -xf /mnt/backup/user-configs.2.tar.gz -G -C /home
```

As in the previous example the "-C /home" can be left out (backup will be restored to the current directory) or "/home" replaced with some other path (backup will be restored to that path).

1.6 General Information

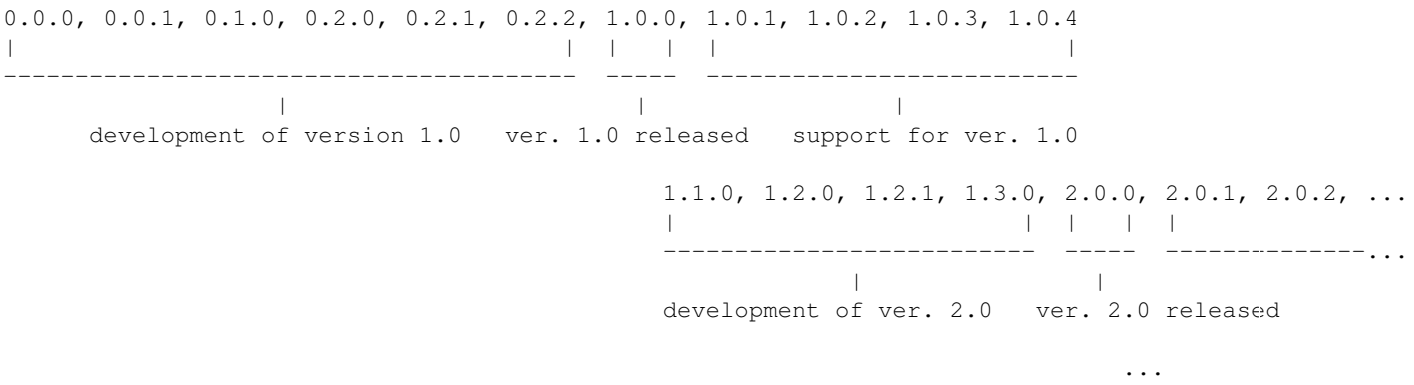
1.6.1 Versioning Scheme

AutoArchive version has form *X.Y.Z*, where *X* is the **major**, *Y* is **minor** and *Z* the **bugfix** version number.

A major version is released when all features for it are implemented. When it happens *X* is increased and other numbers are set to 0 (e. g. from 0.14.5 to 1.0.0). No new features are being added to that version anymore, only bug fixes. The version is supported until the next major version is released.

After a new major version is released the development of the next one starts. It has the same major version number (*X*) as current stable, however the minor (*Y*) is greater than 0 and is increasing (e. g. after 1.0.0 is released, the development of 2.0.*z* starts with version 1.1.0).

This is how releases may look like:



Generally, increasing *X* or *Y* means that new features were introduced. They may bring incompatibilities with previous releases (such as change of the configuration file format and so on). A migration may be necessary after the update.

Increasing *Z* means that only bugs were fixed and the release is compatible with the previous one. Update is seamless, no migration is necessary.

1.6.2 License

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

1.6.3 Contacting the Author

Comments, bug reports, wishes, donations for this piece of software are welcomed. You can send them via the project page at http://sourceforge.net/projects/autoarchive/ or use e-mail openhs@users.sourceforge.net.

Homepage: http://autoarchive.sourceforge.net/.

1.7 Glossary

.aa file A synonym for *archive specification file*.

archive The primary entity that AutoArchive operates with. It has a name and holds the configuration used to create the corresponding *backup*. *Archive* is represented by the *archive specification file*.

archive specification file The configuration of an *archive*. It contains all information needed for creation of a single backup, such as: archive name, archive root directory, list of directories and files which should be **included** and **excluded**, etc. It can also contain some of the configuration options. The file has extension '.aa' and is sometimes referred as ".aa file". For more information see *Archive Specification File*.

archive specifications directory Directory where *archive specification files* are stored. It can be configured via the `archive-specs-dir` option.

backup Result of the backup creation operation. For example a *.tar.gz file.

backup level For incremental *archives* it represents an iteration of a particular *backup*. It start from 0 which always represents the full backup. Values 1 and greater represents diff backups to previous *level*. The physical representation of a backup level is *increment*.

configured archive See *archive*.

increment A *backup* that has a particular *backup level*. For example a *.2.tar.gz file is increment of backup level 2. It applies to incremental *archives*.

keeping ID The identification of *kept backups*. It can have values from a following set: 'aa', 'ab', ..., 'zy', 'zz' where 'aa' is ID of the most recent kept backup.

kept backup A *backup* that normally should have been already removed or overwritten but was preserved under a different name. The new name consists of the original name and its *keeping ID*, for example *.aa.tar.gz is a *kept backup* with keeping ID 'aa'.

level See *backup level*.

orphaned archive *Archive* that has no *archive specification file* but has some data stored (snapshot files, information about last backup level restart etc.).

1.8 AutoArchive

1.8.1 Synopsis

aa [*options*] [*command*] [*AA_SPEC*]...
autoarchive [*options*] [*command*] [*AA_SPEC*]...

1.8.2 Description

AutoArchive is a simple utility to help create backups more easily. The idea of the program is that all essential information for creating a single backup—such as list of directories that should be archived, the archive name, etc.—is stored in a single file – the *archive specification file*. It can use *tar*(1) for creating archives, it has a command line interface and supports incremental backups.

Archive specification files, also called ".aa files" are normally stored in a predefined location from where they are processed by the `aa` command which results to creating of a corresponding *backup* for each.

Command `autoarchive` is alias for `aa`; these commands are equivalent.

1.8.3 Options

Most of the options can be specified also in *configuration files* and in the *archive specification file* (by using the long option form and leaving out leading dashes) – see *aa.conf*(5) and *aa_arch_spec*(5) for complete list of options that can be specified there. Command line options has higher priority than options in configuration files but lower priority than the ones in the archive specification file. `--force-*` options are available for the purpose of overriding some of the options specified in the *.aa file*.

Boolean options can also have a negation form defined. It has the "no-" prefix before the option name. For example: `--incremental` vs. `--no-incremental`. The negation form has always higher priority than the normal form.

List of command line options

Commands:

Commands for program's operations. The default operation is the *backup* creation if no command is specified.

--list Show all *configured* or *orphaned archives*.

--purge Purge stored data for an orphaned archive.

--version Show program's version number and exit.

-h, --help Show this help message and exit.

Archiving options:

-a ARCHIVER, --archiver=ARCHIVER Specify archiver type. Supported types are: 'tar', 'targz', 'tarbz2', 'tarxz', 'tar_internal', 'targz_internal', 'tarbz2_internal' (default: targz).

-c NUM, --compression-level=NUM Compression strength level. If not specified, default behaviour of underlying compression program will be used.

-d DIR_PATH, --dest-dir=DIR_PATH Directory where the *backup* will be created (default: <current directory>).

Incremental archiving options:

-i, --incremental Perform incremental backup.

-l LEVEL, --level=LEVEL Specify backup level which should be created. All information about higher levels—if any exists—will be erased. If not present, the next level in a row will be created.

--restarting Turn on backup level restarting. See other `*restart-*` options to configure the restarting behaviour.

--restart-after-level=LEVEL Maximal backup level. If reached, it will be restarted back to a lower level (which is typically level 1 but it depends on `--max-restart-level-size`) (default: 10).

--restart-after-age=DAYS Number of days after which the backup level is restarted. Similarly to `--restart-after-level` it will be restarted to level 1 or higher.

--full-restart-after-count=COUNT Number of backup level restarts after which the level is restarted to 0.

--full-restart-after-age=DAYS Number of days after which the backup level is restarted to 0.

--max-restart-level-size=PERCENTAGE Maximal percentage size of a *backup* (of level > 0) to which level is allowed restart to. The size is percentage of size of the level 0 backup file. If a backup of particular level has its size bigger than defined percentage, restart to that level will not be allowed.

--remove-obsolete-backups Turn on removing backups of levels that are no longer valid due to the backup level restart. All backups of the backup level higher than the one currently being created will be removed.

Options for keeping old backups

-k, --keep-old-backups Turn on backup keeping. When a *backup* is about to be overwritten, it is renamed instead. If `--incremental` is enabled it applies to all corresponding *increments*. The new name is created by inserting a *keeping ID* in front of backup file(s) extension. The keeping ID is a string from interval 'aa', 'ab', ..., 'zy', 'zz' where 'aa' represents most recent kept backup.

--number-of-old-backups=NUM Number of old backups to keep when `--keep-old-backups` is enabled (default: 1).

General options:

-v, --verbose Turn on verbose output.

-q, --quiet Turn on quiet output. Only errors will be shown. If `--quiet` is turned on at the same level as `--verbose` (e. g. both are specified on the command line) then `--quiet` has higher priority than `--verbose`.

--all Operate on all *configured archives*. See also `--archive-specs-dir`.

--archive-specs-dir=DIR_PATH Directory where *archive specification files* will be searched for (default: ~/.config/aa/archive_specs).

--user-config-file=FILE_PATH Alternate user configuration file (default: ~/.config/aa/aa.conf).

--user-config-dir=DIR_PATH Alternate user configuration directory (default: ~/.config/aa).

Force options:

Options to override standard options defined in archive specification files.

--force-archiver=ARCHIVER Force archiver type. Supported types are: 'tar', 'targz', 'tarbz2', 'tarxz', 'tar_internal', 'targz_internal', 'tarbz2_internal'.

--force-incremental Force incremental backup.

--force-restarting Force backup level restarting.

--force-compression-level=NUM Force compression strength level.

--force-dest-dir=DIR_PATH Force the directory where the backup will be created.

Negation options:

Negative variants of standard boolean options.

--no-incremental Disable incremental backup.

--no-restarting Turn off backup level restarting.

--no-keep-old-backups Turn off backup keeping.

--no-all Do not operate on all *configured archives*.

AA_SPEC is the *archive specification file argument*. It determines the *archive specification file* that shall be processed. None, single or multiple *AA_SPEC* arguments are allowed. If option `--all` or command `--list` is specified then no *AA_SPEC* argument is required. Otherwise at least single *AA_SPEC* argument is required. If it contains the ".aa" extension then it is taken as the path to an archive specification file. Otherwise, if specified without the extension, the corresponding *.aa file* is searched in the *archive specifications directory*.

1.8.4 Exit Codes

AutoArchive can return following exit codes:

- 0: The operation finished successfully.
- 1: The operation finished with minor (warnings) or major (errors) issues.

1.8.5 Files

~/.config/aa/aa.conf User configuration file. See *aa.conf*(5) for its description.

~/.config/aa/archive_specs/ Default directory that contains *archive specification files*. See *aa_arch_spec*(5) for description of the *.aa file* format.

~/.config/aa/snapshots/*.snar Files that stores information about incremental backup. They are created by **GNU tar** archiver.

~/.config/aa/storage/*.realm Application internal persistent storage. It stores various data needed to be preserved between program runs. For example: last backup level restart, number of backup level restart, etc.

/etc/aa/aa.conf System configuration file. See *aa.conf*(5) for its description.

1.8.6 Examples

Let's make a *backup* of configuration files of all users except the user "foo". Let's assume that our system has unix-like style of home directories (directory "/home" contains directories of all users; configuration files begins with dot). Name of this backup will be "user-configs".

First, we need to create the file "user-configs.aa" under the "~/.config/aa/archive_specs/" directory - this is the *archive specification file*. The file doesn't need to have the same name as the *backup*. If it does however, the option `name` can be left out (in this example we specified it anyway, even it is not needed).

In the `path` variable we specify the archive root which is the **the base directory which content we want to backup**.

Variables `include-files` and `exclude-files` contains list of files and directories that we want to be included or excluded respectively. In this example we specify `*/.*` pattern because we want to include home directories of all users (such as /home/**bob**, /home/**joe**, etc.), what the first `*` is for. And from within those user home directories we want to include everything that begins with `.` (for example /home/bob/**.bashrc**), what the `.*` pattern is for. Paths specified in these variables are relative to `path`.

Although, yet we do not want to include *all* user home directories as we specified in `include-files`. Those directories that should not be included we put in `exclude-files` ("foo" in this example, which makes /home/foo excluded). If we would not want to exclude any file then the corresponding variable would be specified as `exclude-files =`.

Content of the "user-configs.aa" file:

```
# ------ begin of user-configs.aa ------
# AutoArchive's archive specification file for users configuration files
[Content]
name = user-configs
path = /home
include-files = */.*
exclude-files = foo

[Archive]
dest-dir = /mnt/backup
# ------ end of user-configs.aa ------
```

Once we configured the *archive* we can create the *backup* easily with command:

```
aa user-configs
```

and in the "/mnt/backup" directory the file "user-configs.tar.gz" will be created.

Given the "user-configs.aa" example file above, the command:

```
aa -i user-configs
```

will create **level 0** incremental backup – "user-configs.tar.gz" which is essentially the same as a non-incremental backup. Another execution of the same command will create **level 1** backup named "user-configs.1.tar.gz" which contains only a differences from level 0. Each subsequent call will create a **next level** which will contain only a differences from previous.

In order to restart to level 0 again, thus create a **fresh full backup**, the following command can be used:

```
aa -i -l 0 user-configs
```

Note that you **should remove** all previously created "user-configs" backups with *level* higher than 0 because they are no longer valid in regards to the newly created level 0 backup. You may pass `--remove-obsolete-backups` option to the command above and they will be removed automatically.

Backup Keeping

We assume that all previously created backups were removed in order to demonstrate the backup keeping.

First we create a standard backup:

```
aa user-configs
```

This creates "user-configs.tar.gz" backup. Some days later let's say, we want to create the same backup again. However we do not want to overwrite the original one. The option `-k` can be used to keep the original backup:

```
aa -k user-configs
```

This will rename the original backup to "user-configs.**aa**.tar.gz" and create the new one "user-configs.tar.gz". If we create the same backup for the third time (still using the `-k`) option, "user-configs.aa.tar.gz" will be removed, "user-configs.tar.gz" will be renamed to "user-configs.aa.tar.gz" and the new "user-configs.tar.gz" will be created. So AutoArchive by default keeps single old backup when `-k` options is specified. To keep more, e.g. four backups we would specify `--number-of-old-backups=4` alongside with `-k`.

Incremental backups can be kept as well. Again, we assume that all previously created backups were removed. Let's create a few levels of incremental "user-configs" archive:

```
aa -i -l 0 user-configs
aa -i user-configs
aa -i user-configs
aa -i user-configs
```

This will create following files:

```
user-configs.tar.gz
user-configs.1.tar.gz
user-configs.2.tar.gz
user-configs.3.tar.gz
```

Then we (manually) restart to level 2 while asking to keep old backups:

```
aa -i -l 2 -k user-configs
```

After this command following files will be present:

```
user-configs.tar.gz
user-configs.1.tar.gz
user-configs.2.tar.gz
user-configs.2.aa.tar.gz
user-configs.3.aa.tar.gz
```

Let's explain what happened. The original file "user-configs.2.tar.gz" was going to be overwritten therefore it was renamed to "user-configs.2.aa.tar.gz". As all backup levels higher than the renamed one depends on it they have to be renamed as well. In this example "user-configs.3.tar.gz" depends on "user-configs.2.tar.gz" therefore it was renamed to "user-configs.3.aa.tar.gz". Finally the new *increment* "user-configs.2.tar.gz" was created.

1.8.7 License

1.8.8 See Also

aa.conf(5), *aa_arch_spec*(5), *tar*(1), *gzip*(1), *bzip2*(1), *xz*(1)

1.9 Archive Specification File

1.9.1 Synopsis

~/.config/aa/archive_specs/*.aa

1.9.2 Description

Archive specification file contains all information needed for creation of a single archive.

1.9.3 File Format

Options in the *.aa file* are divided into sections. A section begins with the section name enclosed in square brackets. It contains variables which represents the options.

Variables are written in the `option-name = value` form, one variable per line. Boolean values are written as `yes` and `no`. For path values, tilde (`~`) is expanded to the user's home. Form `option-name =` can be used to specify a variable with undefined value.

Values of `include-files` and `exclude-files` options that contains spaces has to be enclosed in double quotes (`""`). They may contain standard shell wildcards.

When specifying the value it is possible to refer to other variables in the form `%(variable-name)s`.

Lines beginning with `#` or `;` are ignored and may be used for comments.

Two sections are valid: `[Archive]` (optional) and `[Content]`.

Section `[Archive]`

This section can contain configuration options which are, when specified, overriding the ones specified in configuration files and command line.

Options valid in the [Archive] section:

- archiver
- incremental
- restarting
- restart-after-level
- restart-after-age
- full-restart-after-count
- full-restart-after-age
- max-restart-level-size
- remove-obsolete-backups
- keep-old-backups
- number-of-old-backups
- compression-level
- dest-dir

See *aa*(1) for their description.

Section `[Content]`

This section contains options specific to an archive. All options except `name` are required.

Options valid in the [Content] section:

- name

  Archive name. Created *backup* will be named according value of this variable plus appropriate extension. It is optional; default value is the name of the *.aa file* without the extension.

- path

  Path to archive root. All paths and file names specified in the same archive specification file will be relative to this path. It will be also the root of the created archive.

- include-files

  List of space separated file or directory names to backup. Paths here are relative to the path specified in `path` variable above. Starting forward slash (`/`) from absolute paths as well as parent directory tokens (`..`) will be ignored.

- exclude-files

  List of space separated file or directory names to be excluded from the backup. Use `exclude-files =` to specify that no files should be excluded. Similarly to `include-files` these paths are relative to `path`.

1.9.4 License

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License version 3 as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

1.9.5 See Also

aa(1), *aa.conf*(5)

1.10 AutoArchive Configuration File

1.10.1 Synopsis

~/.config/aa/aa.conf
/etc/aa/aa.conf

1.10.2 File Format

There are two configuration files for AutoArchive – *system-* and *user-*. *System configuration file's* location is: "/etc/aa/aa.conf". *User configuration file's* location is by default: "~/.config/aa/aa.conf". Options in the *user file* have higher priority. Note that some options can only be specified in the *system file* (see the list of the options below).

Structure is similar to the *archive specification file* – options are divided into sections. A section begins with the section name enclosed in square brackets. Sections contains variables which represents the options.

Variables are written in the `option-name = value` form, one variable per line. Boolean values are written as `yes` and `no`. For path values, tilde (`~`) is expanded to the user's home. Form `option-name =` can be used to specify a variable with undefined value.

Lines beginning with `#` or `;` are ignored and may be used for comments.

Two sections are valid: `[General]` and `[Archive]`. Both are optional although a configuration file without any section at all is considered invalid.

Section `[General]`

Contains configuration options for the program itself, which do not alter the backup creation.

Options valid in the [General] section:

- verbose
- quiet
- archive-specs-dir
- user-config-file

  This option can not be specified in the *user configuration file*.
- user-config-dir

  This option can not be specified in the *user configuration file*.

See *aa*(1) for their description.

Section `[Archive]`

This section contains configuration options which are specific to the backup creation.

Options valid in the [Archive] section:

- archiver
- incremental
- restarting
- restart-after-level
- restart-after-age
- full-restart-after-count
- full-restart-after-age
- max-restart-level-size
- remove-obsolete-backups

- keep-old-backups
- number-of-old-backups
- compression-level
- dest-dir
- force-archiver
- force-incremental
- force-restarting
- force-compression-level
- force-dest-dir

See *aa*(1) for their description.

1.10.3 License

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License version 3 as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

1.10.4 See Also

aa(1), *aa_arch_spec*(5)

1.11 AutoArchive ver. 1.1.0

A simple backup utility.

1.11.1 Program Description

AutoArchive is a simple utility to help create backups more easily. The idea of the program is that all essential information for creating a single backup—such as list of directories that should be archived, the archive name, etc.—is stored in a single file – the *archive specification file*. It can use 'tar' for creating archives, it has a command line interface and supports incremental backups.

Archive specification files, also called ".aa files" are normally stored in a predefined location from where they are processed by the `aa` command which results to creating of a corresponding *backup* for each.

Command `autoarchive` is alias for `aa`; these commands are equivalent.

1.11.2 Usage

```
aa [options] [command] [AA_SPEC]...
autoarchive [options] [command] [AA_SPEC]...
```

For complete list of command line options please see the *aa*(1) manual page or *AutoArchive User Manual*.

Most of the options can be specified also in *configuration files* and in the *archive specification file* (by using the long option form and leaving out leading dashes) – see *aa.conf*(5) and *aa_arch_spec*(5) manual pages or *Configuration Files Description* and *Archive Specification File Description* sections in the *AutoArchive User Manual* for complete list of options that can be specified there. Command line options has higher priority than options in configuration files but lower priority than the ones in the archive specification file. `--force-*` options are available for the purpose of overriding some of the options specified in the *.aa file*.

Boolean options can also have a negation form defined. It has the "no-" prefix before the option name. For example: `--incremental` vs. `--no-incremental`. The negation form has always higher priority than the normal form.

AA_SPEC is the *archive specification file argument*. It determines the *archive specification file* that shall be processed. None, single or multiple *AA_SPEC* arguments are allowed. If option `--all` or command `--list` is specified then no *AA_SPEC* argument is required. Otherwise at least single *AA_SPEC* argument is required. If it contains the ".aa" extension then it is taken as the path to an archive specification file. Otherwise, if specified without the extension, the corresponding *.aa file* is searched in the *archive specifications directory*.

1.11.3 Contacting the Author

Comments, bug reports, wishes, donations for this piece of software are welcomed. You can send them via the project page at http://sourceforge.net/projects/autoarchive/ or use e-mail openhs@users.sourceforge.net.

Homepage: http://autoarchive.sourceforge.net/.

CHAPTER 2

Indices and tables

- *genindex*
- *search*
- *Glossary*