Akeeba Backup User's Guide

Nicholas K. Dionysopoulos

Akeeba Backup User's Guide

by Nicholas K. Dionysopoulos

Abstract

This book covers the use of the Akeeba Backup site backup component for Joomla!™ -powered web sites. It does not cover any other software of the Akeeba Backup suite, including Kickstart and the desktop applications which have documentation of their own. Both the free Akeeba Backup Core and the subscription-based Akeeba Backup Professional editions are completely covered.

If you are looking for a quick start to using the component please watch our video tutorials [https://www.akeeba.com/videos].

Table of Contents

Part I. User's Guide to Akeeba Backup for Joomla!™

Table of Contents

Chapter 1. Introduction

1. Introducing Akeeba Backup

Akeeba Backup is a complete site backup solution for your Joomla!™ powered website. It is designed to put your entire site – files and database data – in a backup archive file you can restore on the same or a different server. The restoration uses a web installer script. You do not have to mess with the command line or database management tools – or even edit configuration files by hand.

More than that, Akeeba Backup is putting you in control of your data. If you so with you can fine-tuning your backup choosing which directories, files or database tables to exclude. It can even allow you to backup non-Joomla!™ content, as long as you specify which off-site directories and databases you want to add.

Tip

If you are looking for a quick start to using the component please watch our video tutorials [https://www.akeeba.com/videos].

2. What can I use Akeeba Backup for?

Akeeba Backup can be used for much more than just backing up your site. Some indicative uses are:

- **Security backups**. Taking a snapshot of your site should your server fail, or a hacker exploit some security hole to deface or compromise your site.
- **Template sites**. Web professionals have used Akeeba Backup in order to create "template sites". This means that you can build a site on a local server, install every component you usually do on most clients' sites and back it up. You now have a canned site that can serve as a great template for future clients. Using the same method you can have a snapshot of all the sites you have built for your clients, without the need to have them installed on your local server.
- **Build a site off-line, upload the finished site easily**. Web professionals can build a complete site off-line on a local server and when done take a snapshot with Akeeba Backup, then restore it on the production site.
- **Testing upgrades locally, without risking breaking the on-line site**. Joomla!™ updates have the potential of breaking things, especially in complex or badly written components and modules. Web masters use Akeeba Backup to get a site snapshot, restore it on a local test server, perform the upgrade there and test for any problems without the live site being at risk.
- **Debugging locally**. Almost the same as above, web professionals have used Akeeba Backup to take a snapshot of a client's Joomla!™ site in order to perform bug hunting. Using Akeeba Backup again, they can upload the fixed site back on the live server.
- **Relocating a site to a new host**. Web masters who want to take their site to a new host have found Akeeba Backup to be their saviour. Just backup the original site and restore on the new host; presto, your site is relocated with virtually no effort at all.

Akeeba Backup has the potential to save you hours of hard labor, according to our users. It is licensed under the GNU General Public License version 3 or, at your option, any later version of the license. As a result, you are free to install it on as many sites as you like *without* having to pay for a pricey "developer's license".

Akeeba Backup comes in two editions, Core and Professional. Akeeba Backup Core is provided free of charge and contains all the features a typical webmaster would like to have in order to easily complete backup and restoration jobs. On top of that, the video tutorials and the full documentation are free of charge as well.

Akeeba Backup Professional is designed with the needs of web professionals in mind. It has features such as inclusion of external directories and databases, powerful filters based on regular expressions, and support for

sending your backups on cloud storage services (such as Amazon S3, Dropbox or any other of the 40+ supported providers). Thanks to its GNU GPL v3 license Akeeba Backup Professional can be installed on an unlimited number of clients' websites without having to purchase additional subscriptions.

3. A typical backup/restoration work flow

Tip

If you are looking for a quick start to using the component please watch our video tutorials [https://www.akeeba.com/videos].

As stated, Akeeba Backup is designed to make your life easier. It does that by streamlining the work flow of backing up and restoring (or migrating) your site. From Akeeba Backup's perspective, restoring to the same host and location, copying your site in a subdirectory / subdomain of the same host or transferring your site to a completely new host is identical. That's right, Akeeba Backup doesn't care if you are restoring, copying, cloning or migrating your site! The process is always the same, so you only have to learn it once. The learning curve is very smooth, too!

Warning

Do not attempt to restore to a different database technology, e.g. a backup taken on MySQL restored on PostgreSQL. This won't work and the restoration script won't even let you try. This is a consequence of how different database server technologies structure the databases and the tables in said databases. Restoring to different servers of the same database technology usually does work. This means that MySQL, MariaDB and Percona are mostly compatible with each other. Please remember that Akeeba Backup only supports database servers compatible with MySQL.

The typical work flow involves using two utilities from the Akeeba Backup suite: the Akeeba Backup component itself, and Akeeba Kickstart. Here is the overview:

1. Install Akeeba Backup and configure it to taste. Or use the automated Configuration Wizard to automatically configure it with the perfect settings for your server. Hit on the Backup Now button and let your site back up. When it finishes up, click on the Manage Backups button. Click on the download links on the far-right of the only backup entry from the list - or, better yet, use FTP to do that - saving all parts of the backup archive somewhere on your local PC.

2. Extract the kickstart-*`VERSION`*.zip file you downloaded from our Downloads repository. The only contained files are `kickstart.php` and the translation INI files. Upload them to the server on which you want to restore your site to.

3. Upload all parts of the backup archive (do not extract it yet, just upload the files) to the server on which you want to restore your site to (called here forth the *target server*). Your server's directory should now contain the `kickstart.php` and the parts of the backup archive (.jpa, .j01, etc).

4. Fire up your browser and visit the Kickstart URL on your target server, for example `http://www.example.com/kickstart.php` .

5. Change any option - if necessary - and hit the Start button. Sit back while Kickstart extracts the backup archive directly on the server! It's ultra-fast too (when compared to FTP uploading all those 4000+ files!). If it fails with an error, go back, select the `Upload using FTP` option and supply your FTP connection information, then click on Start again.

6. A new window pops up. It's the Akeeba Backup Installer (ABI), the site restoration script which was embedded inside your archive. Do not close the Kickstart window yet!

7. Follow the prompts of the Akeeba Backup Installer, filling in the details of the new server (most importantly, the new database connection and FTP connection information).

8. When the Akeeba Backup Installer is done, it prompts you to delete the installation directory. Ignore this prompt and simply close the ANGIE window.

9. Back to the Kickstart window, click the button titled Clean Up. Kickstart removes the installation directory, restores your .htaccess file (if you had one in the first place), removes the backup archive and itself.

10. Click on the View the front-end button to visit your new site. You're done.

If you are restoring to a different subdirectory on the same server as the original site, or to a whole different host, you might need to edit your .htaccess file for your site to work properly. Also note that some third party extensions which store absolute filesystem paths, absolute URLs or contain host- or directory-specific settings may require manual reconfiguration after the restoration is complete. This is all described in the restoration section of this guide. If you need help backing up your site, take a look in the Backup Now section of this guide.

4. Server environment requirements

In order to work, Akeeba Backup requires the following server software environment:

- Joomla!™ and PHP version compatibilities are detailed in our Compatibility page [https://www.akeebabackup.com/compatibility.html].
- MySQL 5.0.42 or later. MySQL 5.6 or later recommended. MySQL 4.x is not supported.
- Minimum 24Mb of PHP `memory_limit` (sufficient *only* for smaller web sites, without many plug-ins and modules running). More is better. 32Mb to 64Mb recommended for optimal performance on large sites. 128Mb is recommended for sites containing deep-nested directories with thousands of files.

  Even though Akeeba Backup may run on servers with a smaller memory limit, it is unlikely that it will ever finish the backup process.
- Enough available free space or quota limit to store your backup archives.
- The cURL PHP module must be installed for FTP and cloud uploads to work.

As far as the browser is concerned, you can use any modern version (i.e. published within the last year) of Microsoft Edge, Safari, Opera, Firefox or Google Chrome. We no longer support Internet Explorer; our software will display incorrectly or not work at all on this old, buggy and obsolete browser.

In any case, you must make sure that Javascript is enabled on your browser for the backup to work. If you are using AVG antivirus, please disable its Link Checker feature (and reboot your computer) as it is known to cause problems with several Javascript-based web applications, including Akeeba Backup and its tools.

You are very strongly advised to disable Internet firewalls, antivirus applications and browser extensions which interfere with the site's loading such as script blockers (such as NoScript) and ad blockers (such as AdBlockPlus) *only for the domains you are backing up from and restoring to*. Remember that these applications and browser extensions are designed to protect you against third party sites. As a result they are very aggressive and WILL break your own sites. We can't do anything about it: your computer and your browser are under your control alone.

Chapter 2. Installation, updates and upgrades

1. Installing Akeeba Backup

Installing Akeeba Backup is no different than installing any other Joomla!™ extension on your site. You can read the complete instructions for installing Joomla!™ extensions on the official help page [https://docs.joomla.org/Installing_an_extension]. Throughout this chapter we assume that you are familiar with these instructions and we will try not to duplicate them.

1.1. Installing or manually updating the extension

Just like with most Joomla! extensions there are three ways to install or manually update Akeeba Backup on your site:

- Install from URL. This works only with the Professional release of our component. It is the easiest and fastest one, if your server supports it. Most servers do support this method.
- Upload and install. That's the typical extension installation method for Joomla! extensions. It rarely fails.
- Manual installation. This is the hardest, but virtually fail-safe, installation method.

Please note that installing and updating Akeeba Backup (and almost all Joomla! extensions) is actually the same thing. If you want to update Akeeba Backup please remember that you **MUST NOT** uninstall it before installing the new version! When you uninstall Akeeba Backup you will lose all your backup settings and all backup archives stored inside Akeeba Backup's directories (including the default backup output directory). This is definitely something you do not want to happen! Instead, simply install the new version on top of the old one. Joomla! will figure out that you are doing an update and will treat it as such, automatically.

Tip

If you find that after installing or updating Akeeba Backup it is missing some features or doesn't work, please try installing the same version a second time, without uninstalling the component. The reason is that very few times the Joomla! extensions installer infrastructure gets confused and fails to copy some files or entire folders. By repeating the installation you force it to copy the missing files and folders, solving the problem.

1.1.1. Install from URL

The easiest way to install Akeeba Backup Professional is using the Install from URL feature in Joomla!.

Important

This Joomla! feature requires that your server supports fopen() URL wrappers (`allow_url_fopen` is set to 1 in your server's `php.ini` file) or has the PHP cURL extension enabled. Moreover, if your server has a firewall, it has to allow TCP connections over ports 80 and 443 to `www.akeebabackup.com`, `www.akeeba.com` and `cdn.akeebabackup.com`. If you don't see any updates or if they fail to download please ask your host to check that these conditions are met. If they are met but you still do not see the updates please file a bug report in the official Joomla! forum [http://forum.joomla.org/]. In the meantime you can use the manual update methods discussed further below this page.

First, go to our site's download page for Akeeba Backup [https://www.akeebabackup.com/downloads/akeeba-backup.html]. Make sure you are logged in. If not, log in now. These instructions won't work if you are not logged in! Click on the All Files button of the version you want to install. Please note that the latest released version is always listed *first* on the page. On that page you will find both Akeeba Backup Core and Professional. Next to the

Professional edition's Download Now button you will see the Direct Install Link link. Right click on it and select Copy link address or whatever your browser calls this.

Now go to your site's administator page and click on Extensions, Manage. Click on the Install from URL tab. Clear the contents of the Install URL field and paste the URL you copied from our site's download page. Then click on the Install button. Joomla! will download and install the Akeeba Backup update.

If Joomla! cannot download the package, please use one of the methods described in this section of the documentation. If, however you get an error about copying files, folder not found or a cryptic "-1" error please follow our installation troubleshooting instructions [https://www.akeebabackup.com/documentation/troubleshooter/abinstallation.html].

1.1.2. Upload and install.

You can download the latest installation packages our site's download page for Akeeba Backup [https://www.akeebabackup.com/downloads/akeeba-backup.html]. Please note that the latest version is always on top. If you have an older version of Joomla! or PHP please consult our Compatibility page [https://www.akeebabackup.com/compatibility.html] to find the version of Akeeba Backup compatible with your Joomla! and PHP versions. In either case click on the version you want to download and install.

If you are not a subscriber, click on the Akeeba Backup Core to download the ZIP installation package of the free of charge version.

If you are a subscriber to the Professional release, please make sure that you have logged in first. You should then see an item on this page reading Akeeba Backup Professional. If you do not see it, please log out and log back in. Click on the Professional item to download the ZIP installation package.

All Akeeba Backup installation packages contain the component and all of its associated extensions. Installing it will install all of these items automatically. It can also be used to upgrade Akeeba Backup; just install it *without* uninstalling the previous release.

In any case, do not extract the ZIP files yet!

Warning

Attention Mac OS X users! Safari, the default web server provided to you by Apple, is automatically extracting the ZIP file into a directory and removes the ZIP file. In order to install the extension through Joomla!'s extensions installer you must select that directory, right-click on it and select Compress to get a ZIP file of its contents. This behaviour was changed in Mac OS X Mountain Lion, but people upgrading from older versions of Mac OS X (Mac OS X Lion and earlier) will witness the old, automatic ZIP extraction, behaviour.

Log in to your site's administrator section. Click on Extensions, Manage link on the top menu. Please click on the Upload Package File tab. Drag and drop the installation ZIP file you had previously downloaded to start the upload and the installation. After a short while, Joomla!™ will tell you that the component has been installed.

Warning

Akeeba Backup is a big extension (over 2Mb for the Professional release). Some servers do not allow you to upload files that big. If this is the case you can try the Manual installation or ask your host to follow our installation troubleshooting instructions [https://www.akeebabackup.com/documentation/troubleshooter/abinstallation.html] under "You get an error about the package not being uploaded to the server".

If you have WAMPServer (or any other prepackaged local server), please note that its default configuration does not allow files over 2Mb to be uploaded. To work around that you will need to modify your php.ini and restart the server. On WAMPserver left-click on the WAMP icon (the green W), click on PHP, php.ini. Find the line beginning with `upload_max_filesize`. Change it so that it reads:

```
upload_max_filesize = 6M
```

Save this file. Now, left-click on the WAMP icon, click on Apache, Service, Restart Service and you can now install the component. Editing the `php.ini` file should also work on all other servers, local and live alike.

If the installation did not work, please take a look at our installation troubleshooting instructions [https://www.akeebabackup.com/documentation/troubleshooter/abinstallation.html] or try the manual installation described below.

1.1.3. Manual installation

As of Akeeba Backup 5.0.0 this method can no longer be supported for technical reasons which have to do with the way Joomla! works when installing extensions of the type "package".

Warning

DO NOT UNZIP THE PACKAGE AND TRY TO INSTALL THE EXTENSIONS MANUALLY! This will very likely make your site fail with an error. When you are installing the package extension Joomla! makes a few checks to make sure that your server meets the minimum requirements. Moreover, the installation order in the package matters. It is designed to make sure that failure to install one of the included extensions will minimize the chance of a cascading effect which breaks your site.

1.1.4. The installation / update broke my site!

If your installation gets corrupt (e.g. Joomla didn't manage to successfully complete the installation or update of Akeeba Backup) you might find with a broken site. This is not a bug in our software. If critical files are missing or files from different versions are mixed up, any software will malfunction.

If you have access to your site's backend try to install the latest Akeeba Backup version's ZIP file twice in a row, without uninstalling the existing version before or in between.

Otherwise, use your FTP client to remove the following directories (some of them may not be present on your site; this is normal):

```
administrator/component/com_akeeba
component/com_akeeba
media/com_akeeba
plugins/quickicon/akeebabackup
plugins/actionlog/akeebabackup
plugins/system/backuponupdate
```

This will do the trick! You will now be able to access your site's administrator page again and retry installing Akeeba Backup without uninstalling it first. Remember, uninstalling Akeeba Backup will remove your settings and your backups; you do not want that to happen!

Note

If you get a username and password dialog *from your browser* (not Joomla!) OR a server error when you access your site's backend (administrator) URL, try deleting the `.htaccess` and `.htpasswd` files inside your site's administrator folder.

In some cases Joomla! forgets to install files for the FOF 3 library used by most of our components (Akeeba Backup, Admin Tools, Akeeba Ticket System and others). This could mean that even removing the directories above you could still be unable to access your site. If this happens, try the following solution:

1. Delete the folder libraries/fof30 from your site. **ATTENTION**! Do NOT remove the libraries/fof folder, it's something entirely different and you will break your site if you remove that folder instead!

2. Go to our Download page [https://www.akeebabackup.com/download.html] and download the latest version of FOF. This downloads a file named something like `lib_fof30-1.2.3.zip` on your computer.

3. Extract (unzip) the file you downloaded. You see a `fof` directory being extracted.

4. Rename to `fof` directory to `fof30`

5. Upload the `fof30` directory into your site's `libraries` directory.

6. You now have a `libraries/fof30` directory and you can log in to your site's backend.

7. Reinstall our extension *twice* in a row

2. Upgrading from Core to Professional

Upgrading from Akeeba Backup Core to Akeeba Backup Professional is by no means different than installing the component. You do not have to uninstall the previous version; in fact, you MUST NOT do that. Simply follow the installation instructions to install Akeeba Backup Professional over the existing Akeeba Backup Core installation. That's all! All your settings are preserved.

Important

When upgrading from Core to Professional you sometimes have to install the Professional package **twice**, without uninstalling anything in between. Sometimes Joomla! does not copy some of the files and folders the first time you install it. However, if you install the package again (without uninstalling your existing copy of Akeeba Backup) Joomla! copies all of the necessary files and performs the upgrade correctly.

3. Automatic updates

Akeeba Backup can be updated just like any other Joomla! extension, using the Joomla! extensions update feature. Please note that Joomla! is fully responsible for discovering available updates and installing them on your site. Akeeba Ltd does not have any control of the update process.

Note

This Joomla! feature requires that your server supports fopen() URL wrappers (`allow_url_fopen` is set to 1 in your server's `php.ini` file) or has the PHP cURL extension enabled. Moreover, if your server has a firewall, it has to allow TCP connections over ports 80 and 443 to `www.akeebabackup.com`, `www.akeeba.com` and `cdn.akeebabackup.com`. If you don't see any updates or if they fail to download please ask your host to check that these conditions are met. If they are met but you still do not see the updates please file a bug report in the official Joomla! forum [http://forum.joomla.org/]. In the meantime you can use the manual update methods discussed further below this page.

Warning

Akeeba Backup Professional needs you to set up the Download ID before you can install the updates. You can find your main download ID or create additional Download IDs on our site's Add-on Download IDs [http://akee.ba/downloadid] page. Then go to your site's administrator page and click on Components, Akeeba Backup, Options (in the toolbar). Click on the Live Update tab and paste your Download ID there. Finally, click on Save & Close.

You can access the extensions update feature in two different ways:

- From the icon your Joomla! administrator control panel page. You will find the icon in the left-hand sidebar, under the Maintenance header. When there are updates found for any of your extensions you will see the Updates are available message. Clicking on it will get you to the Update page of Joomla! Extensions Manager.

- From the top menu of your Joomla! administrator click on Extensions, Manager. From that page click on the Update tab found in the left-hand sidebar. Clicking on it will get you to the Update page of Joomla! Extensions Manager.

If you do not see the updates try clicking on the Find Updates button in the toolbar. If you do not see the updates still you may want to wait up to 24 hours before retrying. This has to do with the way the update CDN works and how Joomla! caches the update information.

If there is an update available for Akeeba Backup tick the box to the left of its row and then click on the Update button in the toolbar. Joomla! will now download and install the update.

If Joomla! cannot download the package, please use one of the manual update methods described below. If, however you get an error about copying files, folder not found or a cryptic "-1" error please follow our installation troubleshooting instructions [https://www.akeebabackup.com/documentation/troubleshooter/abinstallation.html].

If you get a white page while installing the update please try either the Built-in method (described above) or the manual update method (described below).

Updating manually

As noted in the installation section, installing and updating Akeeba Backup is actually the same thing. If the automatic update using Joomla!'s extensions update feature does not work, please install the update manually following the instructions in the installation section of this documentation.

Important

When installing an update manually you MUST NOT uninstall your existing version of Akeeba Backup. Uninstalling Akeeba Backup will always remove all your settings and any existing backup archives stored on your server. You definitely do not want that to happen!

Sometimes Joomla! may forget to copy some files when updating extensions. If you find Akeeba Backup suddenly not working or if you get a warning that your installation is corrupt you need to download the latest version's ZIP file and install it *twice* on your site, *without* uninstalling it before or in-between these installations. This will most certainly fix this issue.

If the error occurs again after a while, without you updating our software, please contact your host. Some hosts will delete or rename files automatically and without any confirmation as part of a (broken and unfit for purpose) "malware scanner / antivirus". Unfortunately, these scanners return a lot of false positives -innocent files mistakenly marked as malicious- but rename / delete them nonetheless, breaking software installed on the server. If you are on such a host we very strongly recommend that you move to a decent host, run by people who actually know what they are doing. It will be far less headache for you and would actually improve your site's security.

4. Requesting support and reporting bugs

Support can be provided only to subscribers and only through our site's Support section. If you already have an active subscription which gives you access to the support for Akeeba Backup you can request support for it through our site. You will need to log in to our site and go to Support, Akeeba Backup for Joomla! and click on the New Ticket button. If you can't see the button please make sure you have an active subscription that gives you access to Akeeba Backup for Joomla! support. If you do and still don't see the button please use the Contact Us page to let us know of the ticket system problem and remember to tell us your username.

If you want to report a bug, please use the Contact Us page of our site. You don't need to be a subscriber to report a bug. Please note that unsolicited support requests sent through the Contact Us page will not be addressed. An issue is not a bug unless it can be reliably reproduced *on multiple sites and servers*. Please make sure you include clear instructions on reproducing the issue. If the issue cannot be reproduced it's not a bug report, it's a support request.

Important

Support cannot be provided over Twitter, Facebook, email, Skype, telephone, the official Joomla! forum, our Contact Us page or any other method except the Support section on our site. We also cannot take bug reports over any other medium except the Contact Us page and the Support section on our site. Support is not provided to non-subscribers; if you are using the Core version you can request support from other

users in the official Joomla! forum or any other Joomla!-related forum in your country/region. We have to impose those restrictions in support to ensure a high level of service and quality. Thank you for your understanding.

Chapter 3. Using the Akeeba Backup component

In this chapter you are going to find detailed reference of all the pages, options and features of the Akeeba Backup components. To get things organized in a logical manner, we chose to present the individual pages in the same manner they appear on the component's Control Panel page, i.e. the first page which is presented to you when you launch the component's back-end. Some of the pages are not available as Control Panel icons, but from different areas of the component. These are discussed first.

1. Menu items

Joomla! 3.7 and later versions allow you to create custom administrator menus. Akeeba Backup fully supports this new feature by providing custom menu item types.

Most of these custom menu item types were created with site integrators / web site agencies in mind. Typically you want to offer your client a simple, obvious way of doing backup operations (take, restore or transfer backups). Up until now you had to tell them to go to the quite busy Akeeba Backup page and click on just the one thing you want them to. As we all know, clients get distracted and start changing things they shouldn't be touching. The custom menu types below are designed to offer perfectly tailored access to the component areas that most users need. Taking and restoring a backup can become a no-brainer, reduced to simply clicking on a back-end menu item.

1.1. Control Panel

This menu item type lets you access Akeeba Backup's main page (control panel). This is the same menu item type Joomla! creates by default when you install the component.

Please remember that excluding files, folders and database tables as well as including external folders and additional databases (for the Professional edition) can only be done through the Control Panel page. It's always a good idea having a link of this type in your custom menu.

1.2. Backup

This menu item type allows the users to take backups. The default options let this work just like clicking on the Backup Now icon in Akeeba Backup's Control Panel page, i.e. the user can select an alternative backup profile, enter a backup description and/or comment and then take a backup or change their mind and return back to the Control Panel page. However the additional options let you do more interesting stuff.

The available options are:

Force backup profile
: Select the backup profile which will be pre-selected in drop-down of the Backup Now page. Selecting (None) default to the currently active backup profile, as selected in other pages of the Akeeba Backup component. By default that's profile #1. This is especially useful with the Start immediately option below.

Start immediately
: When enabled the backup will start right away, without asking the user to enter a backup description or comment and without the option to change their mind. This is equivalent to using the One Click Backup feature inside Akeeba Backup.

  We strongly recommend using this with the Force backup profile option above. Use it to set up which profile you want the backup to be taken with. This allows you to set up one-click backup menu items.

Hide toolbar
: When this option is disabled the user will see the Control Panel and Help buttons at the top of the page. The former will take them back to Akeeba Backup's main page whereas the latter opens the documentation page for the Backup Now page. If you are setting up a one-click

backup menu item with the options above it's a good idea to enable this option to hide these buttons. That's especially useful when you are setting up a simple menu for use by your client and you don't want them to accidentally cancel the backup by clicking on these buttons.

Return URL — Set up an internal URL to redirect the user after a successful backup. An "internal URL" is a URL pointing to a page in your site's administrator area, *without* the domain name and `/administrator/` part of it. For example, to take someone back to the Joomla! main page set this to `index.php` without anything else before or after it. To take someone back to Akeeba Backup's main page set this to `index.php?option=com_akeeba`.

Warning

Due to the way Joomla's menu manager works, it expects the URL to be URL-encoded. This means that question marks must be replaced %3F and so on. Don't worry about it. Enter the URL regularly and save the menu item **twice** in a row. We have employed a trick to force URL-encoding of the value when re-saving the menu item. Unfortunately due to a missing feature in Joomla's API we can't employ the same or a similarly clever trick the `<first>` time you save the URL.

1.3. Configuration

This menu item type allows the users to modify the main configuration of the current backup profile. It's equivalent to pressing the Configuration button in Akeeba Backup's main page.

1.4. Manage Backups

This menu item type allows the users to manage backup attempts. This includes viewing all backup attempts, viewing / changing the backup description and comments, have access to logs, download the backups, manage remotely stored backups and restore any of the past backups (as opposed to only the latest backup). It's equivalent to pressing the Manage Backups button in Akeeba Backup's main page.

1.5. Restore Latest Backup

This menu item type allows the users to restore the latest backup taken with the specified backup profile. This is especially useful if you teach your site administrators (or the clients for whom you're building sites) to take a backup right before trying to do something which could go wrong such as updating a component, changing configuration settings or doing batch operations on content.

The only option is **Backup Profile** which lets you choose which backup profile's latest backup attempt will be restored. Idea: use the same profile you've set up in a menu item of the Backup type that you've told the client to always use before any dangerous operation. This way you can offer your clients an easy way to undo their most common mistakes!

1.6. Transfer Site Wizard

This menu item type allows the users to transfer and restore the latest backup on a different server. It's equivalent to pressing the Site Transfer Wizard button in Akeeba Backup's main page.

Idea: you can train your clients to use this to deploy a site from the staging to the live server.

1.7. What to do if you don't have any menu items to Akeeba Backup

Depending on how you've set up your site's administrator menu and/or if you've hit a Joomla! bug that sometimes occures on extension update you may end up without a menu item to Akeeba Backup. Other times you may have

deliberately chosen not to display a menu to Akeeba Backup to keep clients from changing the backup settings. The question remains. How can you access Akeeba Backup and how can you restore menu items manually?

The following instructions are generic Joomla! usage tips and don't have to do with how our software works. We provide them as a courtesy. If these instructions don't work for you please do not contact Akeeba Ltd for support. We cannot offer support for generic Joomla! use. Instead please do ask for help in the Joomla support forum at `http://forum.joomla.org`.

Accessing Akeeba Backup

You can always access Akeeba Backup by visiting the `/administrator/index.php?option=com_akeeba` URL on your site, *after* logging in to your site's back-end.

That is to say, if your site's administrator URL is `http://www.example.com/administrator/index.php` enter the URL `http://www.example.com/administrator/index.php?option=com_akeeba` in your browser's address bar to access Akeeba Backup.

Restoring Joomla's default administrator menus

Important

These instructions only work on Joomla! 3.7 and later and only with the default administration template supplied with Joomla. If you have a third party administrator template please contact the template's developer for instructions regarding missing menu items or reverting to the Joomla! default administrator menu.

You need to access the `/administrator/index.php?option=com_modules` URL on your site, *after* logging in to your site's back-end.

From the drop-down that currently reads `Site` select the option `Administrator`.

Find the module which displays your administrator menu. Usually it's called Admin Menu. Click on it to edit it.

From the Menu To Show drop-down select Use System Preset. Then click on Save & Close.

2. Pages outside the Control Panel panes

2.1. Common navigation elements

All pages have their title displayed above their contents. On the tool bar there is a Control Panel icon. Clicking it will bring you back to Akeeba Backup's Control Panel (the first page of the component, with all the buttons).

On pages where editing takes place (e.g. the Configuration page, the profiles editor, etc) instead of the Control Panel icon there is a Cancel icon which discards any changes made and returns you to the previous page. On those pages you will also find a Save & Close icon which saves settings and returns you to the previous page, as well as a Save icon which saves settings and returns you to the same editing page.

On the bottom of each page, just above the Joomla!™ footer, there is the license information. On the Control Panel page of the Akeeba Backup Core editions there is also a donation link appearing on the right sidebar; if you feel that Akeeba Backup was useful for you do not hesitate to donate any amount you deem appropriate.

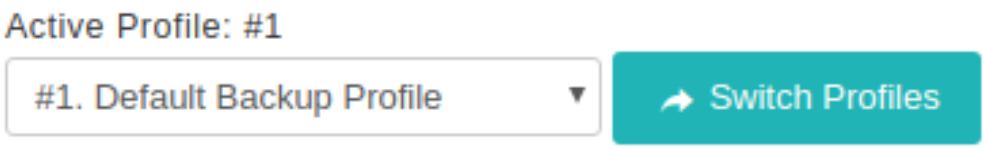
2.2. The Control Panel

The main page which loads when you click on Components, Akeeba Backup is called the Control Panel screen. From here you can see if everything is in working order and access all of the component's functions and

configuration options. If any problems or configuration issues are detected, Akeeba Backup will report one or more error or warning messages.

If you see a blank page instead of the Control Panel, you may have a very old version of PHP installed on your server. Please check the minimum requirements of your currently installed Akeeba Backup version. Akeeba Backup will try to detect incompatible PHP versions but this is not always possible.

The profile selection box

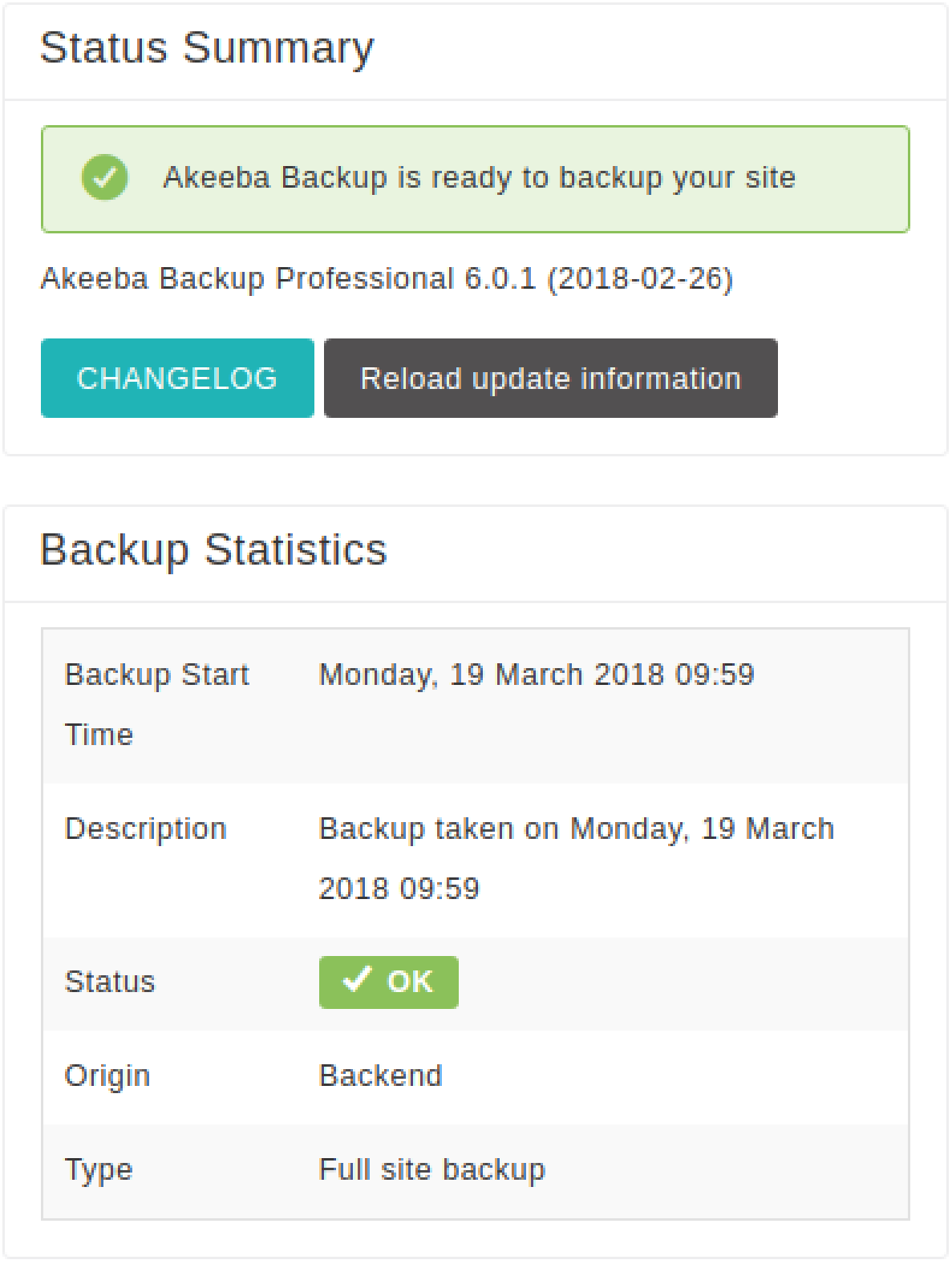


Towards the top of the page, there is the profile selection box. It serves a double purpose, indicating the active profile and letting you switch between available profiles. Clicking on the drop down allows you to select a new profile. Changing the selection (clicking on the drop down list and selecting a new profile) automatically makes this new profile current and Akeeba Backup notifies you about that. Should this not happen, you can manually click on the Switch Profile button on the right to forcibly make the selected profile current.

Tip

The active profile is applied in all functions of the component, including configuration, filter settings, inclusion options, etc. The only settings which are not dependent on the active profile are those accessible from the Options toolbar button. Keep this in mind when editing any of Akeeba Backup's settings!

On the right hand side of the page, you will find a column with useful information.



There are two areas:

Status Summary — In this area you can find information regarding the status of your backup output directory. Akeeba Backup will warn you if this directory is unwritable. If the text reads that there are potential problems you **must** take a look at the details below to find out what these might be!

Important

It is possible that your host enforces open_basedir restrictions which only allow you to have an output directory under a handful of predefined locations. On this occasion, Akeeba Backup will report the folder unwritable even though you might have enforced 0777 (read, write and execute allowed for all) permissions. These restrictions are reported in the section below the overall status text as an item entitled "open_basedir restrictions".

If any potential problems have been detected, right below the overall status you will find one or several warnings links. Just click on each warning's description to get a pop up window explaining the potential problem, its impact on your backup and precautionary or corrective steps you can take. If this section is empty, no detectable problems were found; this is a good thing, indeed!

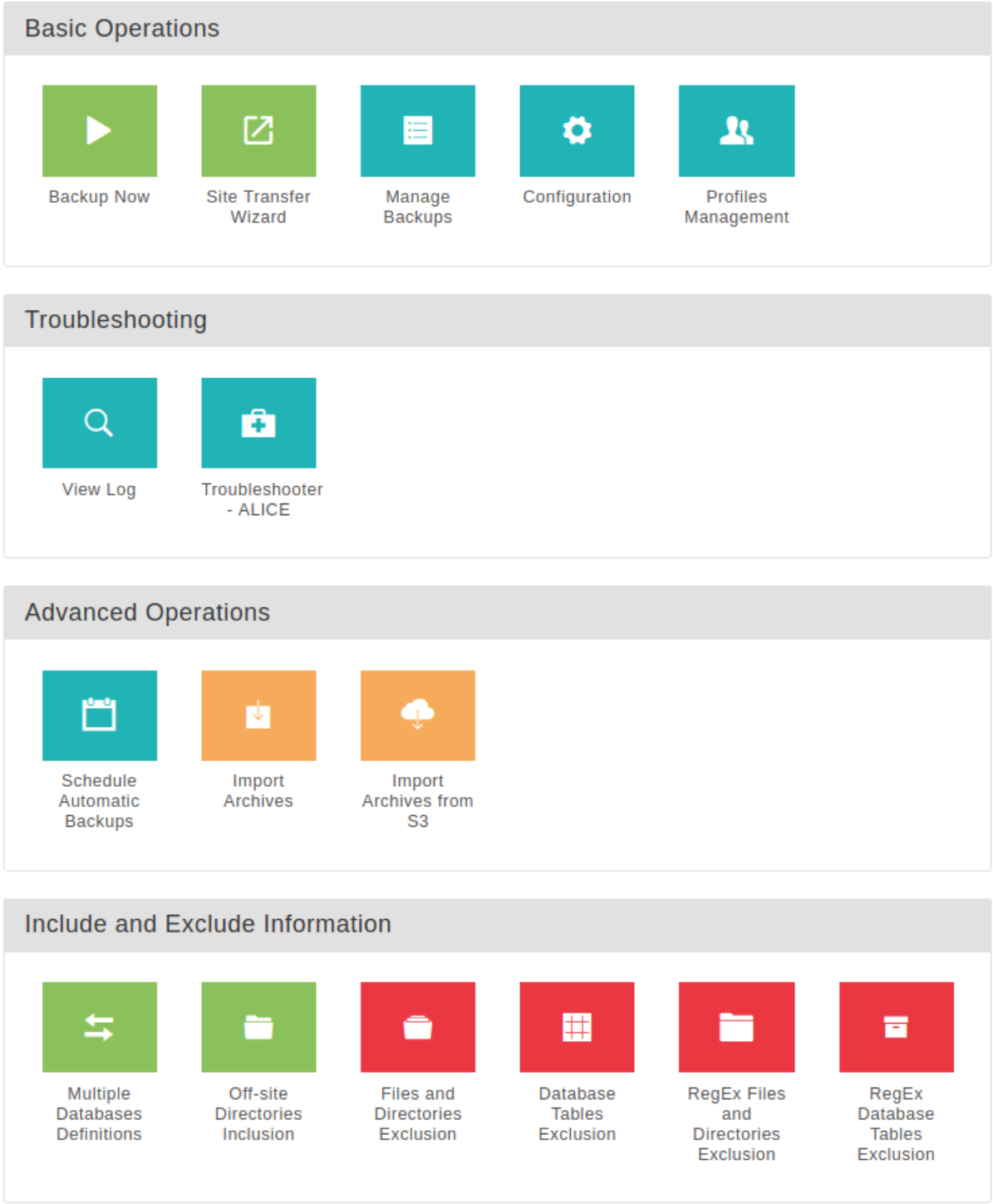
If you see any problems reported please read the full text of the warnings by clicking on each item. It contains instructions to solve the issue. These are the same instructions our support staff will provide as a first response to these issues.

If you are a Core (free version) user, you will see a donation link. If you feel that Akeeba Backup has helped you - and you do not wish or can't afford subscribing to the Professional edition - you can donate a small amount of money to help us keep the free version going. Thank you!

Backup Statistics

This panel informs you about the status of your last backup attempt. The information shown is the date and time of backup, the origin (e.g. remote, backend, frontend and so on), the profile used and the backup status.

The main navigation panel set



The main navigation panel set allows access to the different functions of the component. You can access them by clicking on the respective each icon. Please note that the screenshot in this documentation displays the Professional version. If you are using the Core version you will have fewer options.

Depending on your backup profile settings, at the top of this area you may find a series of buttons under the header **One-click backup**. Clicking one of these buttons will start a backup with the corresponding backup profile, without asking you for confirmation.

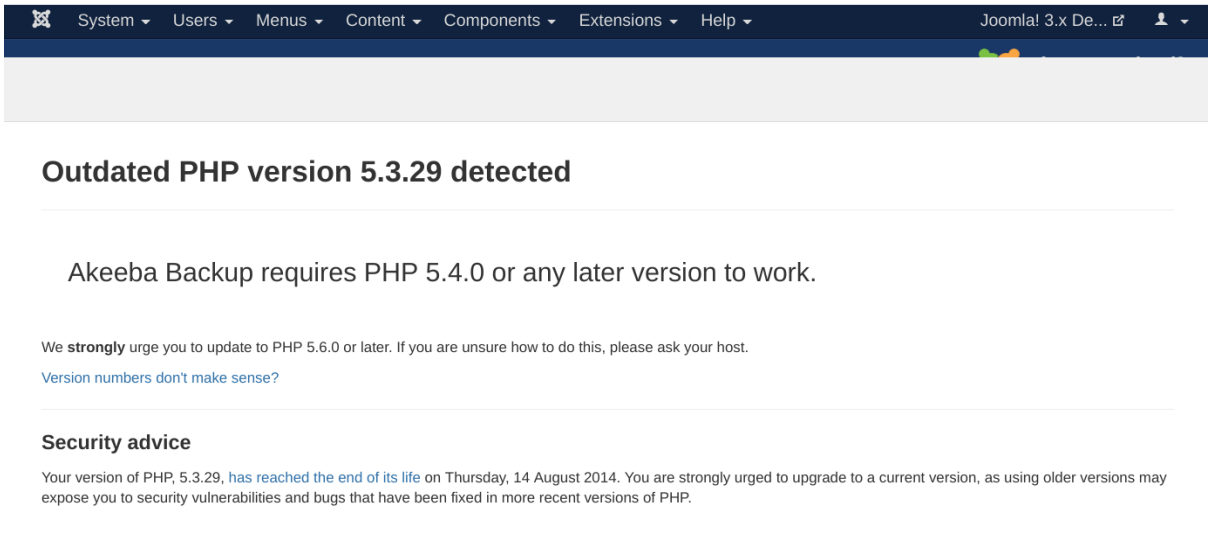
Finally, you can edit the global options of Akeeba Backup by clicking on the Options button towards the top right hand of the page, in the Joomla! toolbar area at the top of the page.

2.2.1. Warning and error messages in the Control Panel

Full page error messages

In the unlikely event that your server has a major configuration issue, e.g. using an outdated PHP version or PHP module, which prevents you from running our software *at all* instead of the Control Panel page you will see a full screen error message. The page will tell you what the problem is and how to fix it. For example, if you have an outdated PHP version:

Outdated PHP error page



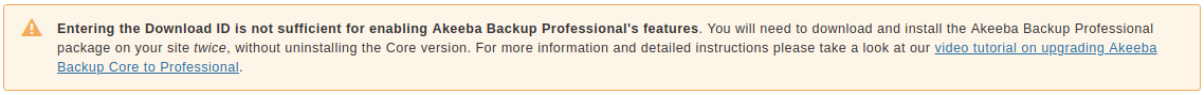
Download ID messages

The Download ID message in the Professional

If you are using Akeeba Backup Professional for Joomla! you will be asked to enter your Download ID. This is necessary to receive updates to the software. Without it you will be notified for updates but you will not be able to install them. Click on the link in the message to log in to our site and receive personalized instructions for entering the Download ID.

> ## Tip
>
> You can create Add-On Download IDs free of charge on our site. Just log into our site and click on the "add-on download IDs" link below the header. You can have a different Download ID for each of your clients. If the client stops working with you, you will be able to unpublish (deactivate) their Download ID.

The Download ID message in the Core version

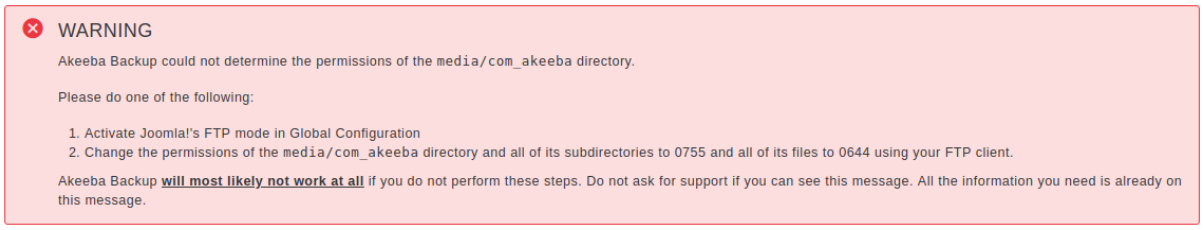


Conversely, if you are using the Akeeba Backup Core for Joomla! but have entered a Download ID you will receive an error message reminding you that this is not the proper way to upgrade to the Professional version. Instead, you

should install the Professional version on top of the Core version. If you are not sure how to do this please click the link at the end of the message. It takes you to a video tutorial telling you how to do this.

Media files' permissions

Media folder permissions



If Akeeba Backup detects a problem with the permissions of the media folder, where its JavaScript, CSS and image files are stored, it will try to automatically do the necessary changes for you. It requires that you have provided FTP connection information to your site's Global Configuration and enabled the FTP option in that page.

If these changes cannot be done automatically it will display this error message. Please follow the instructions in the message.

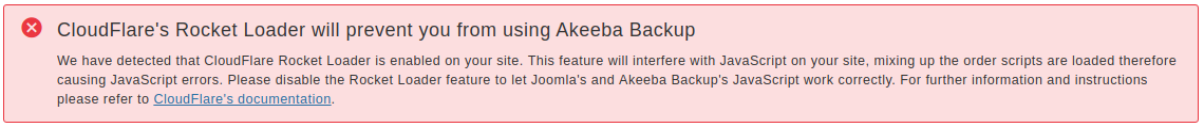
If you have already followed the instructions in the message but the interface behaves erratically or appears "broken" one of your system plugins is killing Akeeba Backup's JavaScript. Check your browser's developer tools to see which third party JavaScript is causing that. If you can't figure it out yourself please contact us and give us Super User and FTP access to your site so we can help you.

Tip

Due to the way this warning works you may see a yellow or red flash in the Control Panel, Configuration or Backup Now pages. This is normal and nothing to worry about. It's just your browser being faster in rendering the page than Javascript files loading from your server.

CloudFlare RocketLoader

CloudFlare RocketLoader warning

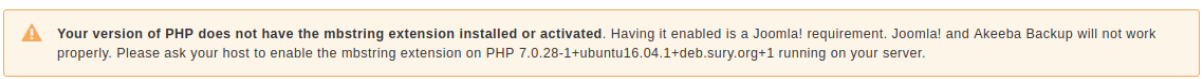


CloudFlare's RocketLoader changes the load order of the JavaScript on every page of your site, deferring the loading of every file at the end of the page load. Unfortunately, this causes applications depending on JavaScript, like Akeeba Backup *or even Joomla! itself*, to fail due to no fault of their developers.

If Akeeba Backup detects that you are using CloudFlare's RocketLoader it will try to apply a workaround which prevents RocketLoader from mangling Akeeba Backup's JavaScript. This may NOT work on all sites. In case the automatic workaround fails you see this error message. Please follow the link at the end of the message for instructions to manually fix this problem.

Missing mbstring

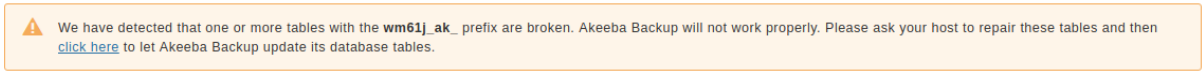
Missing mbstring warning



Akeeba Backup, like Joomla! itself, requires the PHP extension called "mbstring" to be loaded and activated. Without it it is impossible to handle extended characters and find the length of binary data. Therefore, if mbstring is missing your backup will fail. Please ask your host to enable mbstring on your site. The PHP version for which mbstring needs to be activated, as reported by PHP running on your server, is printed on the message on your screen. Please copy the message from *your* site to your host's support – do not copy the version displayed in the example screenshot above.

Broken database

Broken database warning



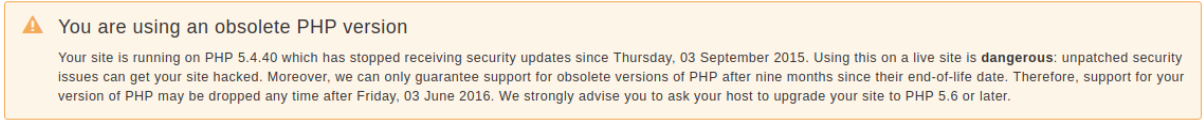
Akeeba Backup checks the structure of its database tables every time you visit the Control Panel page. If the structure is out of date it tries to fix them. If it fails it prints the message above. The possible reasons for failure are:

- Broken tables. Per the message, ask your host to fix the table with the prefix printed in the error message on your site (NOT the one in the documentation; the documentation is just an example).
- Insufficient database user permissions. Your database user does not have the necessary permissions to modify the database tables. Please contact your host and ask them to help you.
- Using HHVM instead of PHP. You are using HHVM instead of PHP proper. HHVM is actually incompatible with PHP, it causes problems and we do NOT support it. Please use PHP 7 instead. PHP 7 is as fast as -and in many cases *much faster than*- HHVM without the incompatibility issues. Our software is fully compatible with PHP 7 and we recommend it for daily use. If you are not sure how to check that / how to switch to PHP 7 please contact your host.
- PHP is compiled with a very old version of the MySQL connector library. Please contact your host and have them check that if all else fails.
- Custom / unsupported Joomla! database driver. If you don't know what this means it's probably not the case and you should stop reading this paragraph now. If you know what it means please do check that your custom database driver class talks to a MySQL compatible server (MySQL, Persona or MariaDB) and returns a name that contains the string "mysql" (case insensitive). Otherwise Akeeba Backup won't know how to install or update its tables.

After fixing the problem visit Akeeba Backup's Control Panel page again. If you see the message again, click the link in the message to have Akeeba Backup retry fixing its tables' structure.

Obsolete PHP version

Obsolete PHP version



Akeeba Backup always checks the PHP version it runs under. If your PHP version is very old and declared end-of-life (EOL) by the developers of PHP we will warn you. EOL versions of PHP have known bugs which prevent software from running correctly, slow, and insecure. Even if your Linux distribution's vendor claims to still support them, the fact remains that major security and functional flaws are NOT fixed.

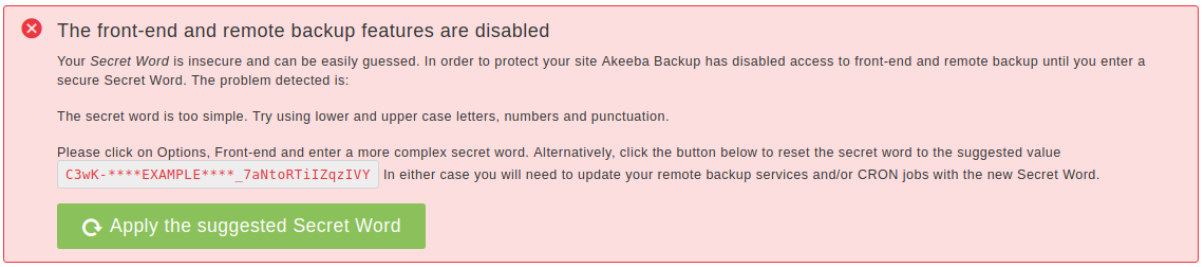
Because of these reasons, Akeeba Ltd only officially supports running our software on the versions of PHP which are still under active maintenance per the official PHP site [http://be2.php.net/supported-versions.php]. We only

guarantee support End Of Life versions of PHP for 3 to 6 months after their End Of Life date as published in the official PHP site [http://www.php.net/eol.php].

If your PHP version is out of date you will see a message similar to the one above, with the relevant PHP version, EOL and end of support dates printed in it. There is no reason to stick with an old PHP version. Upgrade your PHP: your site will be more secure and faster. As a bonus, your site will rank better in search engines: slow sites are penalized in search results since 2014.

Front-end backup Secret Word

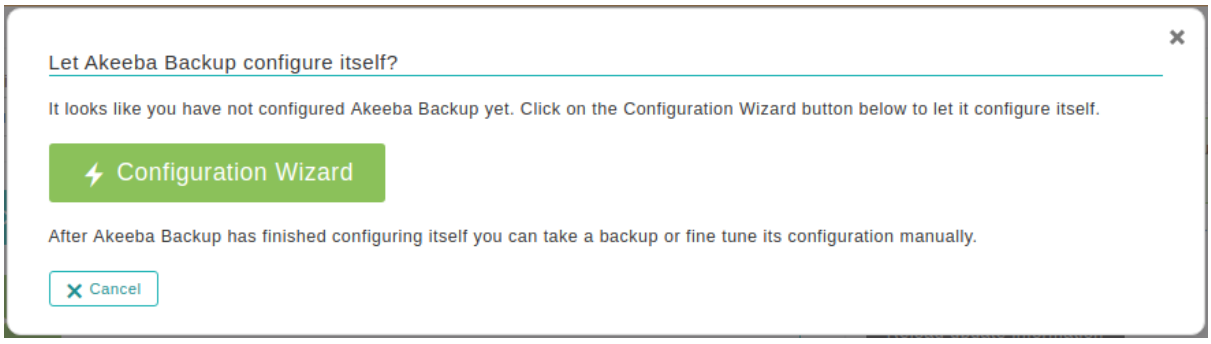
Front-end backup Secret Word



If you have enabled the legacy front-end backup feature, Akeeba Backup checks the quality of the Secret Word. If it's found too simple / easily guessable it will decline to run front-end and remote backups until you use a more secure Secret Word. This is a security precaution: an easily guessable Secret Word could be used to launch a Denial of Service attack to your site or steal information from it. If you are not sure how to create a secure secret word click the large button to apply an automatically generated, secure secret word.

Configuration Wizard

Configuration Wizard notice



If Akeeba Backup detects that you have a brand new backup profile that has not been configured yet it will ask you to run the Configuration Wizard. The Wizard runs without requiring any input from you. Sit back and let Akeeba Backup figure out what are the best backup settings for your site. We recommend all of our users to use the Configuration Wizard as it prevents the most common backup problems you may encounter.

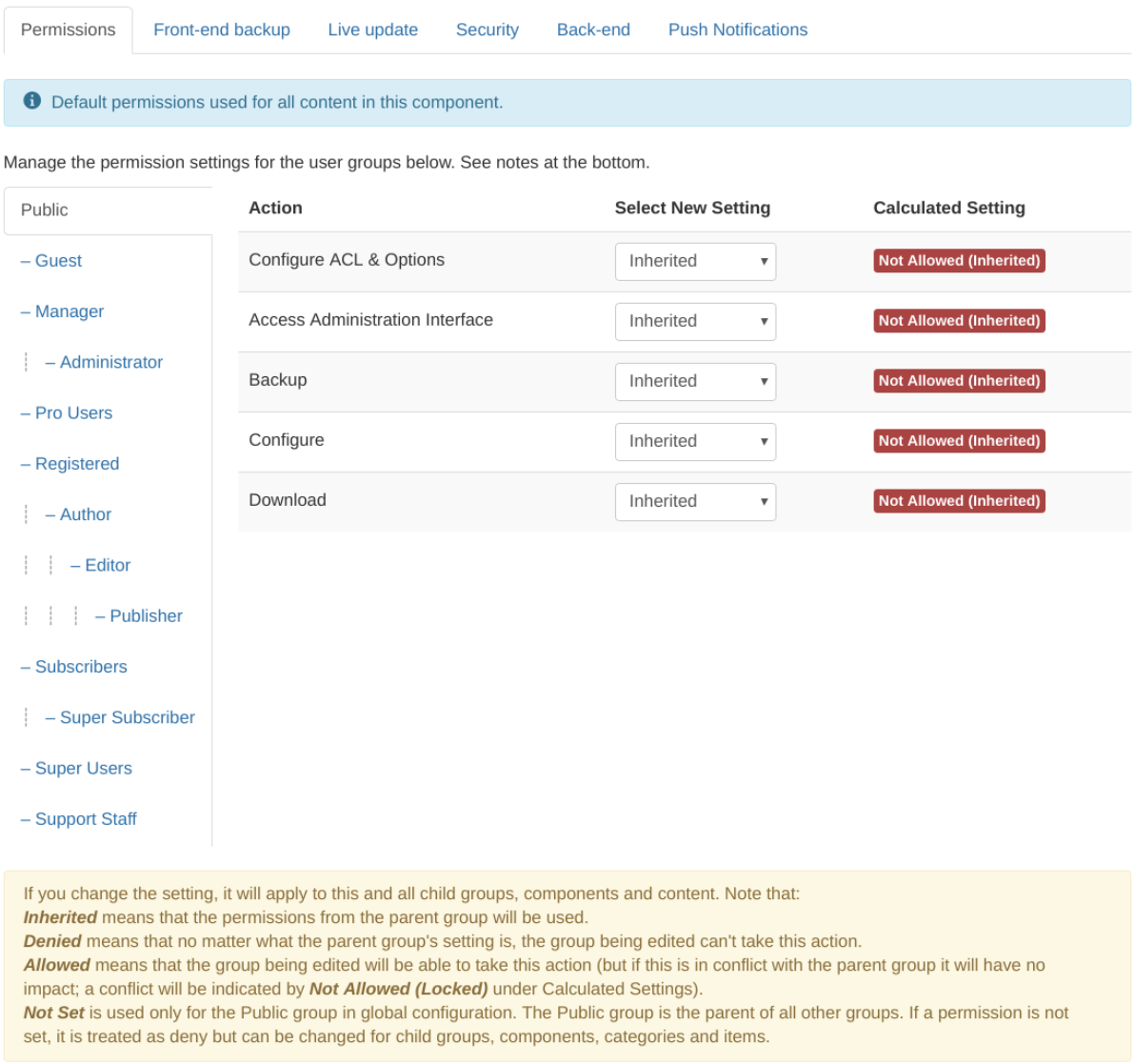
2.2.2. Editing the component's Options

You can edit the global, component-wide options by clicking on the Options button towards the top right hand of the page, in the Joomla! toolbar area at the top of the page. The Options editor opens in a new page.

Please note that the component Options are component-wide and take effect regardless of the active profile.

There are several tabs:

Permissions



This is the standard Joomla! ACL permissions setup tab. Akeeba Backup fully supports supports Joomla! ACLs and uses the following three custom permissions:

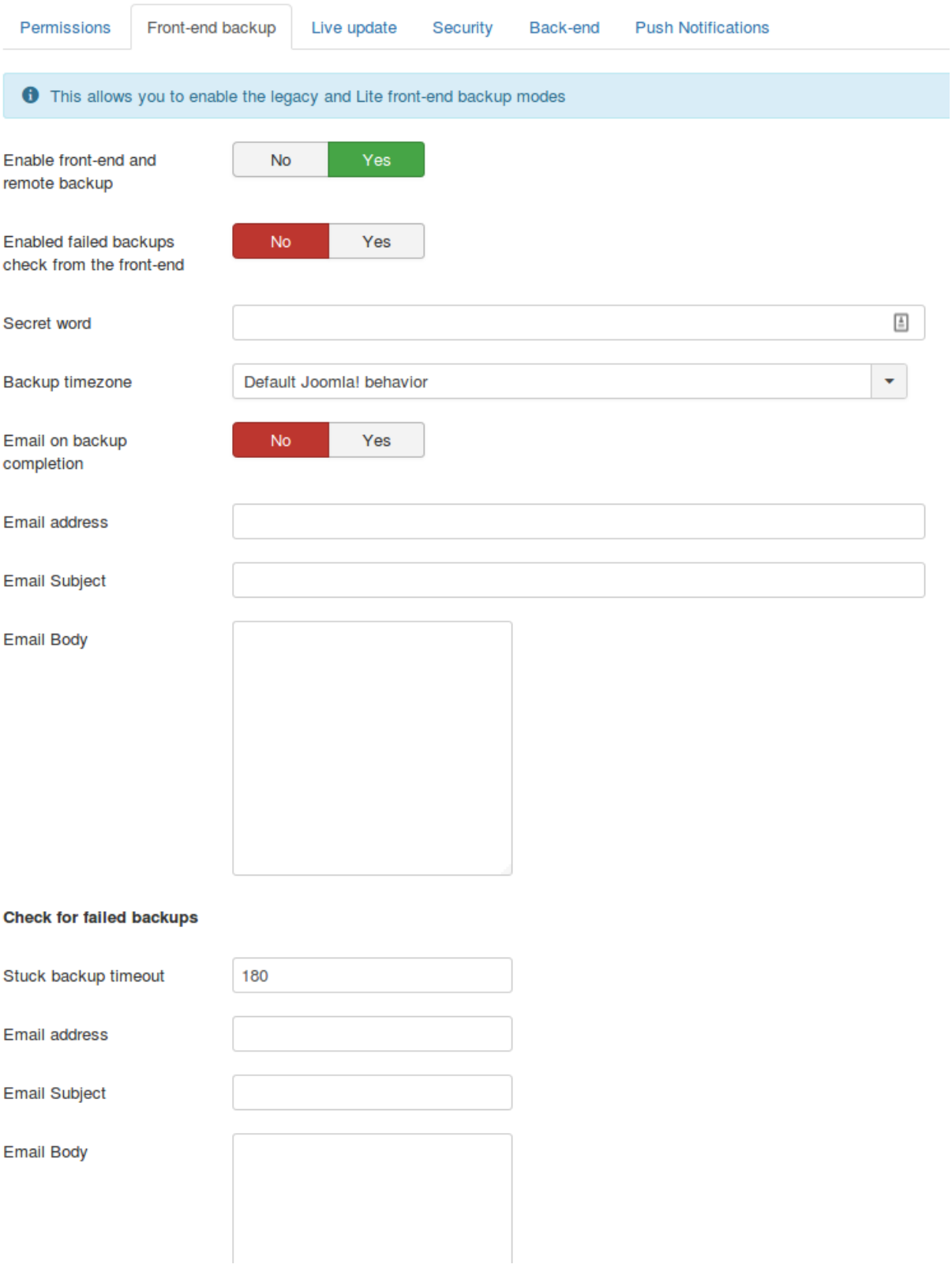
Backup Now — Allows the users of the group to take backups.

Configure — (The second one displayed in each group) Allows the users of the group to access the Configuration page, as well as all features which define what is included/excluded from the backup

Download — Allows the users of the group to download backup archives from the Manage Backups page.

Front-end backup

Here you can define options which affect front-end, CRON and remote backups.



Enable front-end and remote backup

Akeeba Backup allows you to take backups from the front-end, or from compatible remote clients (e.g. Akeeba Remote CLI and other third party products or services). In order to be able to do so, you have to enable this option.

Enable failed backups check

Akeeba Backup allows you to access a special URL on the front-end of the site. This lets it detect failed / stuck backups and send you email notifications about them. In order to be able to do so, you have to enable this option.

from the front-end

Secret word

Required to authenticate a remote backup method. Also protects the front-end backup feature from Denial of Service attacks by requiring you to pass this secret word in the front-end backup URL.

Please note that if you use any character other than a-z, A-Z and 0-9 you MUST NOT use the secret word verbatim in the front-end backup URL. Instead, you have to URL-encode it. The Schedule Automatic Backups page does that automatically for you. Just go to Components, Akeeba Backup, click Schedule Automatic Backups, scroll all the way down and use one of the tabs to get the URL or command line you need to use with the secret word properly encoded in the URL.

For security reasons, you must use a complex enough secret word. Akeeba Backup enforces that by disabling the front-end backup feature and the JSON API if you are using a Secret Word with a low complexity. We strongly recommend using a "secret word" consisting of at least 16 random, mixed case alphanumeric characters. It should not be a dictionary word or based off a dictionary word. One good resource for truly random secret words is Random.org's password generator [https://www.random.org/passwords/?num=1&len=24&format=html&rnd=new].

Note

Why is this field not a password field? The Secret word is transmitted in the clear when you load the page and is also visible when you view the source of the page or right click on the field and choose Inspect Element. In other words, as long as someone has access to the component configuration page they can trivially find out the secret word. Not to mention that the secret work is also plainly visible in the Schedule Automatic Backups page. Always use HTTPS with a commercially signed SSL certificate when configuring or backing up your site.

Backup timezone

The timezone which will be used for all of the naming variables processed by Akeeba Backup. These are variables such as [DATE] and [TIME] which you can use in the filename template of backup archives, the backup output directory name, the front-end and remote backup emails and elsewhere.

The default option is called `Default Joomla! behavior` and it will find out the timezone the same way Joomla! does: if there is a logged in user it will use their timezone. If there is no logged in user (e.g. front-end or remote backup) or there is a logged in user but they do not have a timezone set in their user profile the Server Timezone used in the site's Global Configuration will be used instead. If that is not set it will fall back to GMT (Greenwich Mean Time).

We recommend setting this option to the timezone the people responsible for taking and restoring backups are most familiar with. If you are the only Super User use the timezone where you normally live in.

Email on backup completion

When enabled, Akeeba Backup will send an email regarding the backup status every time a front-end or remote backup is complete or failed.

Email address

When the above option is enabled, the email will be sent to this email address. If you leave it blank, Akeeba Backup will send a copy of the email to all Super Administrators of the site.

Email subject

This option lets you customise the subject of the email message which will be sent when a remote, CRON or front-end backup succeeds. You can use the same variables you can use in file names, i.e. [HOST] for the domain name of your site and [DATE] for the current date and time stamp. Leave blank to use the generic default option.

Email body — This option lets you customise the body of the email message which will be sent when a remote, CRON or front-end backup succeeds. Leave blank to use the generic default option. The email is delivered as plain text; you may not use any HTML to format it. You can use the same variables you can use in file names, i.e. [HOST] for the domain name of your site and [DATE] for the current date and time stamp, inside the body text. Moreover, you may also use any or all of the following variables in order to enhance the clarity of your message:

- [PROFILENUMBER] The numeric ID of the current backup profile
- [PROFILENAME] The description of the current backup profile
- [PARTCOUNT] The number of archive parts of the backup archive which was just generated
- [FILELIST] A list of filenames of the archive parts of the backup archive which was just generated
- [REMOTESTATUS] Available since Akeeba Backup 3.5.3. Shows the status of post-processing, e.g. uploading the file to remote storage like Amazon S3. If you are not using post-processing, this is always empty. If the transfer to the remote storage was successful it will output "Post-processing (upload to remote storage) was successful". If the transfer fails it will output "Post-processing (upload to remote storage) has FAILED".

The options under Check for failed backups are used with the feature for checking for failed backups automatically.

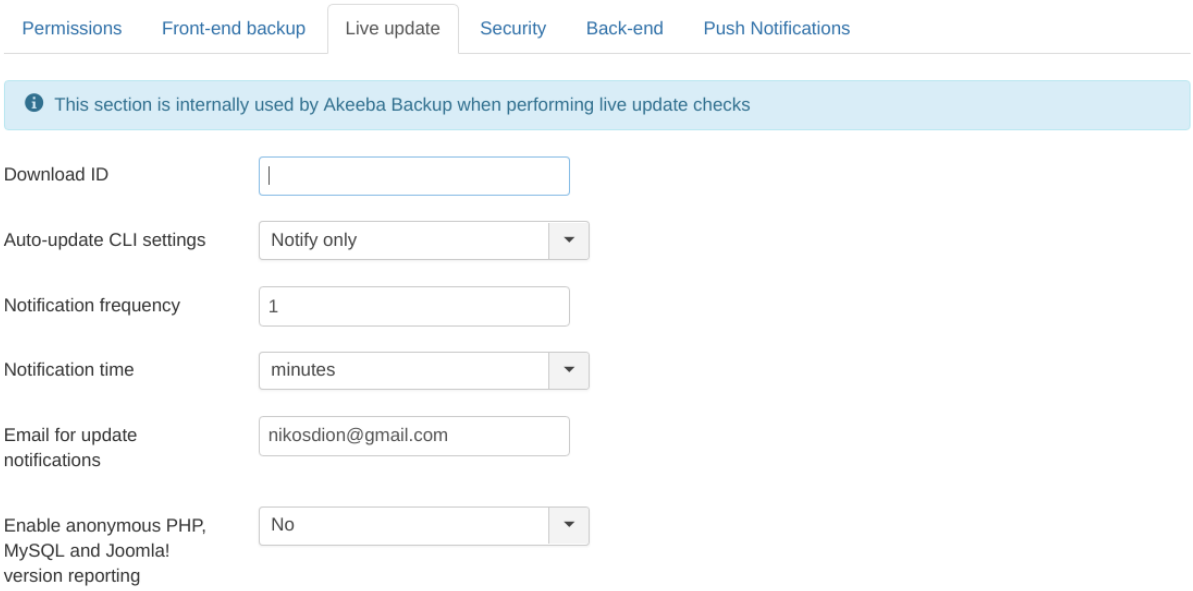
Stuck backup timeout — A backup will be considered stuck (failed) after this many seconds of inactivity. Please note that uploading backup archives to remote storage, such as Amazon S3, using the native CRON mode might take substantially longer than that. We advise you to leave this value as is and schedule the backup failure checks to take place a substantial amount of time (e.g. 1 hour) after the expected end time of your scheduled backups. If a backup failure check takes place before a backup has finished it is very possible that you will end up with a failed backup!

Email address — The email address which will be notified for failed backups

Email subject — Leave blank to use the default. You can use all of Akeeba Backup's variables you can use for naming archive files, e.g. [HOST] and [DATE]

Email body — Leave blank to use the default. You can use all of Akeeba Backup's variables you can use for naming archive files, e.g. [HOST] and [DATE].

Live update



These options define how Akeeba Backup will notify you regarding available updates

Download ID
: If and only if you are using the Professional release you have to specify your Download ID for the live update feature to work properly. You can get your Download ID by visiting AkeebaBackup.com and clicking My Subscriptions. Your Download ID is printed below the list of subscriptions. Filling in this field is required so that only users with a valid Professional subscription can download update packages, just as you'd expect from any commercial software.

Note

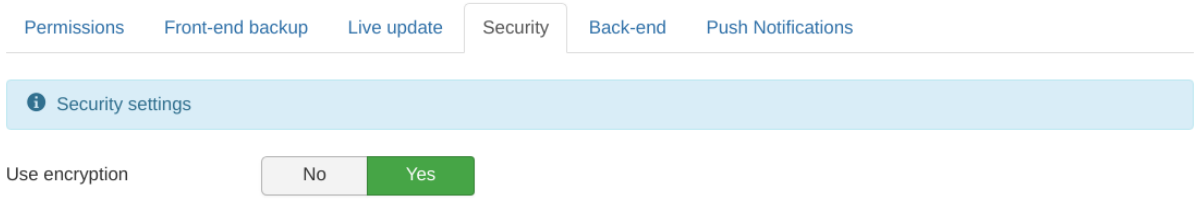
Users of Akeeba Backup Core do not need to supply this information.

Enable anonymous PHP, MySQL and Joomla! version reporting
: Help us improve our software by anonymously and automatically reporting your PHP, MySQL and Joomla! versions. This information will help us decide which versions of Joomla!, PHP and MySQL to support in future versions.

Note: we do NOT collect your site name, IP address or any other directly or indirectly unique identifying information.

Security

These options define how Akeeba Backup will secure your settings



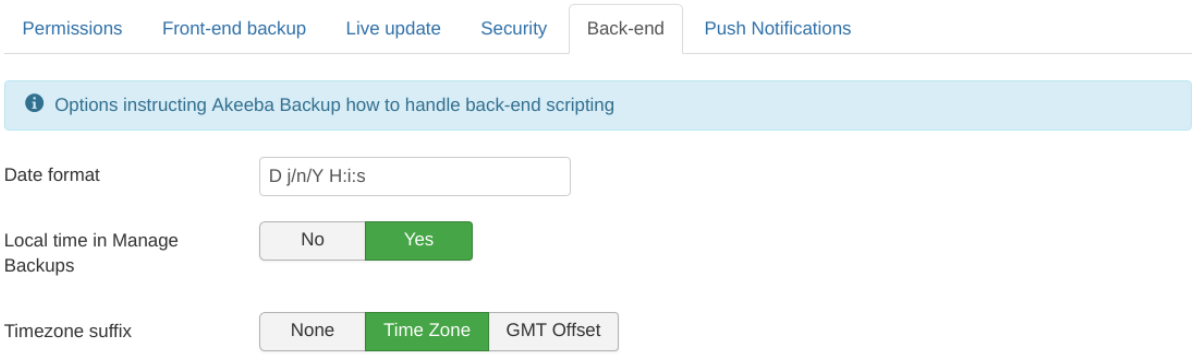
Use Encryption
: Your settings can be automatically stored encrypted using the industry standard AES-128 encryption scheme. This will protect your passwords and settings from prying eyes. If, however, you do not want to use this feature, please set this option to No and reload the Control Panel page to apply this setting. Do note that your server must have either the mcrypt or the OpenSSL PHP extension installed for this feature to work. Please keep in mind that even if your site is using HTTPS this doesn't mean that you have the OpenSSL *PHP extension* installed. You usually have to ask your host to enable it for you.

Tip

For security reasons, we recommend always having this option turned on

Please note that you may have to go to the Configuration page and click on the Save & Close button before Akeeba Backup can successfully detect if your server supports encryption or not. Before doing that, Akeeba Backup might always report that your server does not support encryption.

Back-end



These options define how Akeeba Backup will display its administration interface

Date format
: Defines how the Start time of backups will display in the Manage Backups page. Leave blank to use the default date format. The date format follows the conventions of the PHP date() function [http://www.php.net/date].

Local time in Manage Backups
: When this option is set to No the time the backup started is shown in GMT timezone in the Manage Backups page. If you set it to Yes the time will be shown in the logged in user's timezone.

Please note that this feature will not work reliably unless you have set the correct server timezone in Joomla's Global Configuration. Keep in mind that your server's timezone may be different than the timezone you live in or the timezone of the hosting company's offices. For example, it's possible for an Australian to be hosted with a British hosting company whose servers are in Amsterdam. The correct server timezone in this case would be Europe/Amsterdam.

Moreover, you need to have selected your local timezone in your user profile in Joomla!.

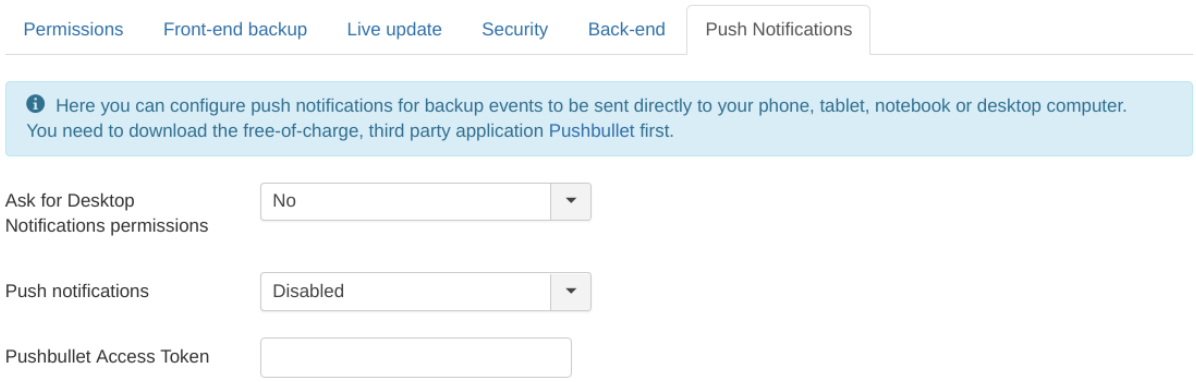
If these prerequisites are not met the time displayed will be off. Lack of configuration on your part is not a bug on our part. Please triple check your timezone settings before filing a "bug" with this feature.

Timezone suffix
: Choose the suffix to append to the backup time in the Manage Backups page. `None` will result in no suffix. We don't recommend it as it's not immediately obvious which timezone is being used. `Time Zone` is the recommended and default option. It will print the human readable timezone setting, e.g. EEST for Eastern Europe Summer Time, PDT for Pacific Daylight Time and so on. `GMT Offset` will instead display the timezone as an offset from GMT, for example GMT+3 for Eastern Europe Summer Time or GMT-7 for Pacific Daylight Time.

Show the “Delete everything before extraction” option
: When this option is enabled, users restoring a backup will see the Delete everything before extraction option. This is a **dangerous** option, meant for advanced users. It will try to delete all files under the backed up site's root before starting the restoration. The obvious danger in this option is that it might delete more than you expected since it cannot and does not know the meaning of each folder under your site's root. It might end up deleting your subdomains, add-one domains, your emails or your cat photos.

Before using this option please make sure that you have kept copies of your backups and any important files outside of your site. If you screw up when restoring your site we take no responsibility. You have been warned.

Push notifications



Akeeba Backup 4.2.2 and later can notify you on backup start, finish and –sometimes– on backup failure using push notifications delivered through the third party application Pushbullet. Push messages are delivered to all your devices running the Pushbullet client software including smartphones and tablets (iOS, Android, Windows) as well as laptops and desktops (Windows, Linux, Mac OS X).

Please note that backup *failure* notifications can only be delivered for backups started through the back-end. For technical reasons beyond our control these notifications can not be delivered for remote (JSON API) and scheduled (CRON job) backups: if the backup fails the PHP executable stops working, therefore our PHP code to send notifications can not work.

Ask for Desktop Notifications permissions
: Enable this option to allow Akeeba Backup to display desktop notifications. Unlike push notifications, these are only shown when you are taking a backup from the backend of your site, though your browser. They are displayed by your browser, not PushBullet. This feature is only compatible with browsers implementing the desktop notifications API for JavaScript such as Firefox, Safari or Google Chrome.

Push notifications
: Select the push notifications type. Currently only Pushbullet and None are supported. If you choose None the push notifications are disabled.

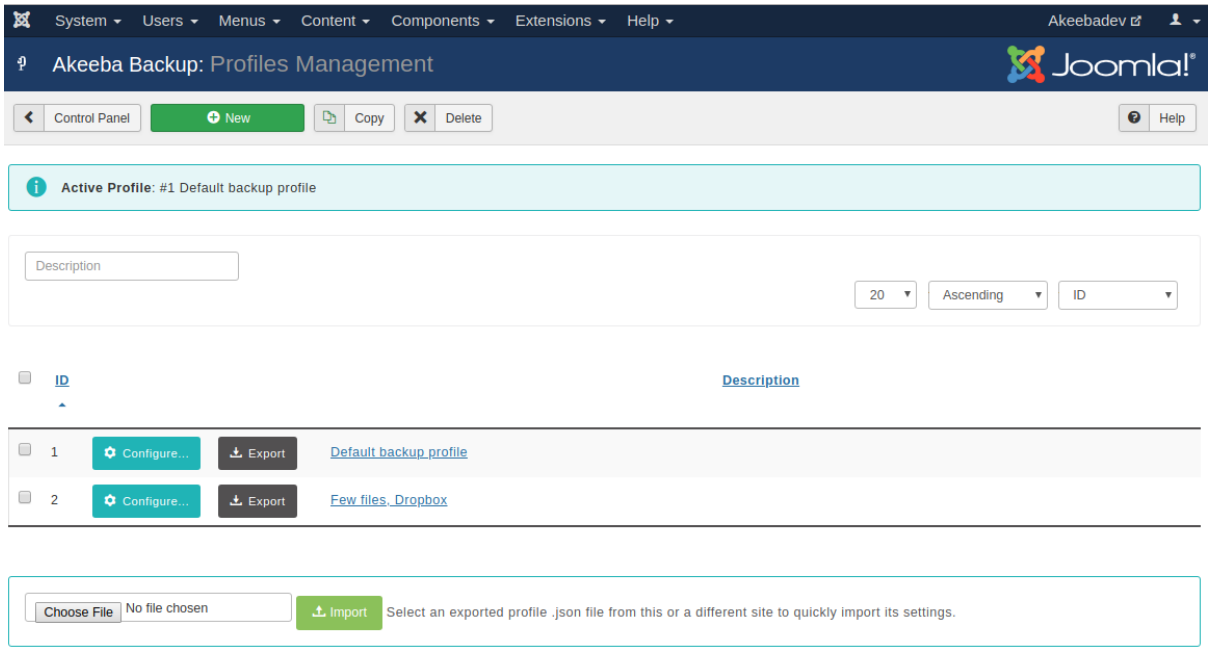
Pushbullet Access Token
: Enter your Pushbullet Access Token. You can find it in your Pushbullet account page [https://www.pushbullet.com/account]. Do note that this token gives full access to your Pushbullet account and is visible by everyone who can view and edit Akeeba Backup's settings.

3. Basic Operations

The Basic Operations group contains the most common functions you will need on your daily Akeeba Backup usage. In fact, you will only use the other pages sparingly, mostly when you create a backup profile or want to update it after doing significant changes to your site.

3.1. Profiles Management

Profiles Management page



The Profiles Management page is the central place from where you can define and manage *backup profiles*. Think of each backup profile as an isolated container holding Akeeba Backup configuration values and filter settings. Each one uniquely and completely defines the way Akeeba Backup will perform its backup process.

The main page consists of a list of all backup profiles. On the left hand column there is a check box allowing the selection of a backup profile so that one of the toolbar operations can be applied. The other column displays the description of the backup profile. Clicking on it leads you to the editor page, where you can change this description.

On the page's toolbar you can find the operations buttons:

New
: Creates a new, empty profile. Clicking on this button will lead you to the editor page, where you can define the name of the new profile, or cancel the operation if you've changed your mind.

Copy
: Creates a prostine copy of the selected backup profile. The copy will have the same name and include all of the configuration options and filter settings of the original.

Delete
: Permanently removes all selected backup profiles. All associated configuration options and filter settings are removed as well. This is an irreversible operation; once a profile is deleted, it's gone forever.

  You can only delete one profile at a time. If you select multiple profiles, only the first one (topmost) will be removed.

When you create a new profile or copy an existing profile, the newly generated profile becomes current. This means that you can work on your new profile as soon as you're finished creating it, without the need to manually make it current from the Control Panel page.

To the left of each profile's name you will find two buttons:

Configure...
: Clicking this button makes that profile current and opens the Configuration page. This is equivalent to going back to the Control Panel, selecting that profile in the list, waiting for the page to reload and clicking on Configuration. We figured that having to click to just one button is much faster – and simpler!

Export

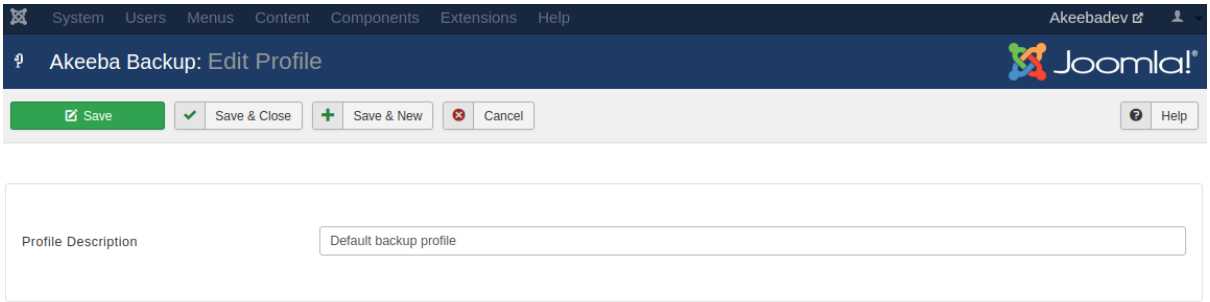
You can export a profile in JSON format. Clicking this button will ask you to download a file with all of the profile settings. You will be able to import that file on the same or a different site using the Import feature further down the page.

Please note that the file you are downloading contains all of the configuration information **UNENCRYPTED**. We strongly advise you to only use this feature when connected to your site over HTTPS. We also strongly advise against storing exported profile files in media which could reasonably be lost (e.g. USB keys) or cloud services without encryption that you manage yourself. Whenever possible, encrypt the exported backup profiles e.g. with GPG or even in a password-protected ZIP file before storing them.

You can also find an Import area below the list of profile. Use the file browser field to select a previously exported profile file from the same *or a different* site. Then click the Import button. Akeeba Backup will import the profile at the end of the profiles list.

We strongly advise you to review your settings after importing a profile. If the profile comes from another site you may have used an absolute path or overridden the database connection information. In this case you will have to change those settings to reflect the current site's configuration.

The Edit Profile page



The editor page which appears when creating or editing a profile is trivial. The only changeable parameter is the profile's description. Clicking on Save & Close applies the settings and gets you to the main Profiles Management page. Clicking on Apply applies the settings and returns you to the editor page. Finally, clicking on Cancel will disregard any changes made and get you to the main Profiles Management page.

Back in the list of profiles, you can click on the Configure button next to a profile to configure it. This changes your currently active backup profile and takes you to the Configuration page.

3.2. Configuration Wizard

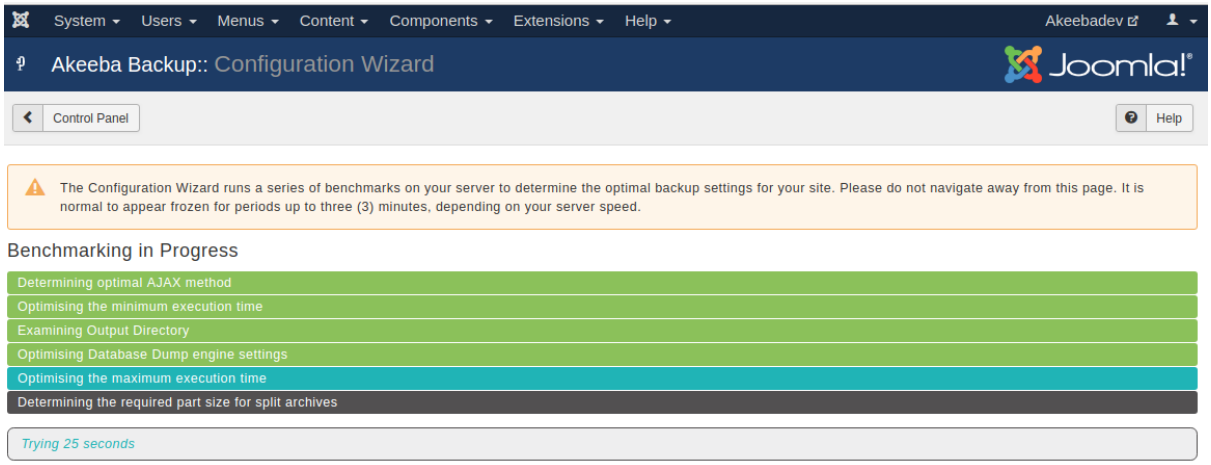
The Configuration Wizard is an automated process which benchmarks your server's performance and tries to fine tune common configuration variables for optimal backup performance on your server. The Configuration Wizard settings are applied to the current profile only. If you want to automatically configure a different profile, you have to select it from the drop-down list in the Control Panel page before clicking on the Configuration Wizard button.

Do note that using the Configuration Wizard has the following effects:

- Your backup type is switched to "Full site backup".
- The archiver engine is switched to "JPA (Recommended)".
- Post processing options are reset to "None" i.e. your backup will not be uploaded to a remote storage location.

If you want to use a different backup type and/or archive type, you can review the configuration changes after the wizard is finished.

The Configuration Wizard page



The Configuration Wizard will automatically fine tune the following configuration parameters:

- AJAX method (use AJAX or IFrames)
- Optimize the minimum execution time so as to make the backup as fast as possible without your server throwing 403 Forbidden errors
- Adjust the location and/or permissions of the output directory. Useful if you just transferred your site to a new server or location.
- Optimize the database dump engine settings to make database dump as fast as possible, while avoiding memory outage errors
- Optimize the maximum execution time so that as few steps as possible are performed during the backup, without causing a timeout
- Automatically determines if your server needs archive splitting.

Important

The Configuration Wizard does not address archive splitting to smaller parts which may be required in some cases when you are using a post-processing engine (such as FTP, Amazon S3, Dropbox, etc). If you will be using post-processing you may have to manually set the Part Size for Split Archives to a different value manually.

At the end of the wizard process, you can either try taking a backup immediately or review and possibly modify the configuration parameters.

3.3. Configuration

Note

Some of the options discussed below may be only available in the Professional edition which is only available to paying subscribers.

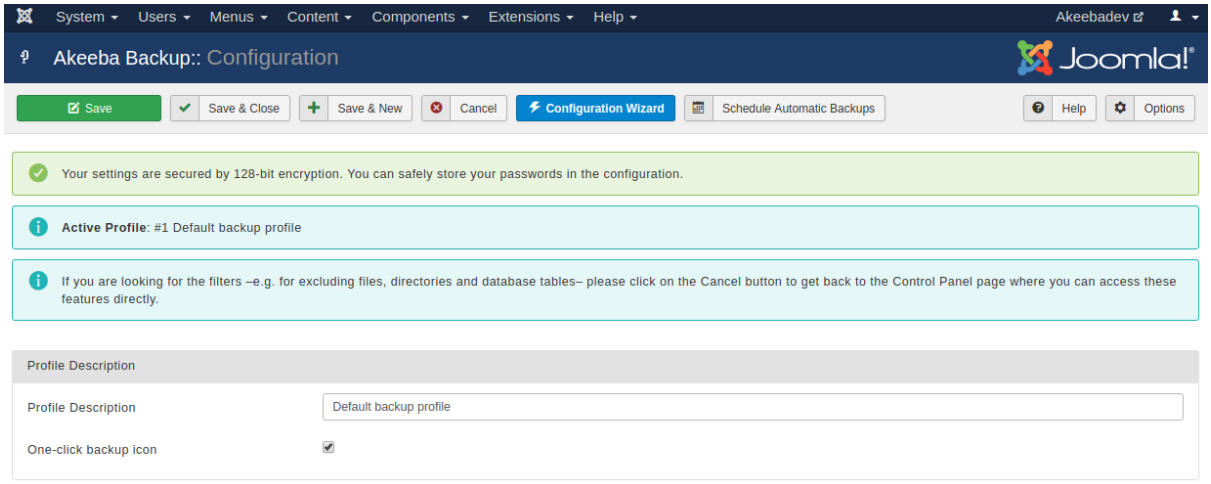
The Configuration page consists of a number of sections, grouping related options together. Each option has a label on the left hand side. You can hover your mouse over the label to see a quick reference (tooltip) for that option. Click on the label to make the quick reference sticky. Click again to let it disappear after you move your mouse away from it.

Some of the settings feature a button at the right had side. These buttons either do some action on the setting's field, like browsing for a folder and testing connection parameters, or it may be labeled Configure.... In the latter case pressing the button will toggle the display of a subsection which contains further configuration options for

your selection. This is typically used to configure the archiver engine, the database dump and any post-processing option.

Another interface element worth mentioning are the composite drop-downs. Whenever you are supposed to enter a number, Akeeba Backup presents you with a drop-down menu of the most common options. You can either select a value from the list, or select "Custom...". In the latter case, a text box appears to the right of the drop-down. You can now type in your desired value, even if it's not on the list. Do note that all of these elements have preset minimum/maximum values. If you attempt to enter a value outside of that range, or an invalid number, they will automatically revert to the closest value which is within the preset range.

The top of the Configuration page



The top of the page is Joomla's toolbar area. The Save button will save the backup profile settings and get you back to the Configuration page. The Save & Close button will save the backup profile settings and take you to the Control Panel page. The Save & New button will save the backup profile settings, create a new profile which is an exact copy of the saved settings and return to the Configuration page of the *new* backup profile. The Cancel button aborts all unsaved changes and takes you back to the Control Panel page.

The Configuration Wizard button will launch the Configuration Wizard. This is useful if you want to initialize your backup profile or reset any settings you are not sure about.

The Schedule Automatic Backups button will take you to the page where you can find out how to automate your backups.

On the top of the main page you can see a reminder of whether you're using encryption for your backup profile settings (something that you can change in the component's Options, accessible from the Options button in the toolbar). We recommend only saving passwords in the configuration when encryption is enabled.

Note

Encryption is not a panacea. The configuration is stored in the database encrypted and the decryption key is stored in a file. This is meant to protect you from a vulnerability which allows the attacker to only access the database. If the attacker can read or, worse, write to your site's files your settings can be reasonably considered compromised: the attacker has all the information they need to retrieve both the encrypted data and the decryption key.

Below you can find the numeric ID and title of the active backup profile. This acts as a reminder, so that you know which profile's settings you are editing.

Further down you will find the Profile Description area. You can view and change the backup profile's description here, without having to go through to the backup profiles page.

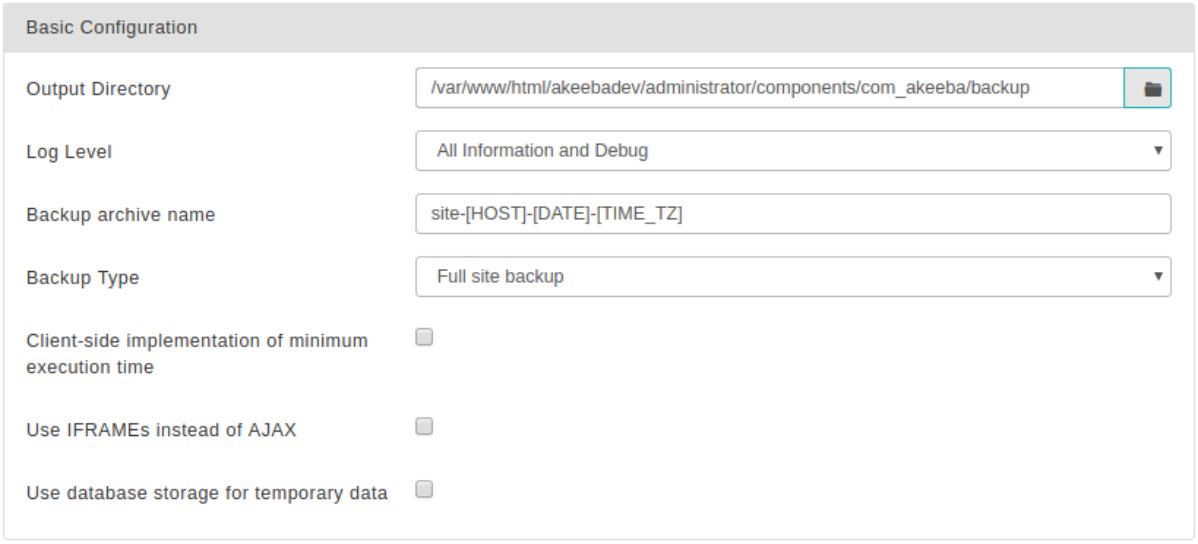
The One-click backup icon box, if checked, will result in a quick icon for this backup profile being displayed in the Control Panel page. Clicking on that icon will start a backup using that profile, without waiting for your confirmation.

Below you'll find a reference for all the options, grouped by the section they belong in.

3.3.1. The main settings

3.3.1.1. Basic Configuration

Basic configuration



Output Directory

This is the directory where the result of the backup process goes. The result of the backup - depending on other configuration options - might be an archive file or a SQL file. This is also where your *backup log file* will be stored. The output directory must be accessible and directly writable by PHP.

Providing a directory with adequate permissions might not be enough! There are other PHP security mechanisms which might prevent using a directory, for example the `open_basedir` restriction which only allows certain paths to be used for writing files from within PHP. Akeeba Backup will try to detect and report such anomalies in the Control Panel page before you attempt a backup.

The output directory, all of its subdirectories and all files contained therein are *automatically excluded from the backup*. Do not use a folder that contains files you want to back up as your backup output directory. Most importantly, do not use your site's root as your output directory! This will lead to a backup that does not have any of your site's files, making it useless! Akeeba Backup will attempt to warn you in this case.

You can use the following variables to make your setting both human readable and portable across different servers - or even different platforms:

- **[DEFAULT_OUTPUT]** is replaced by the absolute path to your site's `administrator/components/com_akeeba/backup` directory. This is assigned as the default location of output files unless you change its location. If you leave it as it is, you are supposed to make sure that the permissions to this directory are adequate for PHP to be able to write to it.
- **[SITEROOT]** is automatically replaced by the absolute path to your site's root
- **[ROOTPARENT]** is automatically replaced by the absolute path to the parent directory of your site's root (that is, one directory above your site's root)

You can always click on the Browse... button to open a directory picker interface. Inside that interface and next to the folder's location there is the button labeled Use. Click on it to make

the current directory the selected one and close the pop-up. To make it even easier for you, Akeeba Backup displays a small icon next to the Use button. If it's a green check mark the directory is writable and you can use it. If it's a red X sign, the directory is not readable and you either have to select a different directory, or change this directory's permissions.

Log Level

This option determines the verbosity of Akeeba Backup's log file:

- **Errors only**. Only fatal errors are reported. Use this on production boxes where you have already confirmed there are no unreadable files or directories. We do not recommend using this setting.
- **Errors and warnings**. The minimum recommended setting, reports fatal errors as well as warnings. Akeeba Backup communicates unreadable files and directories which it wasn't able to backup through warnings. Read the warnings to make sure you don't end up with incomplete backups! Warnings are also reported in the Backup Now page GUI irrespective of the log verbosity setting as a convenience.
- **All information**. As "Error and Warnings" but also includes some informative messages on Akeeba Backup's backup process.
- **All Information and Debug**. This is the recommended setting for reporting bugs. It is the most verbose level, containing developer-friendly information on Akeeba Backup's operation. Please take a backup using this log level before requesting support from us.
- **None**. This log level *is not recommended*. It disables logging altogether.

We recommend using Errors and Warnings after you have confirmed that your backup is running properly and All Information And Debug when you need to request support.

Backup archive name

Here you can define the name of your backup files. You must not enter an extension, it's added automatically.

There are a few available variables. Variables are special pieces of text which will be expanded to something else at backup time. They can be used to make the names of the files harder to guess for potential attackers, as well as allow you to store multiple backup archives on the output directory at any given time. The available variables and their expansion at backup time are:

[HOST] The configured host name of your site.

Note

Whenever you visit Akeeba Backup's Control Panel we store the host name in the database and try to use it when you take a backup from the command line. If this value cannot be stored or if your site's host name changed since the last time you may get an incorrect host name when taking a backup from the command line, i.e. when you are using the akeeba-backup.php script, typically from a CRON job.

[DATE] The current server date, in the format YYYYMMDD (year as four digits, month as two digits, day as two digits), for example 20080818 for August 18th 2008.

[YEAR] The year of the current server date, as four digits

[MONTH] The month of the current server date, as two digits (zero-padded)

[DAY] The day of the current server date, as two digits (zero-padded)

[WEEK]	The current week number of the year. Week #1 is the first week with a Sunday in it.
[WEEKDAY]	Day of the week, i.e. Sunday, Monday, etc. The full name is returned in your current Joomla! language. Front-end, remote and CRON backups may return this in English or your default Joomla! language. This is not a bug, it is how Joomla!'s translation system is supposed to work.
[RANDOM]	A 64-character random string. Use with caution, it can cause backup failure due to the file name being too long for your server.
[TIME]	The current server time, in the format HHMMSS (hour as two digits, minutes as two digits and seconds as two digits), for example 221520 for 10:15:20 pm.
[TIME_TZ]	The current server time, in the format HHMMSSGMTOOOO (hour as two digits, minutes as two digits and seconds as two digits followed by GMT and the the offset to the GMT timezone as four digits), for example 221520GMT+0300 for 10:15:20 pm in Nicosia, Cyprus (which is 3 hours ahead of GMT). We strongly advise using this instead of [TIME] to remove any ambiguity on which timezone is being used. This is especially important if you rely on the filenames to understand which is the backup you are looking for or when you have multiple people taking and restoring backups in different timezones.
[TZ]	The timezone all dates and times are expressed in. This variable gives you the timezone in a manner that is safe for use in filenames, even on Windows. For example, asia_nicosia for Nicosia, Cyprus.
[GMT_OFFSET]	The timezone all dates and times are expressed in. This variable gives you the timezone as an offset to the GMT timezone. For example +0300 for Cyprus (3 hours ahead of GMT), +0530 for India (5 hours 30 minutes ahead of GMT) or -0600 for Chicago (6 hours behind GMT).
[TZ_RAW]	The timezone all dates and times are expressed in. This variable gives you the raw timezone, e.g. Asia/Nicosia for Nicosia, Cyprus. Kindly note that this results in invalid filenames on Windows.
[VERSION]	The version of Akeeba Backup. Useful if you want to know which version of Akeeba Backup generated this archive file.
[PLATFORM_NAME]	The name of the platform Akeeba Backup is currently running under. This always returns "Joomla!".
[PLATFORM_VERSION]	The version of the platform Akeeba Backup is currently running under. This always returns the current Joomla! version, e.g. 1.2.3.
[SITENAME]	The name of the site, lowercased and transformed into a format which guarantees compatibility with all filesystem types commonly found in modern Operating Systems. Please note that the site name will be trimmed at 50 characters if it's longer.

The date and time options are expressed in the timezone selected in the component's Options page under Backup Timezone. By default this is GMT. You are advised to change this to the timezone your site administrators are most familiar with.

Backup Type — It defines the kind of backup you'd like to take. The backup types for Akeeba Backup are:

- **Full site backup** which backs up the Joomla! database, any extra databases you might have defined and all of the site's files. This produces a backup archive with an embedded installer so that you can restore your site with ease. This is the option 90% of the users want; it is the only option which creates a full backup of your site, capable of producing a working site if everything is wiped out of your server.

- **Main site database only (SQL file)** which backs up only the Joomla! database. It results in a single SQL file which can be used with any database administration utility (e.g. phpMyAdmin to restore only your database should disaster strike. This option is recommended for advanced users only.

- **Site files only** which backs up nothing but the site's files. It is complementary to the previous option.

  ## Warning

  Having one "main site database" backup and one "sites files only" backup is not equal to having a full site backup! The full site backup stores the database dump in a more detailed format and also includes an installation script which, just like Joomla!'s web installer, allows you to effortlessly recover your site even if everything is wiped out of your server. It acts as the glue between the two pieces (files and database).

- **All configured databases (archive file)** which creates an archive file containing the database dumps of your site's database and an installer script to restore them. It's like a full backup without the files.

- **Incremental (files only)**. This is the same as the Site files only option, but instead of backing up all of your site's files, it only backs up the files which changed since the last time you performed a backup. The only comparison made is between the file's modification time and the last successful backup's time. The "last successful backup" refers to the last backup made using this backup Profile and which has a status of "OK", "Remote" or "Obsolete".

  Restoring an incremental backup set is a *manual process*. You have to manually extract the files from your "base" backup (an archive made with a Full Site Backup profile), then extract all incremental archives on top of it. Finally, used this collection of extracted files to restore your site. This process should only be used if you really know what you are doing. Do not trust that Akeeba Backup can sort out the collection of incremental backups and help you restore them. It won't.

- **Full site, incremental files**. This backup is a combination of Full site backup and Incremental. It works like a full site backups except for the site files. The site files are treated the same as an Incremental backup, i.e. only modified files are included. This backup type is intended for sites with frequently changing database contents and infrequently changing files. The same warnings about restoration as an Incremental backup apply.

Client-side implementation of minimum execution time

Akeeba Backup splits the backup process into smaller chunks, called backup steps, to prevent backup failure due to server time-out or server protection reasons. Each backup step has a minimum and maximum duration defined by the Minimum Execution Time, Maximum Execution Time and Execution Time Bias parameters in this Configuration page. If the step takes less time to complete than the minimum duration Akeeba Backup will have to wait.

When this box is unchecked (default) Akeeba Backup will have the server wait until the minimum execution time is reached. This may cause some very restrictive servers to kill your backup. Checking this box will implement the waiting period on the browser, working around this limitation.

Important

This option only applies to back-end backups. Front-end, JSON API (remote) and Command-Line (CLI) backups always implement the wait at the server side.

Use IFRAMEs instead of AJAX

Normally, Akeeba Backup uses AJAX to perform each backup step. In some very old browsers this didn't work very well, hence this option. When selected, Akeeba Backup will use a hidden IFRAME on the page to step through the backup process. In modern browsers this might actually prevent the backup from working due to the browsers' ad blocking or security features. You should have no reason to use this option. Please do not use it unless you are told to do so by our support.

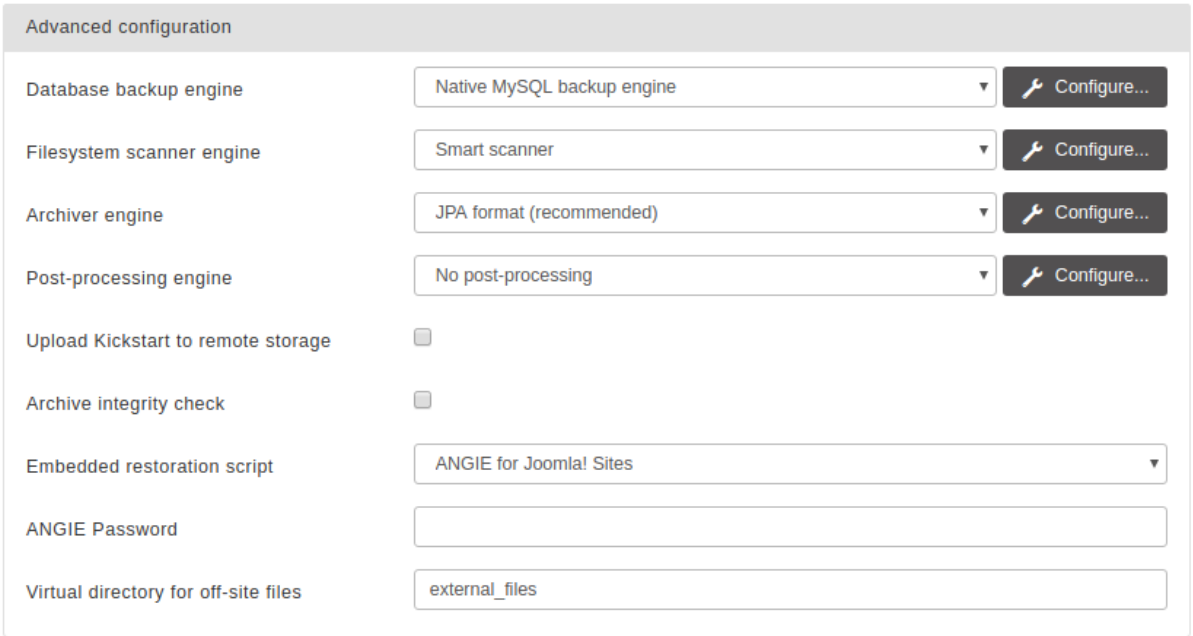
Use database storage for temporary data

Normally, Akeeba Backup stores temporary information required to process the backup in multiple steps in .php files stored in your Output Directory. In very rare circumstances the host mistakes this file for malicious code, deletes it automatically and the backup fails or immediately restarts without any apparent reason. If this happens, enable this option. It tells Akeeba Backup to use your site's database to store this temporary information instead of .php files.

Do note that on some hosts this will cause the backup to fail with a "MySQL server has gone away" error message. That is a problem with the host's configuration. In those cases, nothing can be done. Our suggestion: if you receive such an error, migrate your site to a new host as the one you are using is most likely overcrowded and restricted. Moving to a faster, more reliable host can benefit your site in many more ways than just being able to run a backup.

3.3.1.2. Advanced configuration

Advanced configuration



Database backup engine

This option controls how Akeeba Backup will access your database and produce a dump of its contents to an SQL file. It is used with all backup types, except the files only type. The available options for this setting are discussed in the Database dump engines section of this document.

Filesystem scanner engine

This option controls how Akeeba Backup will scan your site for files and directories to back up. The available options for this setting are discussed in the File and directories scanner engines section of this document.

Archiver engine

This option controls which kind of archive will be produced by Akeeba Backup. The available options for this setting are discussed in the Archiver engines section of this document.

Post-processing engine

Akeeba Backup allows you to post-process the backup archives once the backup process is over. Post-processing generally means sending them somewhere off-server. This can be used, for example, to move your backup archives to cloud storage, increasing your data safety. The available options for this setting are discussed in the Data processing engines section of this document.

Upload Kickstart to remote storage

By selecting this option you instruct Akeeba Backup to also upload kickstart.php on the remote storage alongside your backup archive. When used with the Upload to Remote FTP Server and Upload to Remote SFTP Server you can perform easy site transfers without leaving your browser. Enter the new site's (S)FTP information in the Data Processing Engine configuration and select the Upload Kickstart to Remote Storage option, then take a new backup. When the backup is complete just open the new site's kickstart.php URL (e.g. http://www.example.com/kickstart.php) in your browser to begin the restoration on the new site's server. This even works with mobile devices, allowing you to transfer sites without using a laptop or desktop computer and without using up a lot of bandwidth on your device.

Archive integrity check

When enabled Akeeba Backup will go through the archive extraction process without writing anything to the disk. This makes sure that the archive is not corrupt. If the archive is found to be corrupt an error is raised and the backup process stops.

This feature will NOT work when the Process each part immediately option is enabled in the Post-processing Engine configuration. When you are processing each part immediately the backup archive parts are transferred away from your server before the end of the backup is reached. As a result it is not possible to do a test extraction (the archive file parts are no longer there, so there's nothing to try and extract). It WILL however work when you are simply using a post-processing engine (e.g. Upload to Amazon S3) without the process each part immediately option. Please bear in mind that the integrity check runs BEFORE post-processing (uploading) the backup archive parts to remote storage because there's no reason to put a broken archive for safe-keeping in remote storage.

This feature will only work if you are using an Archiver Engine which creates backup archives. This is typically the case with most Archiver Engines. Notable exceptions are, of course, the DirectFTP and DirectSFTP engines which do not produce backup archives. If you enable this feature on a backup profile using either of these Archiver Engines you'll get a warning.

Enabling this feature will increase the time required to complete the backup process and use substantially more memory and CPU resources. Akeeba Backup goes through the same archive extraction process as Kickstart with the only difference that it does not write anything to the disk.

Finally do keep in mind that this feature only makes sure the archive can be extracted, it does *not* test whether the database data can be restored or if the restored site works correctly. It's still up to you to do a periodic, complete test restoration of your site.

Embedded restoration script

Akeeba Backup will include a restoration script inside the backup archive in order to make restoration easy and the backup archive self-contained. You do not need anything else except the archive in order to restore a site. Restoration scripts honour the settings in your configuration.php, modifying only those necessary (for example, the database connection information), allowing you to create pristine copies ("clones") of your site to any host. You can find more information about restoration scripts in the next Chapter.

ANGIE Password

If you are using the ANGIE embedded installer script you can optionally password-protect it, preventing unauthorised access to the installer. When you run the installer you will be asked to enter this password. Please note that the password is case sensitive, i.e. ABC, abc and Abc are three different passwords.

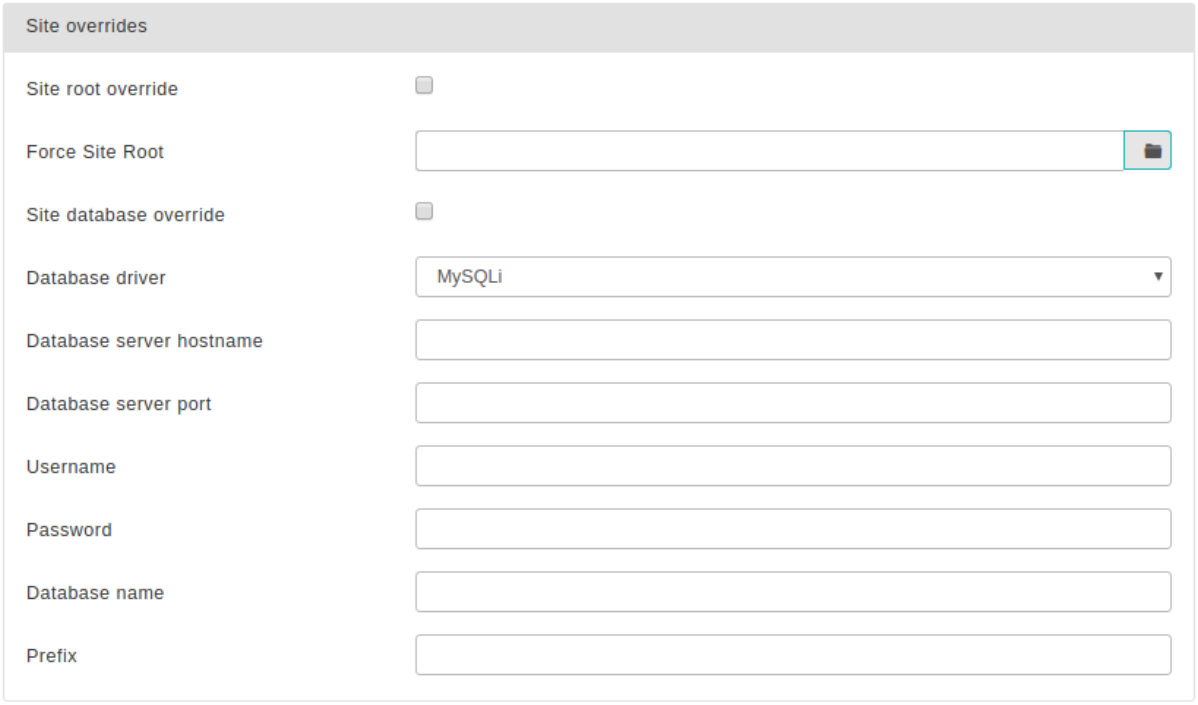
Virtual directory for off-site files
: Using the off-site directories inclusion of Akeeba Backup Professional, the component will be instructed to look for files in arbitrary locations, even if they are outside the site's root (hence the name of that feature). All the directories included with this feature will be placed in the archive as subdirectories of another folder, in order to avoid directory name clashes. We call this folder the "virtual directory", because it doesn't physically exist on the server, it only exists inside the backup archive.

3.3.1.3. Site overrides

These settings are all optional and only available in Akeeba Backup Professional. They allow you to back up a different site than the one Akeeba Backup is currently installed. Essentially, you can install Akeeba Backup on one site and have it back up all sites on the server.

> **Note**
>
> You do not need to set anything up in this section if you only intend to backup or transfer your site. This is only required when you want Akeeba Backup to backup a different site than the one it is installed in.

Site overrides



Site root override
: When not checked (default), Akeeba Backup will back up the files and folders under the root of the site it is installed. When this option is checked, it will use the site root in the Force Site Root option below. Use this when you want to backup a different site than the one Akeeba Backup is installed in.

Force Site Root
: The root of the site to back up. This is only necessary if you have checked the Site root override option above.

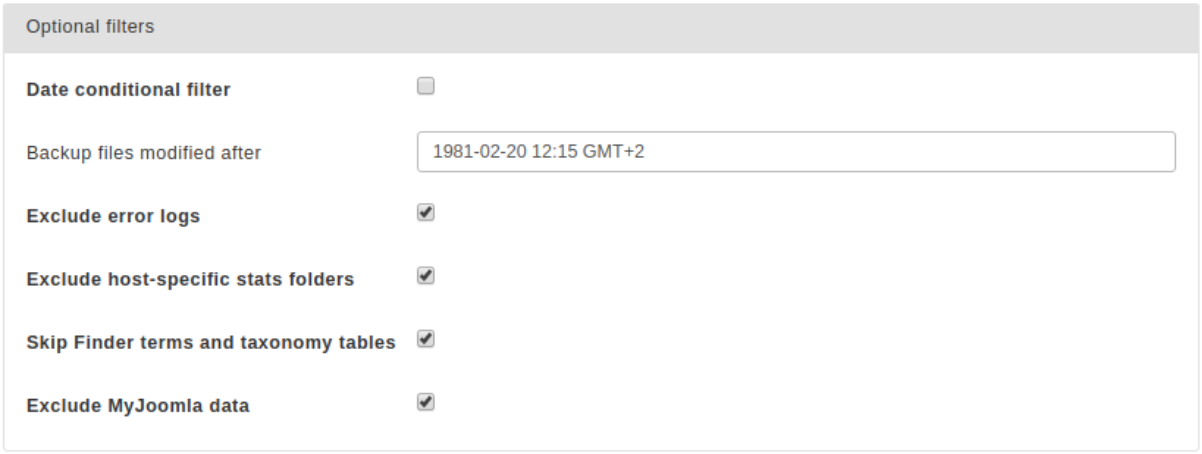
Site database override
: When not checked (default), Akeeba Backup will back up the database tables inside the database to which the site Akeeba Backup is installed in connects to. In other words, when this option is not checked, Akeeba Backup will back up the current site's database.

  On the other hand, if this option is checked, Akeeba Backup will backup the database whose connection information you specify in the settings below. Use this when you want to backup a different site than the one Akeeba Backup is installed in.

Database driver	Choose between the database driver. For MySQL databases you can choose between the MySQL and MySQLi driver. If you do not know the difference between the two, MySQLi (with the trailing "i" which stands for "improved") is the best choice.
Database hostname	The hostname or IP address of the database server. Usually that's `localhost` or `127.0.0.1`. If unsure, ask your host.
Database server port	If your database server uses a non-standard port, enter it here. If you have no idea what this means, you most likely need to leave that field blank.
Username	The username to connect to your site's database.
Password	The password to connect to your site's database.
Database name	The nae of your database.
Prefix	The prefix of the tables of the site you're backing up. That's the common part of their names up to and including the first underscore.

3.3.1.4. Optional filters

Optional filters



These optional filters allow you to exclude files or database tables without having to manually select them in the respective exclusion filter pages.

Date conditional filter

Date conditional filter	Tick the checkbox to activate this filter. It allows you to backup only files modified after a specific date and time. This is different than the incremental file only backup. It allows you to backup files newer than the specified date no matter which backup mode (full site backup, files only backup, incremental files only backup) you are using.
Backup files modified after	Files before this date and time will be skipped from the backup set. The format for the date and time parameter is YYYY-MM-DD HH:MM:SS TIMEZONE. This means that you have to specify the year as four digits, followed by a dash, then the month as two digits (e.g. 09 for September), followed by a dash, then the day as two digits (e.g. 01 for the 1st day of the month). For example, September 1st, 2010 is written as 2010-09-01. If you want to specify the time, leave a space after the date and write down the time as the hour using two digits (00-23, no a.m./p.m. is supported!), then a semicolon, then the minutes as two digits, followed

by a semicolon, then the seconds as two digits. For example 59 seconds after 11:05 p.m. is written as 23:05:59. You can optionally leave a space after the time and specify the timezone as GMT+/-time. For example, GMT-6 is Dallas time which is six hours behind the GMT and GMT+2 is two hours ahead of GMT which is the Eastern Europe Time. If you do not specify a timezone the GMT timezone is assumed.

Important

You have to set your server's timezone in Joomla!'s Global Configuration for this feature to work reliably. If you get strange results, try editing your site's Global Configuration before asking us for support.

Exclude error logs	Automatically exclude error log files, e.g. `error_log`, no matter where they are on the site being backed up. These files change their size while the backup is in progress which may lead to corrupt backups.
Exclude host-specific stats folders	When enabled, Akeeba Backup will automatically exclude the most common host-specific folders for storing access statistics for your site. These folders are read-only by your web site user, causing restoration issues if they are backed up
Joomla! User Actions Log	Skips over the contents of the Joomla! User Actions Log. The database table holding these records can get quite big on a busy site, slowing your backup down and bloating its size. Skipping over this data does not have an adverse impact to the functionality of the site.
Skip Finder terms and taxonomy tables	Since Joomla! 2.5, the Joomla! CMS ships with a feature called "Smart Search", also known as "Finder". This is a mini search engine built into the CMS. It works by scanning your content and keeping a complex database structure linking potential search terms (words) with content items in compatible components. Due to its nature it stores an immense amount of information in the database. This information takes a very long time to back up. Moreover, this information doesn't need to be backed up as it can be regenerated by using the "Reindex" button in Smart Search's back-end interface. In the interest of speeding up your backups and not including redundant information in the backup Akeeba Backup by default has this option enabled. This instructs the database backup portion of our backup engine to skip backing up the contents of Finder's (Smart Search's) tables. If for some reason you want to back up this content please uncheck this box.
Exclude MyJoomla data	MyJoomla.com stores several info about your files in your database, inflating the size of the backup archive. You can exclude this data since they will be restored during the next audit of your site

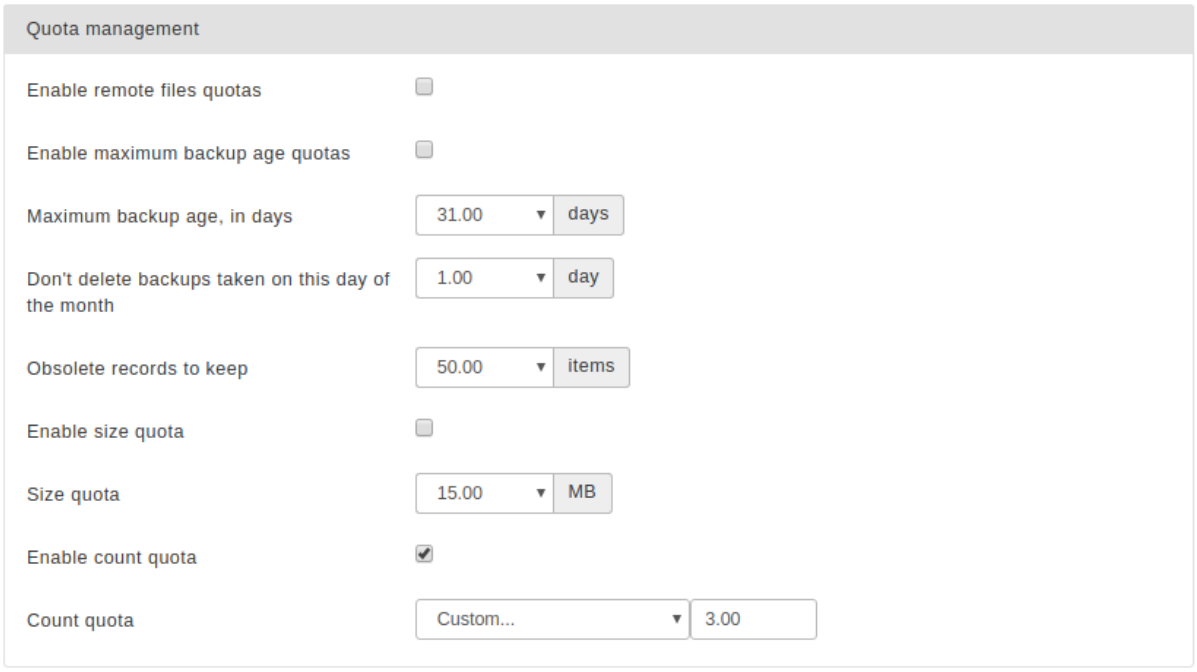
3.3.1.5. Quota management

Quotas let you automatically remove backup archives and / or backup records based on specific criteria. Quotas are always calculated against the **backup records**, not the backup archives on disk on or on remote storage. In other words, if you do not see a backup record in the Manage Backups page it is NOT taken into account when applying quotas.

Furthermore, quotas will take into account only the backup record, without checking if the file exists. If a backup is listed as OK or Remote in the Manage Backups page it participates in the quotas.

Finally, the quotas apply *per backup profile*. They will only take into account backup records in the same backup profile.

Quota management



Enable remote files quotas

When checked, the quota settings will also be applied to remotely stored files. This option only works with the cloud storage engines which support remote file deletion.

Please keep in mind that remote file quotas, just like local file quotas, *only apply to backup records in the database*. Akeeba Backup cannot and will not consider files present in the remote storage which do not have a corresponding backup record in the same backup profile. Furthermore, if you manually delete the files from the remote storage but leave behind the backup record in Akeeba Backup, these backups will still participate in quotas, even though their files no longer exist.

Enable maximum backup age quotas

When checked, Akeeba Backup will only apply quotas based on the date and time the backup was started. This allows you to easily do something like "keep daily backups for the last 15 days and always keep the backup taken on the first of each month".

Warning

Enabling this options makes Akeeba Backup **completely ignore** the size and count quotas.

Maximum back age, in days

Only applies when the Enable maximum backup age quotas option is enabled.

Backups older than this number of days will be deleted. Newer backups will not be deleted.

Don't delete backups taken on this day of the month

Only applies when the Enable maximum backup age quotas option is enabled.

Even when a backup is older than the Maximum back age, in days setting, it won't be deleted if it was taken on this day of the month. For example, if you set this to 1, backups taken on the first day of each calendar month will not be deleted. Setting this option to 1, the backup age to 31 and enabling the maximum backup age quotas you end up keeping all backups taken the last month and keeping the backups taken on the first of each month.

Obsolete records to keep

When the locally stored files of a backup record are deleted (either manually or automatically after uploading it to a remote storage) the record is marked as Obsolete or Remote. Some users prefer to limit the number of the backup entries showing in the Manage Backups (formerly

"Administer Backup Files") page. This option instructs Akeeba Backup to keep at most that many obsolete/remote records and automatically delete older obsolete/remote entries. This is different than the rest of the quotas because it doesn't remove files from your server, it removes the backup entry from Akeeba Backup's interface.

Warning

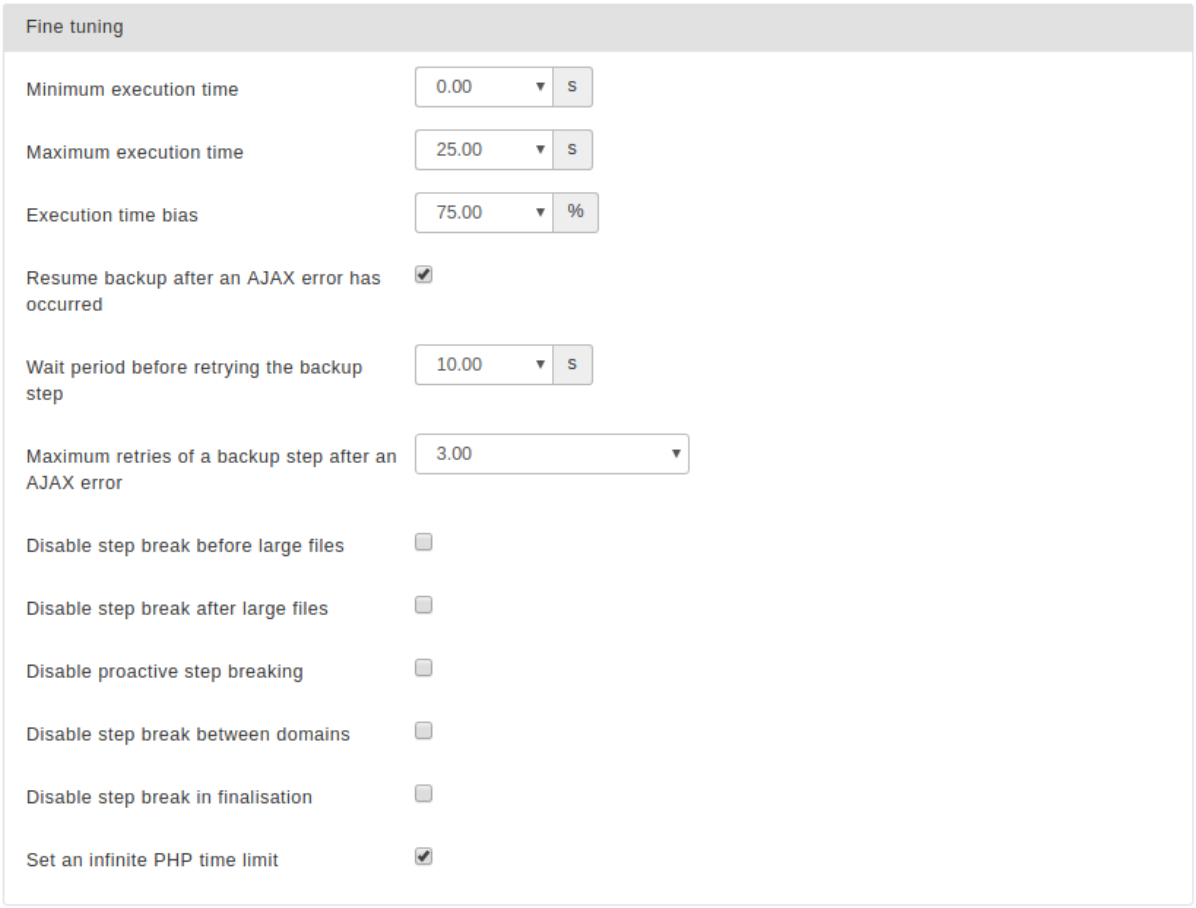
Backups marked as "Remote" are also considered obsolete records: the backup archive does not exist on your server, it only exists on the remote storage. Therefore this setting will also remove the backup records for the Remote backups. Since you are removing the backup records they WILL NOT participate in remote file quotas! Therefore the Obsolete records to keep setting MUST be higher than the total number of backups you will keep before the quotas kick in plus one.

For example, if you are taking 4 backups a day and you have enabled a maximum backup age quota of 30 days you need to set the Obsolete records to keep to at least 121 (4 backups / day x 30 days + 1 = 120 + 1 = 121). Otherwise the maximum backup age quotas will NOT work as expected.

Enable size quota	When checked, old backup archives will be erased when the total size of archives stored under this (and only this) profile exceed the Size quota setting.
Size quota	Defines the maximum aggregated size of backup archives *under the current profile* to keep. Only has an effect if the previous options is activated.
Enable count quota	When checked, old backup archives will be erased when there are more backups stored under this (and only this) profile exceed the Count quota setting.
Count quota	Defines the maximum number of backups *under the current profile* to keep. Only has an effect if the previous options is activated.

3.3.1.6. Fine tuning

Fine tuning



Minimum execution time

Some servers deploy anti-hacker measures (such as mod_evasive or mod_security) which will deny connections to the server if the same URL is accessed multiple times in a limited amount of time. Akeeba Backup has to call its backup URL multiple times, so it runs the risk of being treated as a potential hacker and denied connection to your server, resulting to backup failure.

In order to work around this issue, Akeeba Backup can throttle the rate of server requests using this setting. A minimum execution time of 2 seconds means that calls to the backup URL will happen *at most* once every two seconds. You are suggested to keep the default value.

Maximum execution time

Akeeba Backup has to divide the backup process in individual small steps in order to avoid server timeouts. However, it has to know how small they have to be; that's why this setting exists. Akeeba Backup will try to avoid consuming more time per step than this setting. You have to use a number lower than the `maximum_execution_time` setting in your host's php.ini file. In fact, we suggest using 50% of that value here: if your host allows up to 30 seconds in the php.ini, you have to enter no more than 15-17 seconds here. If unsure, 7 seconds is a very safe value under most configurations.

Execution time bias

When Akeeba Backup calculates the available time left for performing operations within the current backup step a number of external settings may skew this result and lead to timeout errors. This setting defines how conservative the backup engine will be when performing those calculations and is expressed as a percentage of the Maximum execution time parameter. The less this setting is, the more conservative Akeeba Backup gets. It is suggested not to use a value over 75%, unless you have a very fast server. If you experience timeouts, you may want to lower this setting to a value around 50%.

Resume backup after an AJAX error has occurred

When this option is unchecked Akeeba Backup will completely stop the backup when the server responds with an error or the communication with the server is cut short. When this option is enabled (default), Akeeba Backup will try to resume the backup by repeating the last backup step. This will not let you successfully resume all backups which result in an error: only backup attempts temporarily blocked by server CPU usage restrictions or network outage issues can be resumed. If the backup fails due to a timeout error, memory outage, incompatible server software etc the backup resumption will result in the same error until it leads to a permanent backup failure.

Important

This feature only applies to back-end backups. This feature will not be taken into account when you have enabled the Process each part immediately option in the configuration of the Data processing engine since it's impossible to retry backing up to a backup archive which may have already been transferred to remote storage and removed from the server.

Wait period before retrying the backup step

How many seconds to wait before resuming the backup. It is advisable to set this to 30 seconds or more (120 seconds is recommended in most cases) to give your server / network the necessary time to recover from the error condition which caused your backup to fail.

Maximum retries of a backup step after an AJAX error

How many consecutive times should we retry resuming the backup before finally giving up and throwing a permanent error (backup failure). 3 to 5 retries work best on most servers.

Disable step break before large files

When the application detects a large file (see the filesystem scanner engine configuration) it will try to break the execution of the current backup step and start backing up the large file in its own backup step. This is a conservative behaviour that increases the likelihood of being able to backup large files but makes the backup slower. If you check this box the backup will become faster, but it might fail backing up larger files.

Disable step break after large files

When the application finishes backing up a large file (see the filesystem scanner engine configuration) it will try to break the execution of the current backup step and continue the backup process in a step. This is a conservative behaviour that decreases the likelihood of the backup engine timing out after backing up a large file but makes the backup slower. If you check this box the backup will become faster, but it might fail after backing up larger files.

Disable proactive step breaking

The application tries to guess how much time it will take it to backup each file. If it believes that backing up the next file in its queue will take too long it will break the backup step and continue the backup in a new step. This decreases the likelihood of server timeouts, at the expense of making the backup a little slower, especially if you have lots of tiny files. If you check this box the backup will become faster, but it might fail in some cases.

Disable step break between domains

Normally, Akeeba Backup forces the current backup step to finish when it's about to move to a different backup domain, e.g. after finishing backing up the database and getting ready to backup the files of your site. This gives the backup engine the chance to do garbage collection and free up resources.

You can enable that option to make the backup 1-2 seconds faster, risking a backup failure in resource-restricted servers. We consider it a generally unsafe option and we advise against using it.

Disable step break in finalization

Normally, Akeeba Backup forces the current backup step to finish when it's about to move to a different finalization operation e.g. after finishing considering quotas and it's about to start post-processing a backup archive. This gives the backup engine the chance to do garbage collection and free up resources.

	You can enable that option to make the backup 1-2 seconds faster, risking a backup failure in resource-restricted servers. We consider it a generally unsafe option and we advise against using it.
Set an infinite PHP time limit	If your server is using the CGI or FastCGI interface to PHP, checking this option will make it less likely that the backup dies due to a PHP timeout issue. We consider it generally safe checking this box as we have never observed or got reports of any side-effects.

3.3.2. Database dump engines

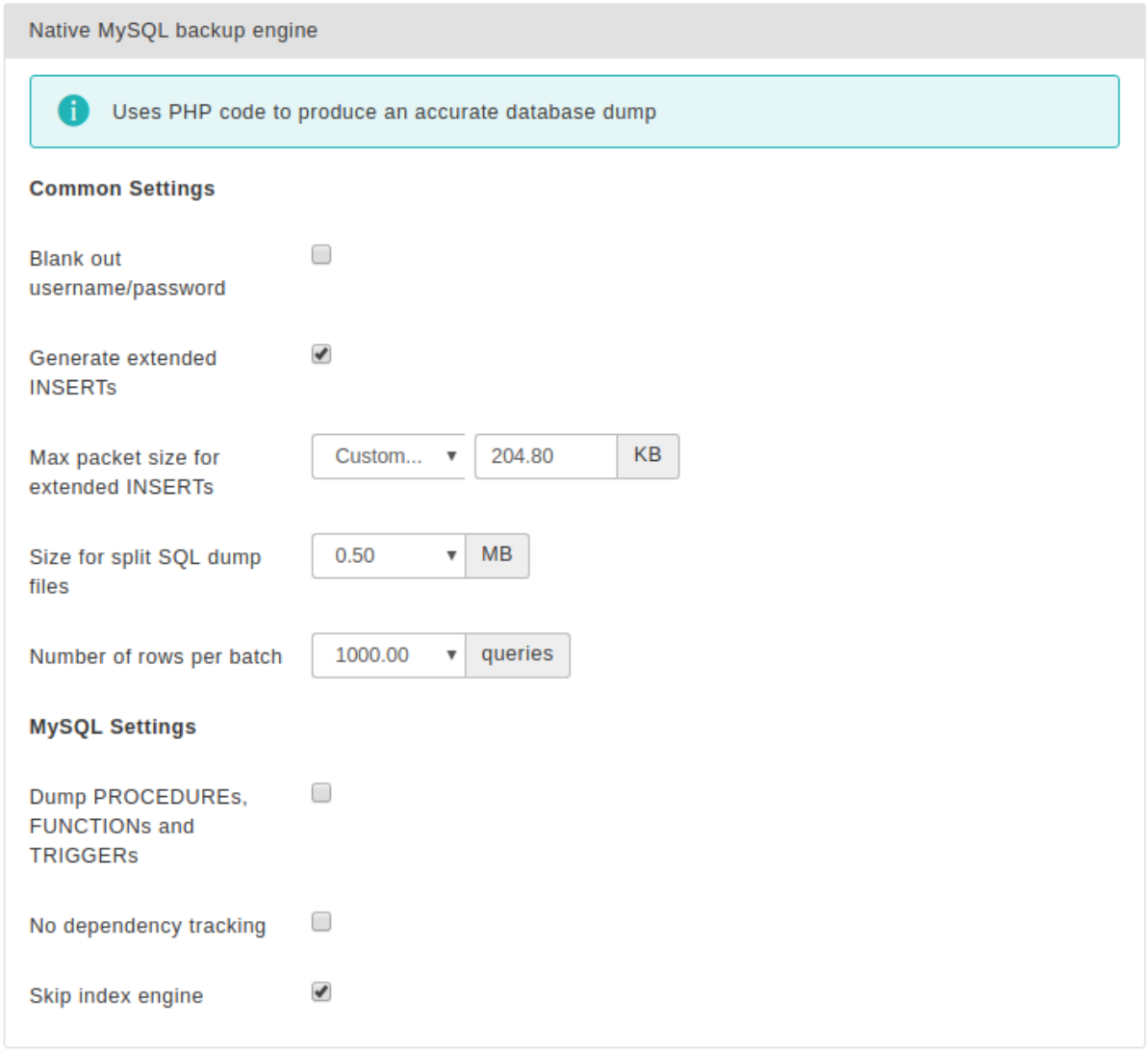
3.3.2.1. Native MySQL Backup Engine

This engine will take a backup of your MySQL database using its native features for reporting the structure of tables, views, triggers etc.

Important

Restoring views, triggers, stored procedures and functions may require elevated privileges for the database user during the restoration process. Most hosts do not assign this kind of privileges. If your restoration fails with a MySQL error when restoring such database entities you may have to ask your host to assign those privileges to your database user.

Native MySQL Backup Engine



Blank out username/password

When enabled, Akeeba Backup will not include the username and password of database connections in the backup. Please note that this option only removes the database username and password from the installation/sql/databases.json (or databases.ini, depending on your Akeeba Backup version) file which is included in the backup. It does not remove the database connection information from the configuration.php file of Joomla!. If you want to remove the database connection information for security reasons you should exclude configuration.php from your backup using the Files and Directories Exclusion filter feature of Akeeba Backup.

Generate extended INSERTs

When this is not checked, Akeeba Backup will create one INSERT statement for each data row of each table. When you have lots of rows with insignificant amount of data, such as banner and click tracking logs, the overhead of the INSERT statement is much higher than the actual data, causing a massively bloated database dump file. When this option is enabled, the dump engine will create a single INSERT statement for multiple rows of data, reducing the overhead and resulting into significantly smaller backup archives. Moreover, this will lead to much less SQL commands being run during restoration, which is of importance on many restrictive shared hosting environments. It is suggested to turn this setting on.

Max packet size for extended INSERTs

If the previous setting is enabled, this setting defines the maximum length of a single INSERT statement. Most MySQL servers have a configured limit of maximum statement length and will not accept an INSERT statement over 1Mb. It is suggested to leave the default

conservative setting (128Kb) unless you know what you're doing. If you get restoration failures indicating that you exceeded the maximum query length, please lower this setting.

Size for split SQL dump files

Akeeba Backup is able to split your MySQL database dump to smaller files. This allows for an improved compression ratio and also helps avoid several problems with certain cheap hosts which put a restriction on the maximum size a file generated by PHP code can have.

Ideally, you should specify a setting which is about half as much as your Big file threshold setting in the archiver engine's configuration options pane. The reason to do that is that the archiver engines will not compress files with sizes over the value this threshold. Since it's impossible to have absolute control of the size of the database dump, using half the value of this setting allows for the expected size fluctuation.

If you want to disable this feature and create a single big SQL dump file instead, just set this option to 0 Mb.

Important

This setting has no effect on "Main site database only" backup profiles. This is because the nature of this backup type does not allow splitting the database archive dump. If you want something equivalent, please use the "All configured databases" backup type instead, as it creates an archive file which contains your (split) database dump and takes up MUCH less space on your web server.

Number of rows per batch

Dumping table data happens in "batches", i.e. a few rows at a time. This parameter defines how many rows will be fetched from the table at any given time. If you are backing up tables with large chunks of binary data (e.g. files stored in BLOB fields) or if you have very large chunks of text stored in the database, the default value - 1000 rows - may cause a PHP memory or MySQL buffer exhaustion.

If you get memory outage errors during the table backup, it is advisable to lower this setting. This is especially true if your MySQL and PHP combination does not allow a cursor to be effectively created and all data has to be transferred in PHP's memory. A value of 20 is a very safe value, at the expense of making your backup process slower and run more queries against your database server. Most servers work fine with the default value of 1000 rows per batch.

Dump PROCEDUREs, FUNCTIONs and TRIGGERs

By default, Akeeba Backup will only back up database tables and VIEWs. If your host supports this, you can also back up and restore advanced aspects of your MySQL database: stored procedures, stored functions and triggers. If your site makes use of any of those features you will have to tick the box. If the backup operation crashes or you the database tables filter page is blank you must turn this option off for Akeeba Backup to work properly.

Warning

Using this feature requires that your host allows you to execute privileged SQL commands against the MySQL database:

- **SHOW PROCEDURE STATUS**
- **SHOW FUNCTION STATUS**
- **SHOW TRIGGERS**

Most shared hosting providers do not allow you to execute these commands. Trying to do so will usually cause the script execution to abruptly halt, most often without indicating the source of error. If you are in doubt, **disable this option** and retry backup. This shouldn't be an issue with dedicated hosting, as long as you grant the **SUPER** privilege to the database user you use to connect to your site's database.

No dependency tracking
: When this option is enabled, Akeeba Backup's database dump engine will no longer try to figure out table and VIEW dependencies. This will speed up the database dump initialization step. This is recommended if and only if you have too many tables (over 200) in your database, you get timeout errors during the database dump initialization step and you do not use foreign keys, VIEWs, FUNCTIONs, PROCEDUREs, TRIGGERs or any tables using the MERGE database engine. If you do use any of those MySQL features in your tables there is a possibility that your backup cannot be restored on an empty database due to unsatisfied references to tables not yet created. Always test your backups if you enable this setting.

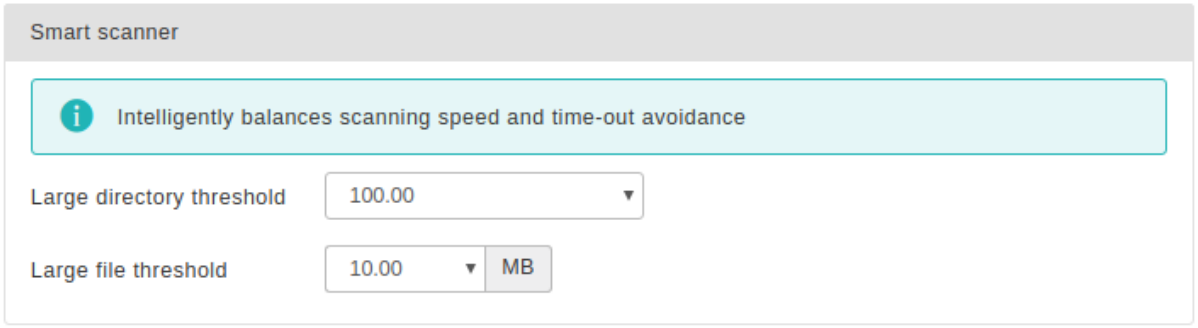
Skip index engine
: Removes USING BTREE and USING HASH from table index definitions in dump files. This is required for restoring to servers which have both indexing engines turned off (e.g. on newest XAMPP versions).

3.3.3. File and directories scanner engines

3.3.3.1. Smart scanner

The Smart Scanner will browse your file system tree for directories and files to include in the backup set, automatically creating a backup step break upon detecting a very large directory which could lead to timeout errors.

Smart Scanner



Large directory threshold
: This option tells Akeeba Backup which directories to consider "large" so that it can break the backup step. When it is encountered with a directory having at least this number of files and subdirectories, it will break the step. The default value is quite conservative and suitable for most sites. If you have a very fast server, e.g. a dedicated server, VPS or MVS, you may increase this value. If you get timeout errors, try decreasing this setting.

Large file threshold
: Normally, Akeeba Backup tries to backup multiple files in a single step in order to provide a fast backup. However, doing that for larger files may result in a timeout or memory outage error. Files bigger than the large file threshold will be backed up in a backup step of their own. If you set it too low you will have a big performance impact in your backup (it will be slower). If you set it too high you might end up with a timeout or memory outage.

  The default setting (10Mb) is fine for most sites. If you are not sure what you're doing you're better off not touching it at all. If you find that your backup consistently fails while backing up a larger file (over 1Mb) you might want to lower this setting, e.g. to 2Mb. If you have a rather big PHP memory limit (128Mb or more) and you can afford the increased memory usage set it to a higher value, e.g. 25Mb (values over that tend to cause issues on all but the higher end dedicated servers).

3.3.3.2. Large site scanner

This engine is specifically optimised for very large sites, containing folders with thousands of files. This is usually the case when you have a huge media collection such as news sites, professional bloggers, companies with a large downloadable reference library or very active business sites storing for example hundreds of invoices daily on the server. In these cases the "Smart scanner" tends to consume unwieldy amounts of memory and CPU time to

compile the list of files to backup, usually leading to timeout or memory outage issues. The "Large site scanner", on the other hand, works just fine by using a specially designed chunked processing technique. The drawback is that it makes the backup approximately 11% slower than the "Smart scanner".

Important

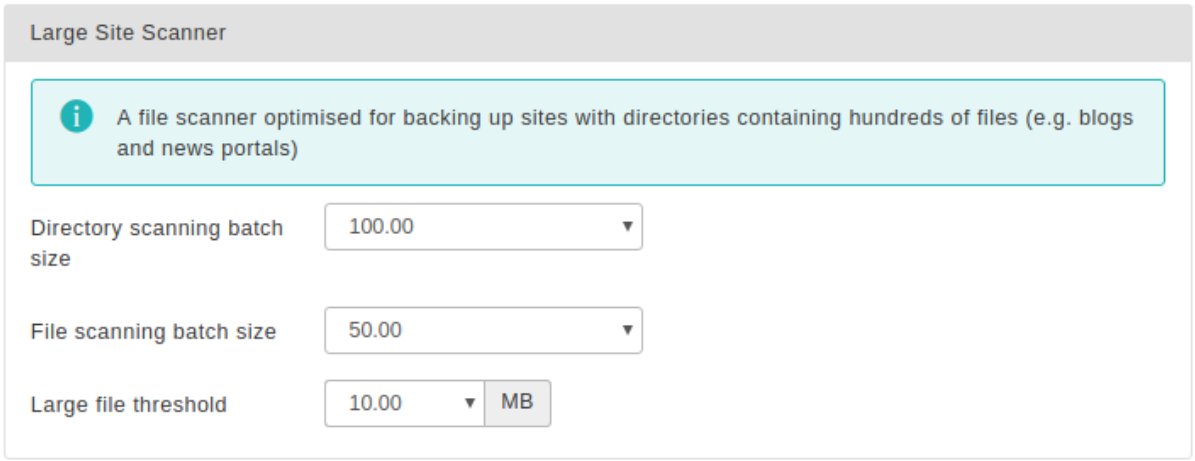
If your backup fails while trying to backup a directory with over 100 files you MUST use the "Large site scanner". It's very likely that this will solve your backup issues.

The developers of Akeeba Backup DO NOT recommend storing several thousands of files in a single directory. Due to reasons that have to do with the way most filesystems work at the Operating System level, the time required to produce a listing of files in a directory or access the files in a directory grows exponentially with the number of files. At about 5000 files the performance impact for accessing the directory, even on a moderately busy server, is big enough to both slow down your site noticeably (adversely impacting your search engine rankings) and make the backup slower and more prone to timeout errors. We strongly recommend using a sane number of subdirectories to logically organise your files and reduce the number of files per directory.

For the technically inclined (we really mean "serious geeks who aspire to do Linux server management as a living"), here is a nice discussion on the subject: http://stackoverflow.com/questions/466521/how-many-files-in-a-directory-is-too-many The problem is that readdir() which is also internally used by PHP only ever reads 32Kb of directory entries at a time. Further down the thread you can see that with 88,000 files in a directory the access becomes ten times slower. Per image. Add that up and you have a dead slow frontpage which is banished to the far end of search indexes. And if you wonder where the 5000 number popped up, it's from http://serverfault.com/questions/129953/maximum-number-of-files-in-one-ext3-directory-while-still-getting-acceptable-per and applies to older Linux distributions without Ext3/4 directory index support or using filesystems without directory index support (e.g. Ext2) which is, of course, the worst case scenario.

In most practical situations, servers become noticeably slow in the frontend and very prone to backup errors due to server timeout or resource usage limits exceeded at about 3000 items (files or folders) inside a directory. We do not recommend storing more than 1000 items inside any directory if you value your sanity.

Large Site scanner



Directory scanning batch size

The Large site scanner creates a listing of folders by scanning a small number of them at a time. This setting determines how much this small number is. The larger this number the faster the backup is, but with the increased possibility of a backup failure on large sites. The smaller this number gets, the slower the backup becomes but the less likely it is to fail. We recommend a setting of 50 for most sites.

If your backup fails on deep nested folders containing many subdirectories we recommend setting this to a lower number, e.g. 20 or even 10. If you have a large PHP maximum memory limit and plenty of memory on your server to spare you may want to increase it to 100 or more. If you are unsure, don't touch this setting.

FIles scanning batch size

The Large site scanner will create a listing of files by scanning a small number of them at a time and then back them up. It will repeat this process until all files in the directory are backed up, then proceed to the next available directory. This setting determines how much this small number of files is. The larger this number the faster the backup is, but with the increased possibility of a backup failure on large sites. The smaller this number gets, the slower the backup becomes but the less likely it is to fail. We recommend a setting of 100 for most sites.

If your backup fails on folders containing many files we recommend setting this to a lower number, e.g. 50 or even 20. If you have a large PHP maximum memory limit and plenty of memory on your server to spare you may want to increase it to 500 or more. If you are unsure, don't touch this setting.

Large file threshold

Normally, Akeeba Backup tries to backup multiple files in a single step in order to provide a fast backup. However, doing that for larger files may result in a timeout or memory outage error. Files bigger than the large file threshold will be backed up in a backup step of their own. If you set it too low you will have a big performance impact in your backup (it will be slower). If you set it too high you might end up with a timeout or memory outage. The default setting (10Mb) is fine for most sites. If you are not sure what you're doing you're better off not touching it at all. If you find that your backup consistently fails while backing up a larger file (over 1Mb) you might want to lower this setting, e.g. to 2Mb. If you have a rather big PHP memory limit (128Mb or more) and you can afford the increased memory usage set it to a higher value, e.g. 25Mb (values over that tend to cause issues on all but the higher end dedicated servers).

3.3.4. Archiver engines

3.3.4.1. ZIP format

The ZIP format is the most well known archive format and is integrated in many operating systems and desktop environments, including Windows™, macOS™, KDE and GNOME.

The ZIP format requires the calculation of CRC32 checksums for each file added in the archive. This is a resource intensive operation which will slow down your backup and may lead to timeouts when archiving big files on slow hosts. If this happens, your only choice is not to use the ZIP format; use JPA instead. Unfortunately, we can't do anything about it: it is a combined limitation of the ZIP specification, how PHP works and how your server is set up.

ZIP Format

Dereference symlinks

Normally, when Akeeba Backup encounters symbolic links ("symlinks"), it follows them and treats them as regular files and directories, backing up their contents. Some site configurations may have symbolic links set up in such a way as to create an infinite loop, causing the backup to fail. When this option is set to `No`, Akeeba Backup will not follow symbolic links, but store their name and their target in the archive. Of course, if your symbolic links use absolute paths, restoring to a different server than the one you backed up from will result in broken symlinks.

> **Note**
>
> Even though Windows 7 supports symbolic links, it does so in a way that it's not possible for PHP to make use of this feature. As a result, this setting will only work on Linux, macOS, FreeBSD and other compatible UNIX-family hosts.

Part size for split archives

Akeeba Backup supports the creation of Split Archives. In a nutshell, your backup archive is spanned among one or several files, so that each of these files ("part") is not bigger than the value you specify here. This is a useful feature for hosts which impose a maximum file size quota. If you use a value of 0Mb, no archive splitting will take place and Akeeba Backup will produce a single backup archive (default).

If you want to post-process your archive files it is suggested that you use small, non-zero values here. The time it takes the post-processing engine to transfer an archive from your server to the remote server equals part size divided by available bandwidth. Since the available execution time is finite and the available bandwidth is constant, the only way to avoid a timeout is creating small parts.

> **Important**
>
> Split ZIP archives can not be opened with 7-zip, Linux unzip and other GUI clients. Only WinZIP and PKZIP understand them. If you want to extract them, you must use WinZIP, PKZIP or Akeeba Kickstart. This is not an Akeeba Backup "bug", it's a problem with most free archiver extraction tools.

Chunk size for large files processing

Each file is read in small increments, we call chunks, while being copied in the archive. Larger chunks will result in faster backup, at the price of taking longer to process each one of them and risking a timeout. Smaller chunks lead to slower but safer backups. On very slow hosts, this parameter should be set to a low value, for example 256Kb, or even lower - especially true if you constantly get timeout errors when backing up large files. On fast hosts you may want to increase this value in order to speed up your backup operation.

Big file threshold

Files over this size will be stored in the archive file uncompressed. Do note that in order for a file to be compressed, Akeeba Backup has to load it in its entirety to memory, compress it and then write it to disk. As a rule of thumb, you need to have free memory equal to 1.8 times the size of the file to compress, e.g. 18Mb for a 10Mb file. Joomla! with a lot of plug-ins might consume as much as 16Mb and Akeeba Backup's engine might consume another 5Mb, so plan this value carefully, or you will run into memory exhaustion errors. Compression is also resource intensive and will increase the time to produce a backup. If this value is too high, you might run into timeout errors.

Chunk size for Central Directory processing

At the end of the ZIP archive creation we have to attach a lookup table containing the names of all included files to the end of the archive file. This table is called the Central Directory. We have to do this in small chunks so as to avoid timeout or memory exhaustion errors. It is recommended that you leave the default value (1Mb) unless you know what you're doing.

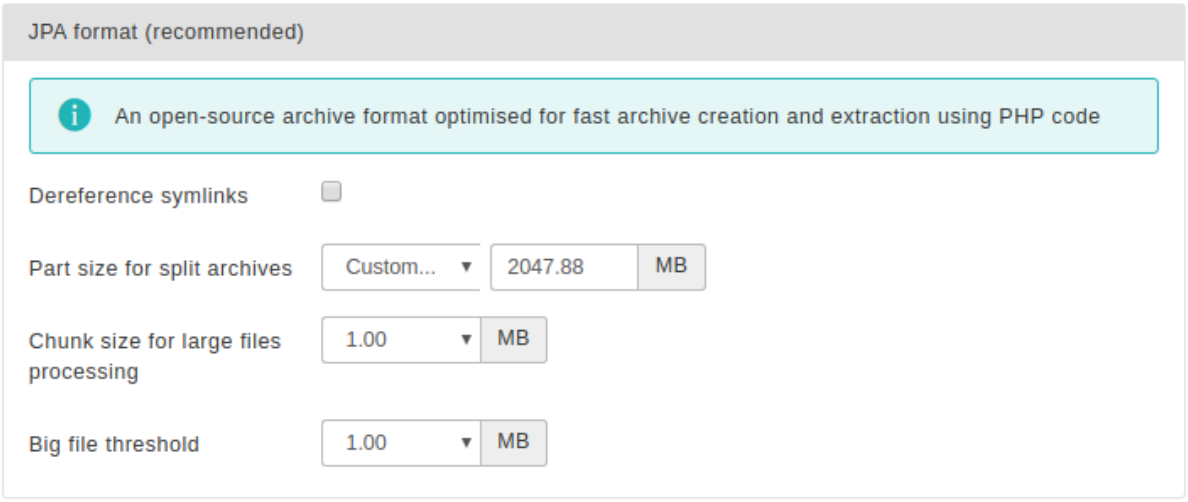
3.3.4.2. JPA format

The JPA format was conceived as an alternative to ZIP, designed to be extremely suitable for PHP scripts. The trick is that the JPA format doesn't store a checksum for each file - therefore it reduces the processing overhead during archiving - and it doesn't use a "lookup table" (central directory) as ZIP does. Both of these design decisions lead to extremely fast, low resource usage archiving processes.

Tip

It is recommended that you use the JPA format for all of your backups. You can extract JPA files using Kickstart.

JPA Format



The settings for this engine are identical to those used in the ZIP engine.

3.3.4.3. Encrypted Archives (JPS format)

Note

This feature is only available in the Akeeba Backup Professional release.

The JPS is a further evolution of the JPA format, designed with the major goals of improving compression rations and enhancing the security of your data by encrypting the entire archive's contents with the industry standard AES-128 encryption format. The latter goal ensures that even in the unlikely event of your backup files ending up in the hands of hacker or another untrusted party, they would be useless. As per the strictest security standards, all information in the archive (including file names and file data) are encrypted. Without the password nobody can deduct any information about your site by examining a JPS archive. The contents of all files in the archive are compressed and encrypted in 64Kb blocks, allowing for better compression ratios over the JPA format.

In order for JPS to work it requires that both the zlib and mcrypt or OpenSSL PHP extensions are installed and activated on your server. Please keep in mind that even if your site is using HTTPS this doesn't mean that you have the OpenSSL *PHP extension* installed. You usually have to ask your host to enable it for you. Moreover, the mcrypt or openssl library installed on the server must support AES-128 in CBC mode. If any of these conditions is not met, the backup process will halt with an error mentioning that encryption is not enabled on your server. In this case, please contact your host with the information in this paragraph so that they can perform the necessary server-side changes.

Important

We STRONGLY recommend using long (64 or more characters), completely random passwords which make use of lowercase and uppercase Latin letters, numbers and special characters (top row on US-format keyboards). Use a password manager to generate and store these passwords. Taking these precautions make password brute forcing with conventional technology highly impractical in the foreseeable future.

JPS Format

The settings for this engine are:

Encryption key
: This is the password to be used for encrypting the archive. For the sake of security, you are encouraged to enter a long passphrase which is hard to guess.

Warning

The key is case sensitive. This means that Abc, ABC and abc are three *completely different* keys!

Dereference symlinks
: This setting is only valid on Linux and compatible *NIX hosts. Normally, when Akeeba Backup encounters symbolic links ("symlinks"), it follows them and treats them as regular files and directories, backing up their contents. Some site configurations may have symbolic links set up in such a way as to create an infinite loop, causing the backup to fail. When this option is set to `No`, Akeeba Backup will not follow symbolic links, but store their name and their target in the archive. Of course, if your symbolic links use absolute paths, restoring to a different server than the one you backed up from will result in broken symlinks.

Note

Even though Windows 7 supports symbolic links, it does so in a way that it's not possible for PHP to make use of this feature. As a result, this setting will only work on Linux, macOS, FreeBSD and other compatible UNIX-family hosts.

Part size for split archives
: Akeeba Backup supports the creation of Split Archives. In a nutshell, your backup archive is spanned among one or several files, so that each of these files ("part") is not bigger than the value you specify here. This is a useful feature for hosts which impose a maximum file size quota. If you use a value of 0Mb, no archive splitting will take place and Akeeba Backup will produce a single backup archive (default).

  If you want to post-process your archive files it is suggested that you use small, non-zero values here. The time it takes the post-processing engine to transfer an archive from your server to the remote server equals part size divided by available bandwidth. Since the available execution time is finite and the available bandwidth is constant, the only way to avoid a timeout is creating small parts.

3.3.4.4. DirectFTP

Important

This feature is not meant for everyday users. It is designed for web professionals. If you don't understand the rest of this section, please do not use it. Akeeba Backup is equally useful as a site migration tool without using DirectFTP.

The DirectFTP engine allows power users to directly export a website from one server to another, without the need to download the backup file to their PC, upload it and extract it on the other server. In order to do so, instead of backing up to an archive, it directly writes the backed up files to the remote server using FTP, hence the name.

This engine uses PHP's native FTP functions. This may not work if your host has disabled PHP's native FTP functions or if your remote FTP server is incompatible with them. In this case you may want to use the DirectFTP over cURL engine instead.

Do note that when using the DirectFTP engine, the post-processing engine will not run, as there is no archive produced.

In a nutshell, when this option is activated, Akeeba Backup operates as usual, backing up your database and files. Instead of putting the site files, installer files and database dump inside a backup archive, it transfers them to a remote server using FTP. You can then visit the installation URL on the remote server to complete the site transfer progress.

This is considered an advanced feature. Since there are many things which might go wrong in the process and due to the fact that the success of the operation depends on the server configuration of both the originating and target servers, you are advised not to use it unless you know what you're doing.

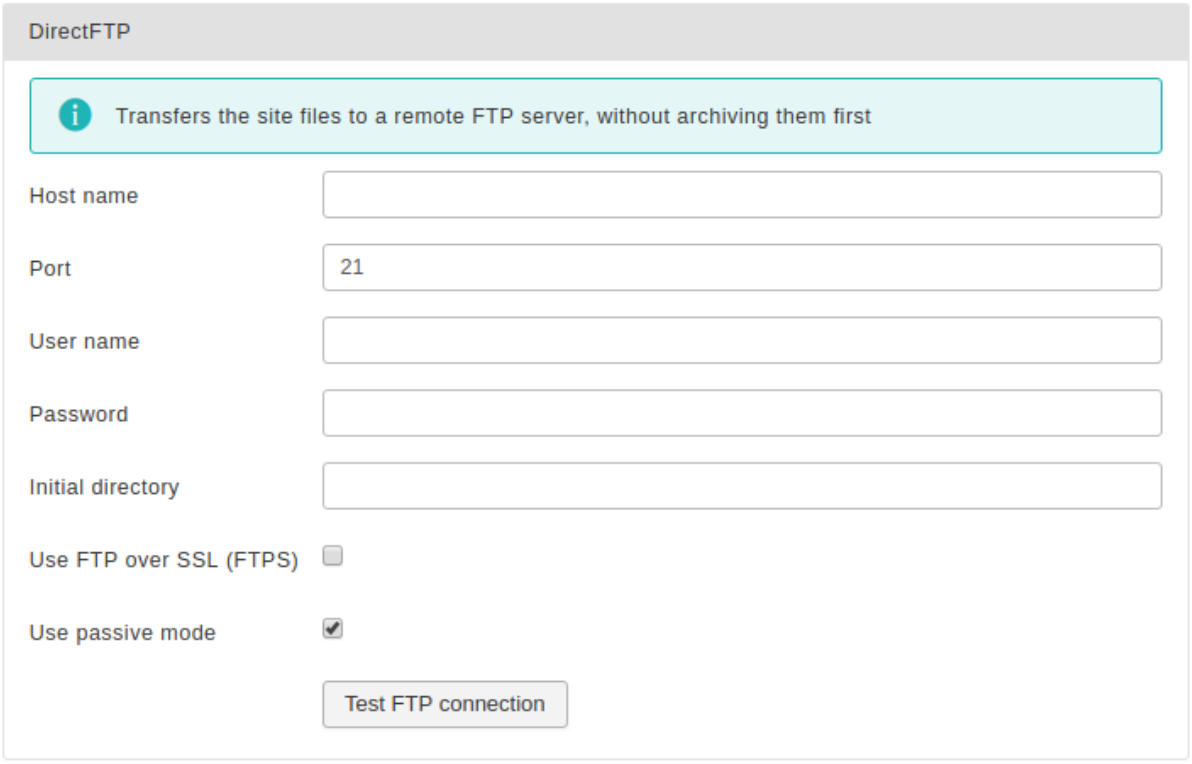
Moreover, bear in mind that the target server *must not* contain any files! If it does, it may not be possible to overwrite them, leading to an incomplete site transfer.

Finally note that due to the backup process being split in several steps (to avoid web server timeouts) a new FTP connection has to be created on each backup step, i.e. for every few files uploaded to your remote server. At best this makes the transfer extremely slow. At worst, your remote FTP server will decline further connections because it sees the same remote IP opening and closing FTP connections in rapid succession. We strongly recommend uploading entire backup archives using the post-processing engines instead of using this feature.

Your originating server must support PHP's FTP extensions and not have its FTP functions blocked. Your originating server must not block FTP communication to the remote (target) server. Some hosts apply a firewall policy which requires you to specify to which hosts your server can connect. In such a case you might need to allow communication to your remote host.

Normally, remote FTP connections consume a lot of time, therefore DirectFTP is very prone to time-outs. Theoretically, Akeeba Backup can automatically estimate the time required for transferring each file and avoid timing out. However, this is not always technically possible. In such a case you might want to lower the maximum execution time allowance and bias in the Configuration. Do note that large files have to be transferred in a single step, as most PHP and FTP configuration combinations disallow resuming uploads (chunked uploads). This means that a very large file, or a very large database dump may cause the process to fail with a timeout error.

DirectFTP



The available configuration options are:

- **Host name**. The hostname of your remote (target) server, e.g. `ftp.example.com`. You must NOT enter the ftp:// protocol prefix. If you do, Akeeba Backup will try to remove it automatically and issue a warning about it.
- **Port**. The TCP/IP port of your remote host's FTP server. It's usually 21.
- **User name**. The username you have to use to connect to the remote FTP server.
- **Password**. The password you have to use to connect to the remote FTP server.
- **Initial directory**. The absolute FTP directory to your remote site's location where your site will be cloned to. This is provided by your hosting company. Do not ask us to tell you what you should put in here because we can't possibly know. There is an easy way to find it, though. Connect to your target FTP server with FileZilla. Navigate to the web server's root (usually it's a subdirectory named httpdocs, htdocs, public_html, http_docs or www). Above the right-hand folder pane you will see a text box with a path. Copy this path and paste it to Akeeba Backup's setting.
- **Use FTP over SSL**. If your remote server supports secure FTP connections over SSL (they have to be explicit SSL; implicit SSL is not supported), you can enable this feature. In such a case you will most probably *have* to change the port. Please ask your hosting company to provide you with more information on whether they support this feature and what port you should use. You must note that this feature must also be supported by your originating server as well.
- **Use passive mode**. Normally you should enable it, as it is the most common and firewall-safe transfer mode supported by FTP servers. Sometimes, you remote server might require active FTP transfers. In such a case please disable this, but bear in mind that your originating server might not support active FTP transfers, which usually requires tweaking the firewall!

3.3.4.5. DirectFTP over cURL

Important

This feature is not meant for everyday users. It is designed for web professionals. If you don't understand the rest of this section, please do not use it. Akeeba Backup is equally useful as a site migration tool without using DirectFTP.

The DirectFTP over cURL engine allows power users to directly export a website from one server to another, without the need to download the backup file to their PC, upload it and extract it on the other server. In order to do so, instead of backing up to an archive, it directly writes the backed up files to the remote server using FTP, hence the name.

This engine uses PHP's cURL functions. This may not work if your host has not installed or enabled the cURL functions. In this case you may want to use the DirectFTP engine instead.

Do note that when using the DirectFTP engine, the post-processing engine will not run, as there is no archive produced.

In a nutshell, when this option is activated, Akeeba Backup operates as usual, backing up your database and files. Instead of putting the site files, installer files and database dump inside a backup archive, it transfers them to a remote server using FTP. You can then visit the installation URL on the remote server to complete the site transfer progress.

This is considered an advanced feature. Since there are many things which might go wrong in the process and due to the fact that the success of the operation depends on the server configuration of both the originating and target servers, you are advised not to use it unless you know what you're doing.

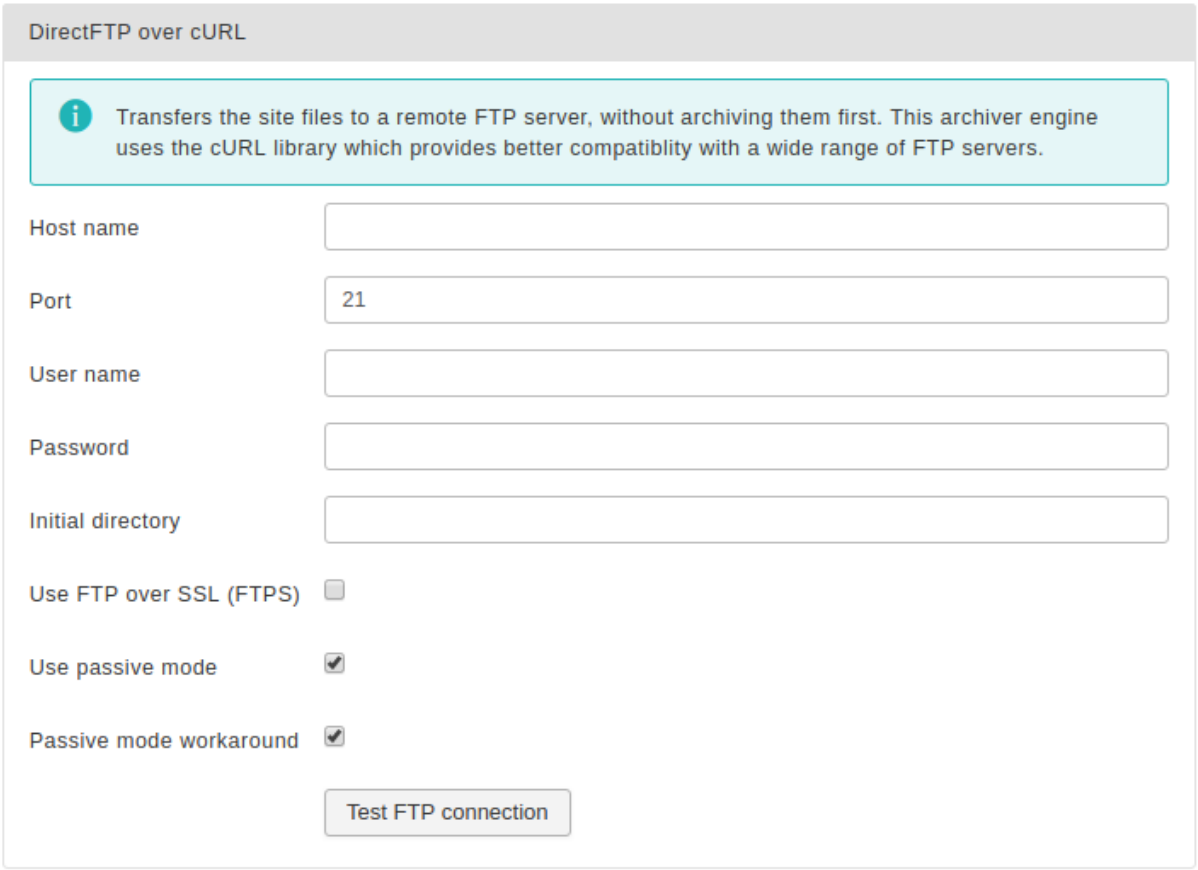
Moreover, bear in mind that the target server *must not* contain any files! If it does, it may not be possible to overwrite them, leading to an incomplete site transfer.

Finally note that due to the nature of the cURL library over a new FTP connection has to be created for each and every file uploaded to your remote server. At best this makes the transfer extremely slow. At worst, your remote FTP server will decline further connections because it sees the same remote IP opening and closing FTP connections in rapid succession. We strongly recommend uploading entire backup archives using the post-processing engines instead of using this feature.

Your originating server must support PHP's cURL extension. Your originating server must not block FTP communication to the remote (target) server. Some hosts apply a firewall policy which requires you to specify to which hosts your server can connect. In such a case you might need to allow communication to your remote host.

Normally, remote FTP connections consume a lot of time, therefore DirectFTP over cURL is very prone to time-outs. Theoretically, Akeeba Backup can automatically estimate the time required for transferring each file and avoid timing out. However, this is not always technically possible. In such a case you might want to lower the maximum execution time allowance and bias in the Configuration. Do note that large files have to be transferred in a single step, as most PHP and FTP configuration combinations disallow resuming uploads (chunked uploads). This means that a very large file, or a very large database dump may cause the process to fail with a timeout error.

DirectFTP over cURL



The available configuration options are:

- **Host name**. The hostname of your remote (target) server, e.g. `ftp.example.com`. You must NOT enter the ftp:// protocol prefix. If you do, Akeeba Backup will try to remove it automatically and issue a warning about it.
- **Port**. The TCP/IP port of your remote host's FTP server. It's usually 21.
- **User name**. The username you have to use to connect to the remote FTP server.
- **Password**. The password you have to use to connect to the remote FTP server.
- **Initial directory**. The absolute FTP directory to your remote site's location where your site will be cloned to. This is provided by your hosting company. Do not ask us to tell you what you should put in here because we can't possibly know. There is an easy way to find it, though. Connect to your target FTP server with FileZilla. Navigate to the web server's root (usually it's a subdirectory named httpdocs, htdocs, public_html, http_docs or www). Above the right-hand folder pane you will see a text box with a path. Copy this path and paste it to Akeeba Backup's setting.
- **Use FTP over SSL**. If your remote server supports secure FTP connections over SSL (they have to be explicit SSL; implicit SSL is not supported), you can enable this feature. In such a case you will most probably *have* to change the port. Please ask your hosting company to provide you with more information on whether they support this feature and what port you should use. You must note that this feature must also be supported by your originating server as well.
- **Use passive mode**. Normally you should enable it, as it is the most common and firewall-safe transfer mode supported by FTP servers. Sometimes, you remote server might require active FTP transfers. In such a case please disable this, but bear in mind that your originating server might not support active FTP transfers, which usually requires tweaking the firewall!

- **Passive mode workaround**. Some badly configured / misbehaving servers report the wrong IP address when you enable the passive mode. Usually they report their internal network IP address (something like 127.0.0.1 or 192.168.1.123) instead of their public, Internet-accessible IP address. This erroneous information confuses FTP information, causing uploads to stall and eventually fail. Enabling this workaround option instructs cURL to ignore the IP address reported by the server and instead use the server's public IP address, as seen by your server. In most cases this works much better, therefore we recommend leaving this option turned on if you're not sure. You should only disable it in case of an exotic setup where the FTP server uses two different public IP addresses for the control and data channels.

3.3.4.6. DirectSFTP

Important

This feature is only available in the Akeeba Backup Professional edition.

Important

This feature is not meant for everyday users. It is designed for web professionals. If you don't understand the rest of this section, please do not use it. Akeeba Backup is equally useful as a site migration tool without using DirectSFTP.

The DirectSFTP engine allows power users to directly export a website from one server to another, without the need to download the backup file to their PC, upload it and extract it on the other server. In order to do so, instead of backing up to an archive, it directly writes the backed up files to the remote server using SFTP (Secure File Transfer Protocol over SSH), hence the name.

This engine uses the PHP extension called SSH2. The SSH2 extension is still marked as an alpha and is not enabled by default or even provided by many commercial hosts. In this case you may want to use the DirectSFTP over cURL engine instead which uses PHP's cURL extension, available on most hosts.

Do note that when using the DirectSFTP engine, the post-processing engine will not run, as there is no archive produced.

In a nutshell, when this option is activated, Akeeba Backup operates as usual, backing up your database and files. Instead of putting the site files, installer files and database dump inside a backup archive, it transfers them to a remote server using SFTP. You can then visit the installation URL on the remote server to complete the site transfer progress.

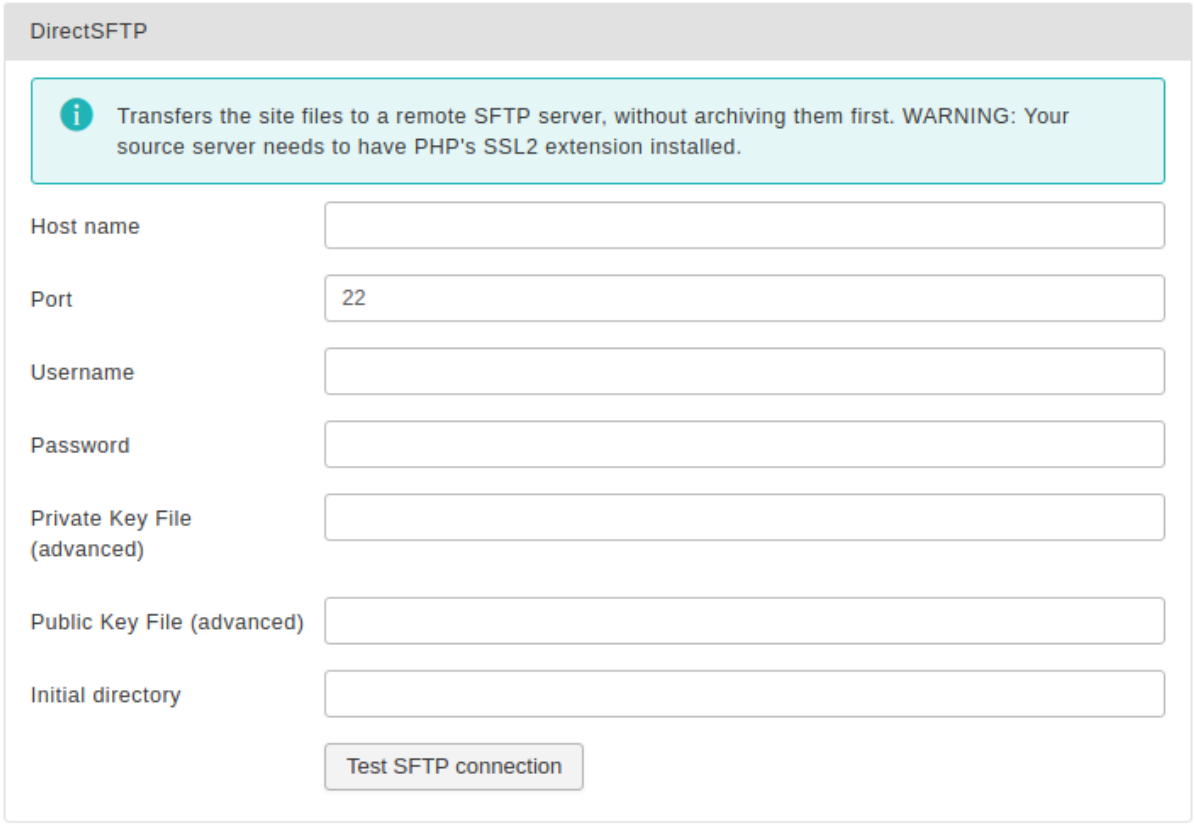
This is considered an advanced feature. Since there are many things which might go wrong in the process and due to the fact that the success of the operation depends on the server configuration of both the originating and target servers, you are advised not to use it unless you know what you're doing.

Moreover, bear in mind that the target server *must not* contain any files! If it does, it may not be possible to overwrite them, leading to an incomplete site transfer.

Your originating server (where you are backing up *from*) must a. support PHP's SSH2 extensions, b. allow outbound TCP/IP connections to your target host's SSH port and c. not have the SFTP functions of the SSH2 extension blocked. Please note that some hosts apply a firewall policy which requires you to specify to which hosts your server can connect. In such a case you might need to allow communication to your remote host.

Normally, remote SFTP connections consume a lot of time, therefore DirectSFTP is very prone to time-outs. Theoretically, Akeeba Backup can automatically estimate the time required for transferring each file and avoid timing out. However, this is not always technically possible. In such a case you might want to lower the maximum execution time allowance and bias in the Configuration. Do note that large files have to be transferred in a single step, as most PHP and SFTP configuration combinations disallow resuming uploads (chunked uploads). This means that a very large file, or a very large database dump may cause the process to fail with a timeout error.

DirectSFTP



The available configuration options are:

- **Host name**. The hostname of your remote (target) server, e.g. `sftp.example.com`. You must NOT enter the sftp:// or ssh:// protocol prefix. If you do, Akeeba Backup will try to remove it automatically and issue a warning about it.
- **Port**. The TCP/IP port of your remote host's FTP server. It's usually 22.
- **User name**. The username you have to use to connect to the remote SFTP server. This field must always be used, even when you're using certificate authentication.
- **Password**. The password you have to use to connect to the remote SFTP server. If you are using certificate authentication please enter the encryption key of your private key file. However, if you're using certificate authentication and your private key file is not encrypted please leave this field blank.
- **Private Key File (advanced)**. Only use this field when you want to perform certificate authentication. Enter the absolute path to your private SSH key file. If your private key file is encrypted with a password please provide the password in the Password field above.

  #### Important

  If the libssh2 library that the SSH2 extension of PHP is using is compiled against GnuTLS (instead of OpenSSL) you will NOT be able to use encrypted private key files. This has to do with bugs / missing features of GnuTLS, not our code. If you can't get certificate authentication to work please try providing an unencrypted private key file and leave the Password field blank.

- **Public Key File (advanced)**. Only use this field when you want to perform certificate authentication. Enter the absolute path to your public SSH key file.
- **Initial directory**. The absolute FTP directory to your remote site's location where your site will be cloned to. This is provided by your hosting company. Do not ask us to tell you what you should put in here because we

can't possibly know. There is an easy way to find it, though. Connect to your target FTP server with FileZilla. Navigate to the web server's root (usually it's a subdirectory named httpdocs, htdocs, public_html, http_docs or www). Above the right-hand folder pane you will see a text box with a path. Copy this path and paste it to Akeeba Backup's setting.

3.3.4.7. DirectSFTP over cURL

Important

This feature is only available in the Akeeba Backup Professional edition.

Important

This feature is not meant for everyday users. It is designed for web professionals. If you don't understand the rest of this section, please do not use it. Akeeba Backup is equally useful as a site migration tool without using DirectSFTP.

The DirectSFTP engine allows power users to directly export a website from one server to another, without the need to download the backup file to their PC, upload it and extract it on the other server. In order to do so, instead of backing up to an archive, it directly writes the backed up files to the remote server using SFTP (Secure File Transfer Protocol over SSH), hence the name.

This engine uses the PHP cURL extension. If your host has disabled the cURL extension but has enabled the SSH2 PHP extension you may want to use the DirectSFTP engine instead which uses PHP's SSH2 extension.

Do note that when using the DirectSFTP engine, the post-processing engine will not run, as there is no archive produced.

In a nutshell, when this option is activated, Akeeba Backup operates as usual, backing up your database and files. Instead of putting the site files, installer files and database dump inside a backup archive, it transfers them to a remote server using SFTP. You can then visit the installation URL on the remote server to complete the site transfer progress.

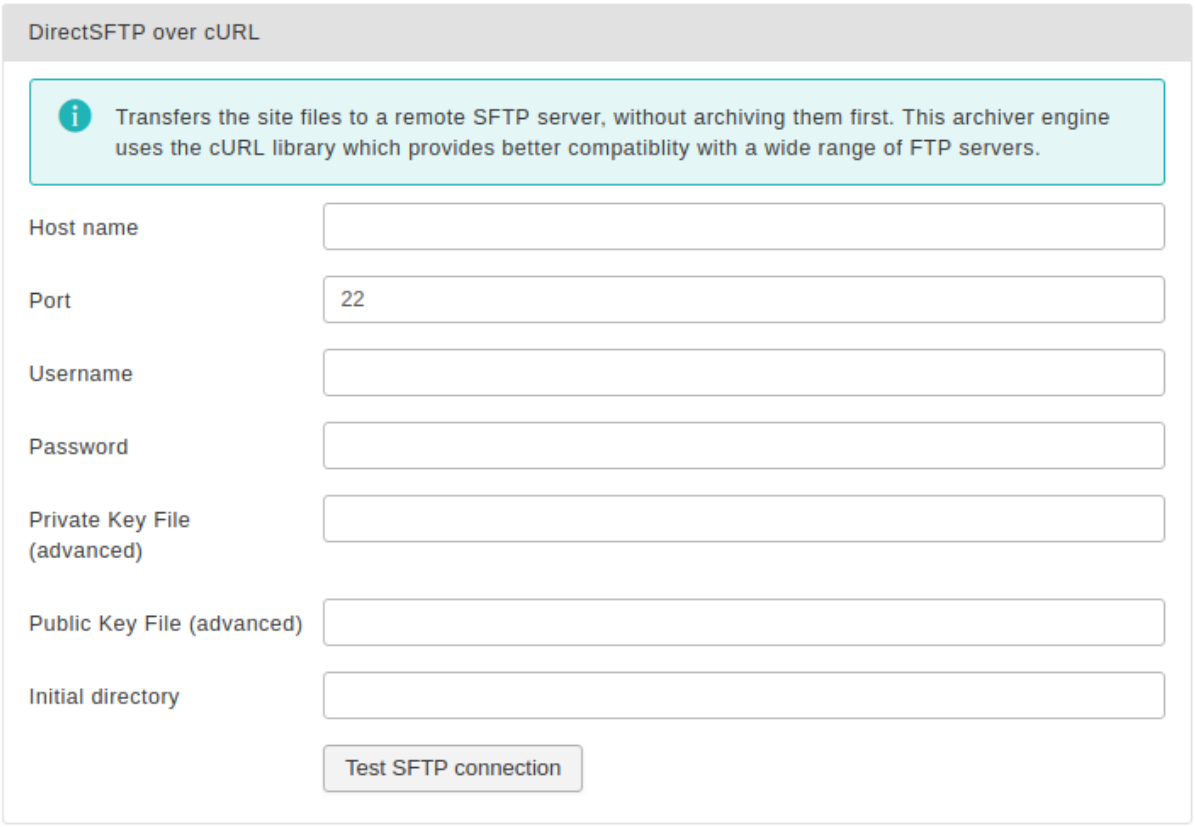
This is considered an advanced feature. Since there are many things which might go wrong in the process and due to the fact that the success of the operation depends on the server configuration of both the originating and target servers, you are advised not to use it unless you know what you're doing.

Moreover, bear in mind that the target server *must not* contain any files! If it does, it may not be possible to overwrite them, leading to an incomplete site transfer.

Your originating server (where you are backing up *from*) must a. have PHP's cURL extension installed and activated, b. have the cURL extension compiled with SFTP support and c. allow outbound TCP/IP connections to your target host's SSH port. Please note that some hosts provide the cURL extension without SFTP support. This feature will NOT work on these hosts. Moreover, some hosts apply a firewall policy which requires you to specify to which hosts your server can connect. In such a case you might need to allow communication to your remote host.

Normally, remote SFTP connections consume a lot of time, therefore DirectSFTP is very prone to time-outs. Theoretically, Akeeba Backup can automatically estimate the time required for transferring each file and avoid timing out. However, this is not always technically possible. In such a case you might want to lower the maximum execution time allowance and bias in the Configuration. Do note that large files have to be transferred in a single step, as most PHP and SFTP configuration combinations disallow resuming uploads (chunked uploads). This means that a very large file, or a very large database dump may cause the process to fail with a timeout error.

DirectSFTP over cURL



The available configuration options are:

- **Host name**. The hostname of your remote (target) server, e.g. `sftp.example.com`. You must NOT enter the sftp:// or ssh:// protocol prefix. If you do, Akeeba Backup will try to remove it automatically and issue a warning about it.
- **Port**. The TCP/IP port of your remote host's FTP server. It's usually 22.
- **User name**. The username you have to use to connect to the remote SFTP server. This field must always be used, even when you're using certificate authentication.
- **Password**. The password you have to use to connect to the remote SFTP server. If you are using certificate authentication please enter the encryption key of your private key file. However, if you're using certificate authentication and your private key file is not encrypted please leave this field blank.
- **Private Key File (advanced)**. Only use this field when you want to perform certificate authentication. Enter the absolute path to your private SSH key file. If your private key file is encrypted with a password please provide the password in the Password field above.

  #### Important

  If cURL is compiled against GnuTLS (instead of OpenSSL) you will NOT be able to use encrypted private key files. This has to do with bugs / missing features of GnuTLS, not our code. If you can't get certificate authentication to work please try providing an unencrypted private key file and leave the Password field blank.

- **Public Key File (advanced)**. Only use this field when you want to perform certificate authentication. Enter the absolute path to your public SSH key file. This is optional: some versions of the cURL library allow you to not provide a public key file, using the information of the private key file to derive this information. If in doubt, always provide both private and public key files to perform certificate authentication.

- **Initial directory**. The absolute FTP directory to your remote site's location where your site will be cloned to. This is provided by your hosting company. Do not ask us to tell you what you should put in here because we can't possibly know. There is an easy way to find it, though. Connect to your target FTP server with FileZilla. Navigate to the web server's root (usually it's a subdirectory named httpdocs, htdocs, public_html, http_docs or www). Above the right-hand folder pane you will see a text box with a path. Copy this path and paste it to Akeeba Backup's setting.

3.3.4.8. ZIP using ZIPArchive class

This engine produces ZIP archive using PHP's built-in ZIP archive class. We have not found any practical use case where this is preferable to Akeeba Backup's own ZIP archiver. This is why we wrote the ZIP archiver in the first place.

The only reason for this ethod's existence was to prove that PHP's ZipArchiver is unsuitable for backing up sites to those very few but extremely vocal users who claimed otherwise.

In short, its only reason of existence is to prove it shouldn't exist in the first place. So, please, do not use it. If you do, your backups will fail at worst, you will be wasting server resources (especially lots of memory) at best.

3.3.5. Data processing engines

3.3.5.1. No post-processing

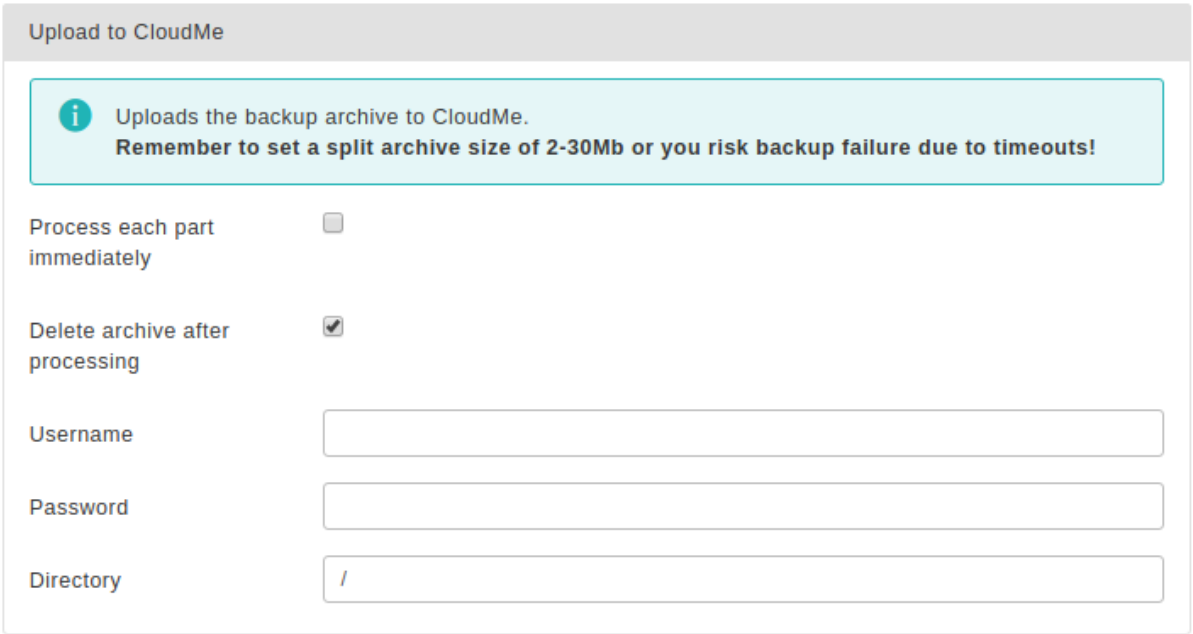
This is the default setting and the only one one available to Akeeba Backup Core. It does no post-processing. It simply leaves the backup archives on your server.

3.3.5.2. Upload to CloudMe

> **Note**
>
> This feature is available only to Akeeba Backup Professional.

Using this engine, you can upload your backup archives to the European cloud storage service CloudMe.

Upload to CloudMe



The required settings for this engine are:

Process each part immediately	If you enable this, each backup part will be uploaded as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the upload fails, the backup fails. If you don't enable this option, the upload process will take place after the backup is complete and finalized. This ensures that if the upload process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.
Delete archive after processing	If enabled, the archive files will be removed from your server after they are uploaded to CloudMe.
Username	Your CloudMe username
Password	Your CloudMe password
Directory	The directory inside your CloudMe Blue Folder™ where your files will be stored in. If you want to use subdirectories, you have to use a forward slash, e.g. `directory/subdirectory/subsubdirectory`. You can use Akeeba Backup's "variables" in the directory name in order to create it dynamically. These are the same variables as what you can use in the archive name, i.e. [DATE], [TIME], [HOST], [RANDOM].

3.3.5.3. Upload to Microsoft Windows Azure BLOB Storage service

Note

This feature is available only to Akeeba Backup Professional.

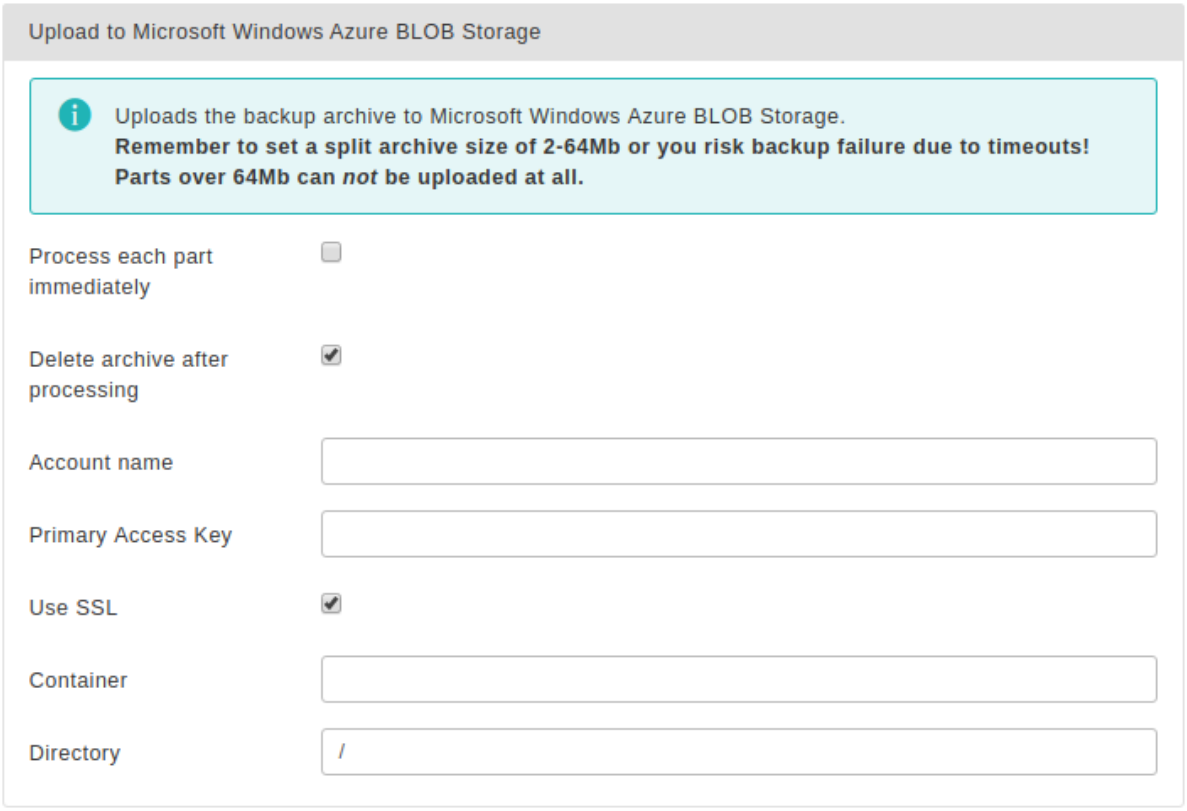
Using this engine, you can upload your backup archives to the Microsoft Windows Azure BLOB Storage [http://www.microsoft.com/windowsazure/windowsazure/] cloud storage service. This new cloud storage service from Microsoft is reasonably priced (the cost is very close to CloudFiles) and quite fast, with lots of local endpoints around the globe.

Azure, typically has a low limit for storing files, depending on container options. In some cases it may be as low as 64MB. Keep that in mind if your backups keep failing.

Before you begin, you should know the limitations. Like most cloud storage providers, Azure does not allow appending to files, so the archive has to be transferred in a single step. PHP has a time limit restriction we can't overlook. The time required to upload a file to Azure equals the size of the file divided by the available bandwidth. We want to time to upload a file to be less than PHP's time limit restriction so as to avoid timing out. Since the available bandwidth is finite and constant, the only thing we can reduce in order to avoid timeouts is the file size. To this end, you have to produce split archives, by setting the part size for archive splitting in ZIP's or JPA's engine configuration pane. The suggested values are between 10 and 20 MB.

If you use the native CRON mode (akeeba-backup.php), there is usually no time limit - or there is a very high time limit in the area of 3 minutes or so. Ask your host about it. Setting up a profile for use only with the native CRON mode allows you to increase the part size and reduce the number of parts a complete backup consists of.

Upload to Microsoft Windows Azure BLOB Storage



The required settings for this engine are:

Process each part immediately	If you enable this, each backup part will be uploaded as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the upload fails, the backup fails. If you don't enable this option, the upload process will take place after the backup is complete and finalized. This ensures that if the upload process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.
Delete archive after processing	If enabled, the archive files will be removed from your server after they are uploaded to Azure.
Account name	The account name for your Microsoft Azure subscription. If your endpoint looks like `foobar.blobl.core.windows.net` then your account name is foobar.
Primary Access Key	You can find this Key in account page. It is lengthy and always ends in double equals marks.
Container	The name of the Azure container where you want to store your archives in.
Directory	The directory inside your Azure container where your files will be stored in. If you want to use subdirectories, you have to use a forward slash, e.g. `/directory/subdirectory/subsubdirectory`. Leave blank to store the files on the container's root.

3.3.5.4. Upload to RackSpace CloudFiles

Note

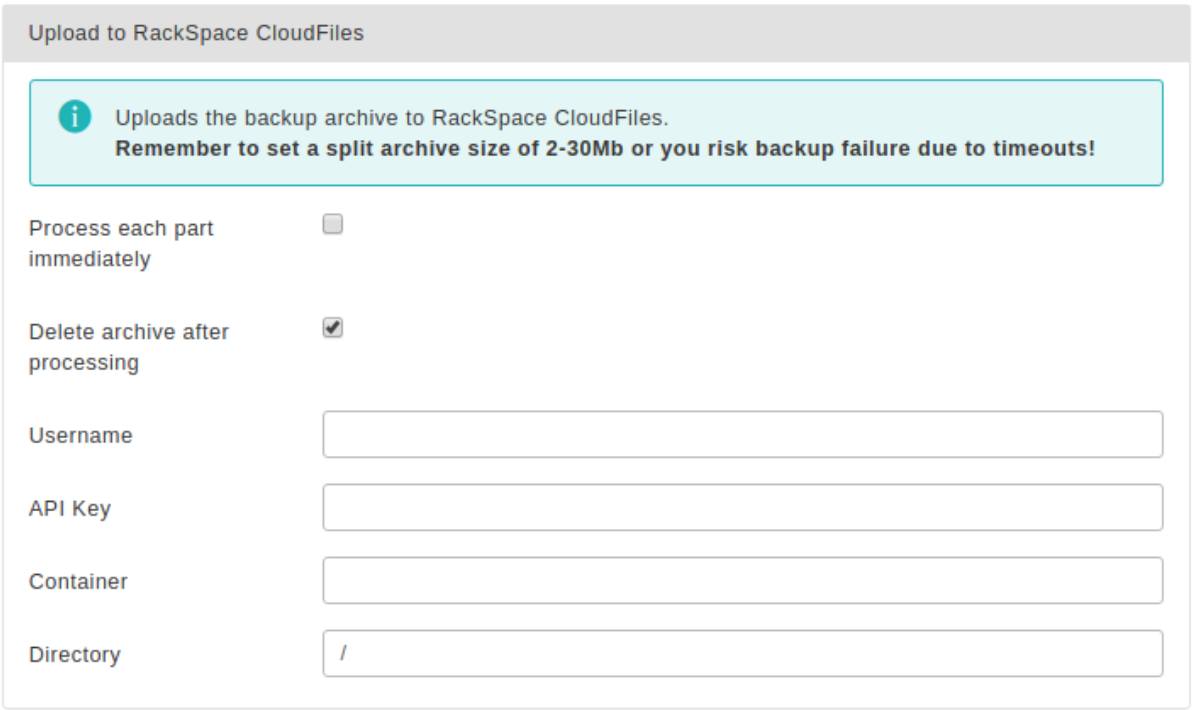
This feature is available only to Akeeba Backup Professional.

Using this engine, you can upload your backup archives to the RackSpace CloudFiles [www.rackspacecloud.com/cloud_hosting_products/files] cloud storage service. This service had previously been called Mosso.

Before you begin, you should know the limitations. As most cloud storage providers, CloudFiles does not allow appending to files, so the archive has to be transferred in a single step. PHP has a time limit restriction we can't overlook. The time required to upload a file to CloudFiles equals the size of the file divided by the available bandwidth. We want to time to upload a file to be less than PHP's time limit restriction so as to avoid timing out. Since the available bandwidth is finite and constant, the only thing we can reduce in order to avoid timeouts is the file size. To this end, you have to produce split archives, by setting the part size for archive splitting in ZIP's or JPA's engine configuration pane. The suggested values are between 10 and 20 MB.

If you use the native CRON mode (akeeba-backup.php), there is usually no time limit - or there is a very high time limit in the area of 3 minutes or so. Ask your host about it. Setting up a profile for use only with the native CRON mode allows you to increase the part size and reduce the number of parts a complete backup consists of.

Upload to RackSpace CloudFiles



The required settings for this engine are:

Process each part immediately	If you enable this, each backup part will be uploaded as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the upload fails, the backup fails. If you don't enable this option, the upload process will take place after the backup is complete and finalized. This ensures that if the upload process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.
Delete archive after processing	If enabled, the archive files will be removed from your server after they are uploaded to CloudFiles.
Username	The username assigned to you by the RackSpace CloudFiles service
API Key	The API Key found in your CloudFiles account
Container	The name of the CloudFiles container where you want to store your archives in.

Directory — The directory inside your CloudFiles container where your files will be stored in. If you want to use subdirectories, you have to use a forward slash, e.g. `/directory/subdirectory/subsubdirectory`. Leave blank to store the files on the container's root.

3.3.5.5. Upload to OVH Object Storage

Note

This feature is available only to Akeeba Backup Professional.

Using this engine, you can upload your backup archives to the OVH Object Storage cloud storage service. This allows you to upload files into OVH's public cloud, powered by the OpenStack technology.

Before you begin, you should know the limitations. As most cloud storage providers, OVH does not allow appending to files, so the archive has to be transferred in a single step. PHP has a time limit restriction we can't overlook. The time required to upload a file to OVH equals the size of the file divided by the available bandwidth. We want to time to upload a file to be less than PHP's time limit restriction so as to avoid timing out. Since the available bandwidth is finite and constant, the only thing we can reduce in order to avoid timeouts is the file size. To this end, you have to produce split archives, by setting the part size for archive splitting in ZIP's or JPA's engine configuration pane. The suggested values are between 10 and 20 MB.

If you use the native CRON mode (akeeba-backup.php), there is usually no time limit - or there is a very high time limit in the area of 3 minutes or so. Ask your host about it. Setting up a profile for use only with the native CRON mode allows you to increase the part size and reduce the number of parts a complete backup consists of.

Before you begin

You will need to set up object storage and collect some necessary but not necessarily obvious information from your OVH account. You can do so through OVH's Cloud Manager [https://www.ovh.com/manager/cloud/index.html#/] portal.

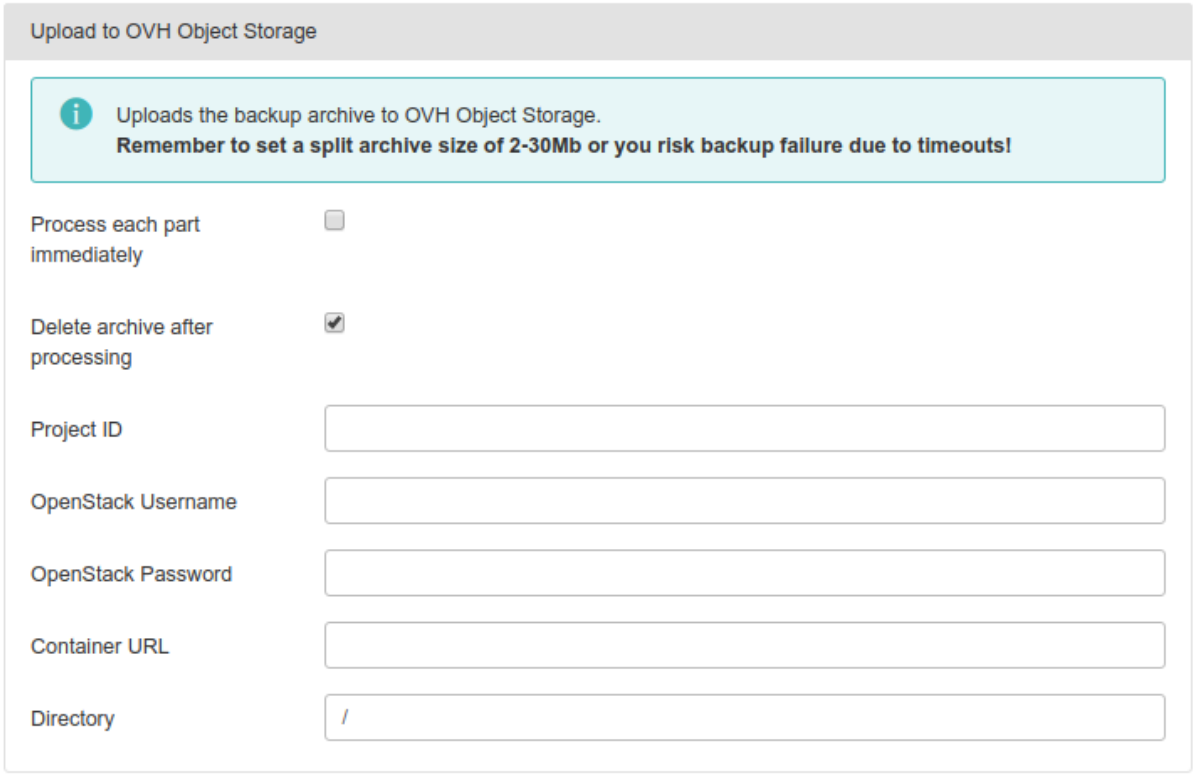
From the left side menu click on Servers and expand your cloud server. If you do not have a server yet you will need to use the Order button to purchase credits. Please note that credits activation can take several days if this is your first order.

Click on the Infrastructure link under your server. In the main area of the manager page you will see the name of your server. Below it, in hard to see grey letters, you will see a 32-digit alphanumeric code such as abcdef0123456789abcdef0123456789. Note it down. This is your *Project ID*.

Click on the Storage link under your server. You will see a list of your containers. If you do not have any containers yet, create a new one. Make sure to select the Private type; you don't want your backups to be publicly accessible! Click on your server's name. The main area changes. You will see a box with information such as objects, container size and Container URL. Note down the *Container URL*.

Click on the OpenStack link under your server. If you have not created an OpenStack user yet, create one now. Copy the values under the ID and Password columns. These are, respectively, your *OpenStack Username* and *OpenStack Password*.

Upload to OVH Object Storage



The required settings for this engine are:

Process each part immediately	If you enable this, each backup part will be uploaded as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the upload fails, the backup fails. If you don't enable this option, the upload process will take place after the backup is complete and finalized. This ensures that if the upload process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.
Delete archive after processing	If enabled, the archive files will be removed from your server after they are uploaded to CloudFiles.
Project ID	See above.
OpenStack Username	See above.
OpenStack Password	See above.
Container URL	See above.
Directory	The directory inside your OVH container where your files will be stored in. If you want to use subdirectories, you have to use a forward slash, e.g. `/directory/subdirectory/subsubdirectory`. Leave blank to store the files on the container's root.

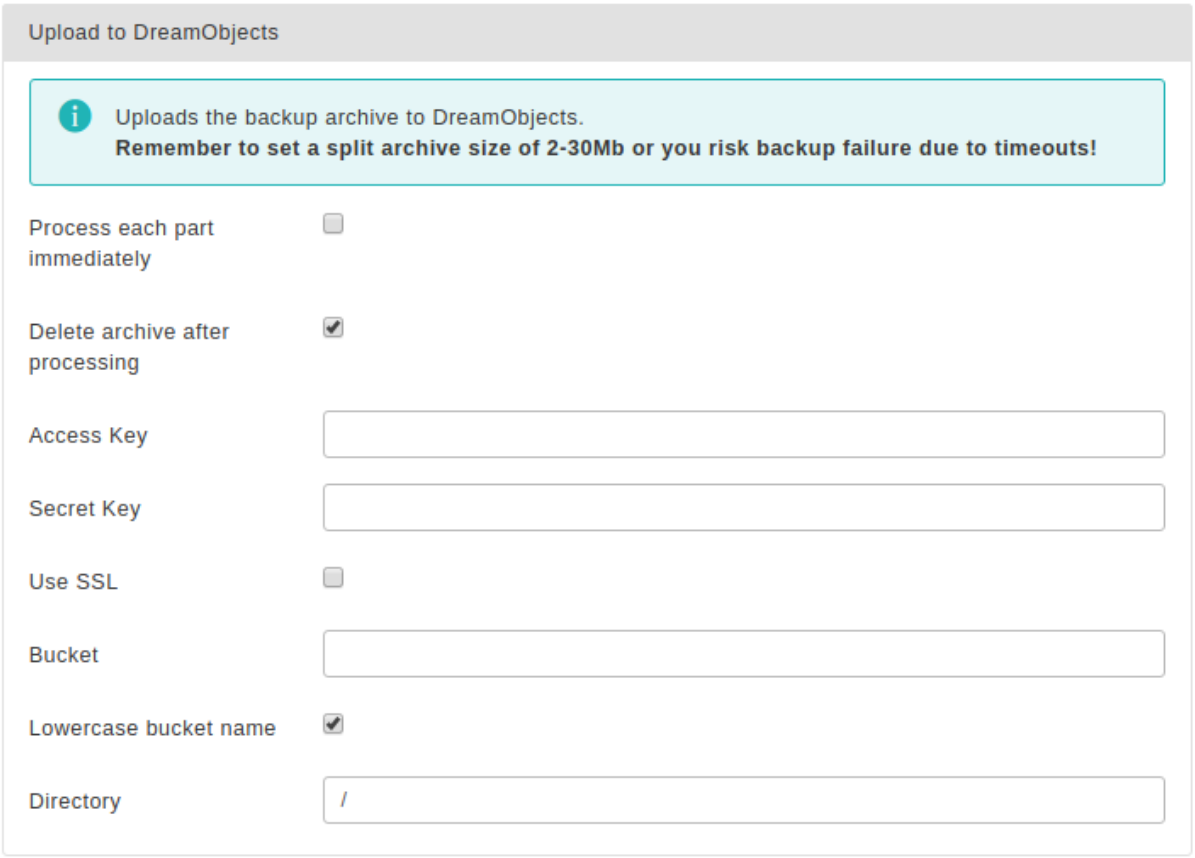
3.3.5.6. Upload to DreamObjects

Note

This feature is available only to Akeeba Backup Professional.

Using this engine, you can upload your backup archives to the DreamObjects cloud storage service by DreamHost.

Upload to DreamObjects



The required settings for this engine are:

Process each part immediately
: If you enable this, each backup part will be uploaded as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the upload fails, the backup fails. If you don't enable this option, the upload process will take place after the backup is complete and finalized. This ensures that if the upload process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.

Delete archive after processing
: If enabled, the archive files will be removed from your server after they are uploaded to DreamObjects.

Access Key
: Your DreamObjects Access Key

Secret Key
: Your DreamObjects Secret Key

Use SSL
: If enabled, an encrypted connection will be used to upload your archives to DreamObjects. In this case the upload will take slightly longer, as encryption - what SSL does - is more resource intensive than uploading unencrypted files. You may have to lower your part size.

Warning

Do not enable this option if your bucket name contains dots.

Bucket
: The name of your DreamObjects bucket where your files will be stored in. The bucket must be already created; Akeeba Backup can not create buckets.

DO NOT CREATE BUCKETS WITH NAMES CONTAINING UPPERCASE LETTERS OR DOTS. If you use a bucket with uppercase letters in its name it is very possible that Akeeba Backup will not be able to upload anything to it for reasons that have to do with the S3 API implemented by DreamObjects. It is not something we can "fix" in Akeeba Backup. Moreover, if you use a dot in your bucket name you will not be able to enable the "Use SSL" option since DreamObject's SSL certificate will be invalid for this bucket, making it impossible to upload backup archives. If this is the case with your site, please don't ask for support; simply create a new bucket whose name only consists of lowercase unaccented latin characters (a-z), numbers (0-9) and dashes.

Directory

The directory inside your DreamObjects bucket where your files will be stored in. If you want to use subdirectories, you have to use a forward slash, e.g. `directory/subdirectory/subsubdirectory`.

You can use Akeeba Backup's "variables" in the directory name in order to create it dynamically. These are the same variables as what you can use in the archive name, i.e. [DATE], [TIME], [HOST], [RANDOM].

Regarding the naming of buckets and directories, you have to be aware of the S3 API rules used by DreamObjects:

- Folder names can not contain backward slashes (\). They are invalid characters.
- Bucket names can only contain lowercase letters, numbers, periods (.) and dashes (-). Accented characters, international characters, underscores and other punctuation marks are illegal characters.

Important

Even if you created a bucket using uppercase letters, **you must type its name with lowercase letters**. The S3 API implemented by DreamObjects automatically converts the bucket name to all-lowercase. Also note that, as stated above, you may NOT be able to use at all under some circumstances. Generally, your should avoid using uppercase letters.

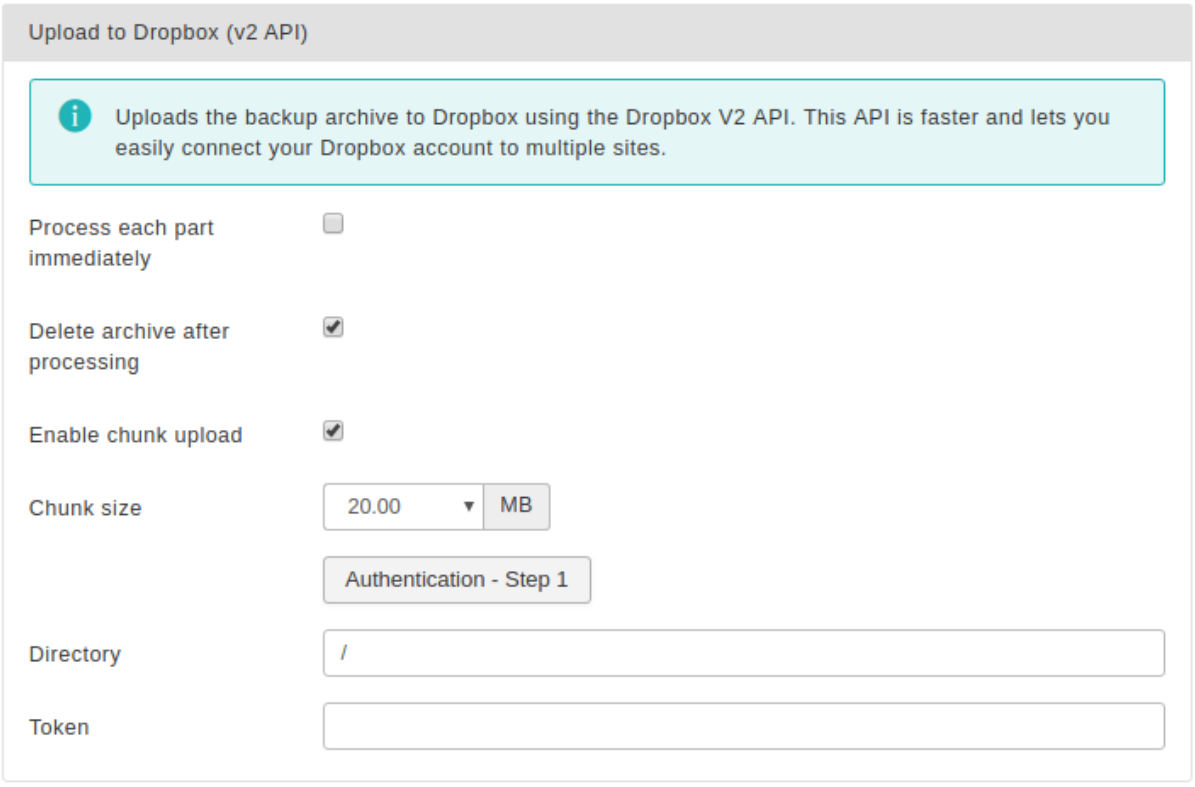
- Bucket names must start with a number or a letter.
- Bucket names must be 3 to 63 characters long.
- Bucket names can't be in an IP format, e.g. 192.168.1.2
- Bucket names can't end with a dash.
- Bucket names can't have an adjacent dot and dash. For example, both `my.-bucket` and `my-.bucket` are invalid. It is preferable to NOT use a dot as it will cause issues.

If any - or all - of those rules are broken, you'll end up with error messages that Akeeba Backup couldn't connect to DreamObjects, that the calculated signature is wrong or that the bucket does not exist. This is normal and expected behaviour, as the S3 API of DreamObjects drops the connection when it encounters invalid bucket or directory names.

3.3.5.7. Upload to Dropbox (v2 API)

Using this engine, you can upload your backup archives to the low-cost Dropbox cloud storage service (http://www.dropbox.com). This is an ideal option for small websites with a low budget, as this service offers 2GB of storage space for free, all the while retaining all the pros of storing your files on the cloud. Even if your host's data center is annihilated by a natural disaster and your local PC and storage media are wiped out by an unlikely event, you will still have a copy of your site readily accessible and easy to restore.

Upload to Dropbox (v2 API)



The required settings for this engine are:

Process each part immediately
: If you enable this, each backup part will be uploaded as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the upload fails, the backup fails. If you don't enable this option, the upload process will take place after the backup is complete and finalized. This ensures that if the upload process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.

Delete archive after processing
: If enabled, the archive files will be removed from your server after they are uploaded to Dropbox.

Authorisation
: Before you can use the application with Dropbox you have to "link" your Dropbox account with your Akeeba Solo / Akeeba Backup installation. This allows the application to access your Dropbox account without you storing the username (email) and password to the application. The authentication is a simple process. First click on the Authentication - Step 1 button. A popup window opens, allowing you to log in to your Dropbox account. Once you log in successfully, click the blue button to transfer the access token back to your Akeeba Solo / Akeeba Backup installation.

  Unlike the v1 API, you can perform the same procedure on every single site you want to link to Dropbox.

Directory
: The directory inside your Dropbox account where your files will be stored in. If you want to use subdirectories, you have to use a forward slash, e.g. `/directory/subdirectory/subsubdirectory`.

Enabled chunked upload
: The application will always try to upload your backup archives / backup archive parts in small chunks and then ask Dropbox to assemble them back into one file. This allows you to transfer larger archives more reliably and works around the 150Mb limitation of Dropbox' API.

When you enable this option every step of the chunked upload process will take place in a separate page load, reducing the risk of timeouts if you are transferring large archive part files (over 10Mb). When you disable this option the entire upload process has to take place in a single page load.

> **Warning**
>
> When you select Process each part immediately this option has no effect! In this case the entire upload operation for each part will be attempted *in a single page load*. For this reason we recommend that you use a Part Size for Split Archives of 5Mb or less to avoid timeouts.

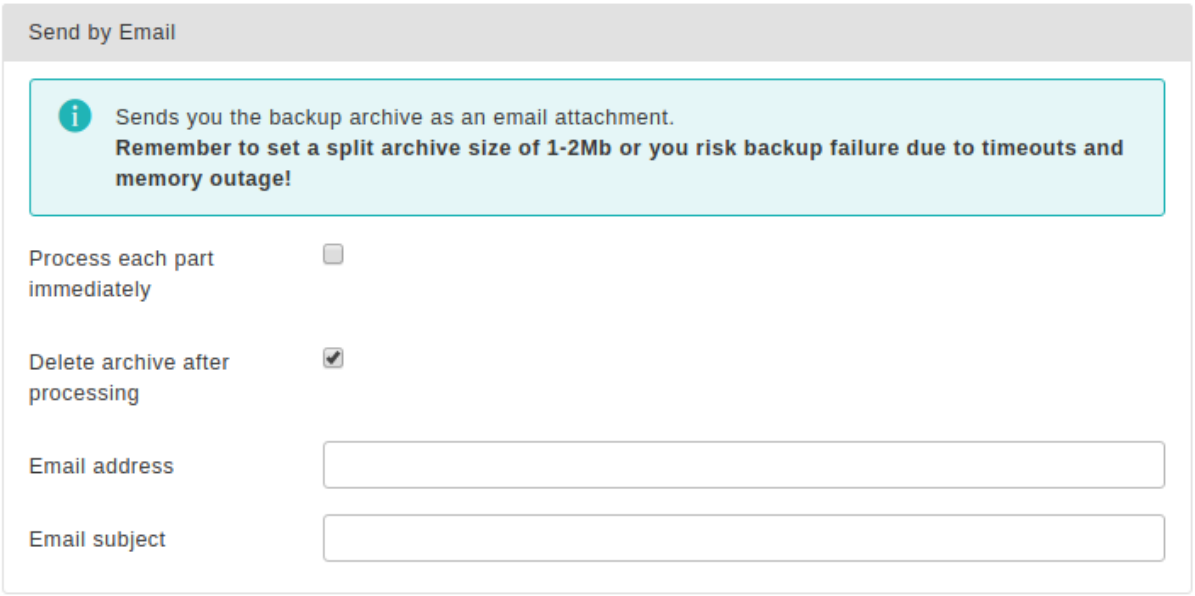
Chunk size — This option determines the size of the chunk which will be used by the chunked upload option above. You are recommended to use a relatively small value around 5 to 20 Mb to prevent backup timeouts. The exact maximum value you can use depends on the speed of your server and its connection speed to the Dropbox server. Try starting high and lower it if the backup fails during transfer to Dropbox.

Token — This is the connection token to Dropbox. Normally, it is automatically fetched from Dropbox when you click on the Authentication - Step 1 button above. If for any reason this method does not work for you you can copy the Token from the popup window or another Akeeba Backup / Akeeba Solo installation you have already connected to Dropbox.

3.3.5.8. Send by email

> **Note**
>
> This feature is available only to Akeeba Backup Professional.

Send by email



This handy feature is available only in Akeeba Backup Professional. It will send you the backup archive parts as file attachments to your email address. That's right! No need to worry about downloading your backup archives, they will be emailed to you. That said, beware of the restrictions:

You **MUST** set the Part size for split archives setting of the Archiver engine to a value between 1-10 Megabytes. If you choose a big value (or leave the default value of 0, which means that no split archives will be generated) you run the risks of the process timing out, a memory outage error to occur or, finally, your email servers not being able to cope with the attachment size, dropping the email.

As a result, this is only suitable for really small sites.

The available configuration settings for this engine, accessed by pressing the Configure... button next to it, are:

Process each part immediately	If you enable this, each backup part will be emailed to you as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the email fails, the backup fails. If you don't enable this option, the email process will take place after the backup is complete and finalized. This ensures that if the email process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.
Delete archive after processing	If enabled, the archive files will be removed from your server after they are emailed to you. Very useful to conserve disk space and practice the good security measure of not leaving your backups on your server.
Email address	The email address where you want your backups sent to. When used with GMail or other webmail services it can provide a cheap alternative to proper cloud storage.
Email subject	A subject for the email you'll receive. You can leave it blank if you want to use the default. However, we suggest using something descriptive, i.e. your site's name and the description of the backup profile.

3.3.5.9. Upload to OneDrive

Note

This feature is available only to Akeeba Backup Professional.

Using this engine, you can upload your backup archives to the low-cost Microsoft Live OneDrive cloud storage service (https://onedrive.live.com). This is an ideal option for small websites with a low budget, as this service offers 15Gb of storage space for free, all the while retaining all the pros of storing your files on the cloud. Even if your host's data center is annihilated by a natural disaster and your local PC and storage media are wiped out by an unlikely event, you will still have a copy of your site readily accessible and easy to restore. Do note that if you are a subscriber to Office 365 you get up to 1Tb of storage in OneDrive.

This feature does NOT support the unrelated, but confusingly similarly named, OneDrive for Business product by Microsoft which you typically get access to as part of an organization-level Microsoft Office 365 *for Business* subscription. Please note that the regular (not "for Business") Microsoft Office 365 subscription gives you access to the regular OneDrive product which is compatible with our software as explained above.

If you have a OneDrive for Business account it is not supported and probably never will. There are two serious reasons for that. The first is that their API documentation is wrong and contradictory. They are aware since at least 2016 and they have not fixed it. The second and most important reason is that they apply an unrealistic rate-limiting. This makes it impossible to use reliably for uploading and retrieving backups, or even applying remote quota settings. We strongly recommend using a reliable storage vendor instead, such as Amazon S3 or BackBlaze B2 - or even Dropbox or Google Drive for Teams. All of the above have been tested and are working far better than Microsoft's offerings.

Important security and privacy information

OneDrive uses the OAuth 2 authentication method. This requires a fixed endpoint (URL) for each application which uses it, such as Akeeba Backup. Since Akeeba Backup is installed on your site, therefore has a different endpoint URL for each installation, you could not normally use OneDrive's API to upload files. We have solved it by creating a small intermediary script which lives on our own server and acts as an intermediary between your site and OneDrive. When you are linking Akeeba Backup to OneDrive you are going through the script on our site. Moreover, whenever the request token (a time-limited key given by OneDrive to your Akeeba Backup installation to access the service) expires your Akeeba Backup installation has to exchange it with a new token. This process also takes place through the script on our site. Please note that even though you are going through our site we DO NOT store this information and we DO NOT have access to your OneDrive account.

WE DO NOT STORE THE ACCESS CREDENTIALS TO YOUR ONEDRIVE ACCOUNT. WE DO NOT HAVE ACCESS TO YOUR ONEDRIVE ACCOUNT. SINCE CONNECTIONS TO OUR SITE ARE PROTECTED BY STRONG ENRYPTION (HTTPS) NOBODY ELSE CAN SEE THE INFORMATION EXCHANGED BETWEEN YOUR SITE AND OUR SITE AND BETWEEN OUR SITE AND ONEDRIVE. HOWEVER, AT THE FINAL STEP OF THE AUTHENTICATION PROCESS, YOUR BROWSER IS SENDING THE ACCESS TOKENS TO YOUR SITE. SOMEONE CAN STEAL THEM IN TRANSIT IF AND ONLY IF YOU ARE NOT USING HTTPS ON YOUR SITE'S ADMINISTRATOR.

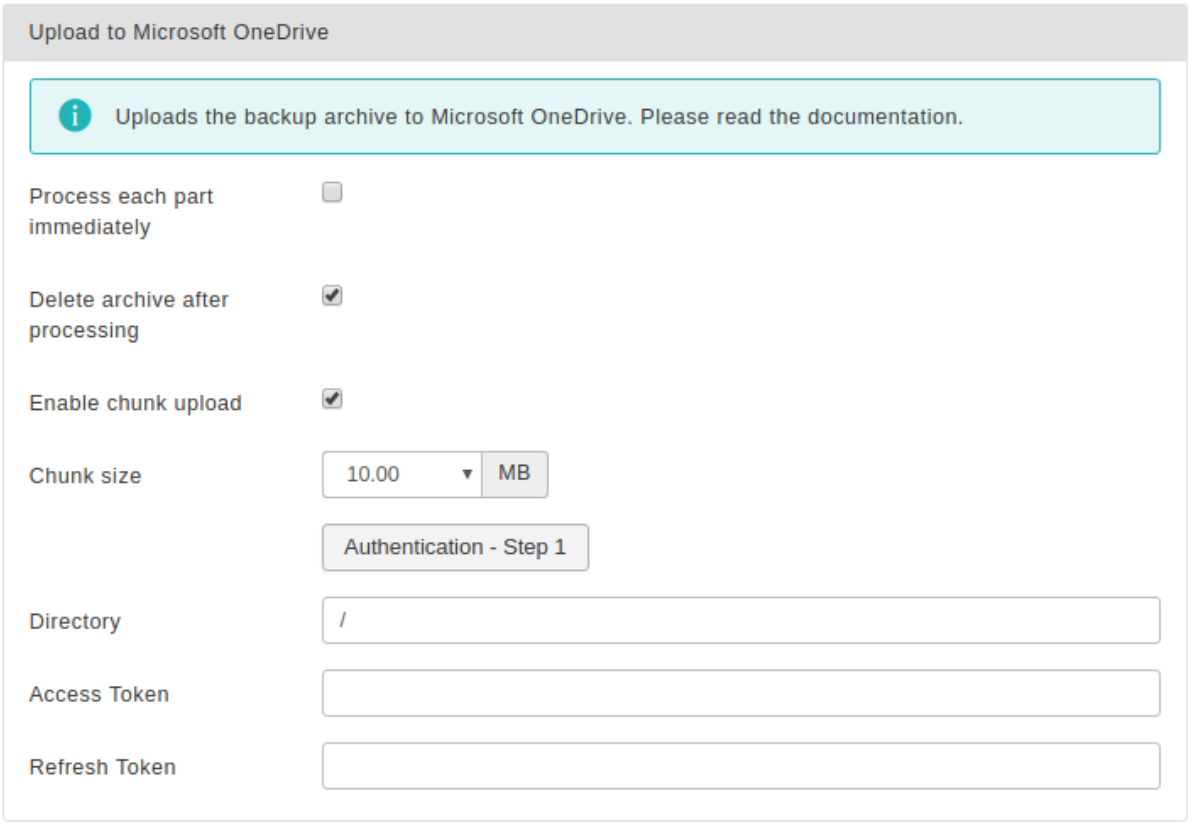
For this reason we DO NOT accept any responsibility whatsoever for any use, abuse or misuse of your connection information to OneDrive. If you do not accept this condition you are FORBIDDEN from using the intermediary script on our site which, simply put, means that you cannot use the OneDrive integration.

Moreover, the above means that there are additional requirements for using OneDrive integration on your Akeeba Backup installation:

- You need the PHP cURL extension to be loaded and enabled on your server. Most servers do that by default. If your server doesn't have it enabled the upload will fail and warn you that cURL is not enabled.
- Your server's firewall must allow outbound HTTPS connections to www.akeebabackup.com over port 443 (standard HTTPS port) to get new tokens every time the current access token expires.
- Your server's firewall must allow outbound HTTPS connections to OneDrive's domains over port 443 to allow the integration to work. These domain names are, unfortunately, not predefined. Most likely your server administrator will have to allow outbound HTTPS connections to any domain name to allow this integration to work. This is a restriction of how the OneDrive service is designed, not something we can modify (obviously, we're not Microsoft).

Settings

Upload to OneDrive



The required settings for this engine are:

Process each part immediately
: If you enable this, each backup part will be uploaded as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the upload fails, the backup fails. If you don't enable this option, the upload process will take place after the backup is complete and finalized. This ensures that if the upload process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.

Delete archive after processing
: If enabled, the archive files will be removed from your server after they are uploaded to OneDrive.

Authorisation – Step 1
: Before you can use Akeeba Backup with OneDrive you have to "link" your OneDrive account with your Akeeba Backup installation. This allows Akeeba Backup to access your OneDrive account without you storing the username (email) and password to. The authentication is a simple process. First click on the Authentication - Step 1 button. A popup window opens, allowing you to log in to your OneDrive account. Once you log in successfully, you are shown a page with the access and refresh tokens (the "keys" returned by OneDrive to be used for connecting to the service) and the URL to your site. Double check that the URL to your site is correct and click on the big blue "Finalize authentication" button. The popup window closes automatically.

  Alternatively, instead of clicking that big blue button you can copy the Access Token and Refresh Token from the popup window to Akeeba Backup's configuration page at the same-named fields. Afterwards you can close the popup.

  ## Important

  As described above, this process routes you through our own site (akeebabackup.com) due to OneDrive's API restrictions. We do NOT store your login information or tokens and we do NOT have access to your OneDrive account. If, however, you do not agree being routed through our site you are FORBIDDEN from using this intermediary service on our site and you cannot use the OneDrive integration feature. We repeat for a third time that this is a restriction imposed by the OneDrive API, not us. We CANNOT work around this restriction, so we created a very secure solution which works within the restrictions imposed by the OneDrive API.

Directory
: The directory inside your OneDrive account where your files will be stored in. If you want to use subdirectories, you have to use a forward slash, e.g. `/directory/subdirectory/subsubdirectory`.

Enabled chunked upload
: When enabled Akeeba Backup will try to upload your backup archives / backup archive parts in small chunks and then ask OneDrive to assemble them back into one file. If your backup archive parts are over 10Mb you are strongly encouraged to check this option.

Chunk size
: This option determines the size of the chunk which will be used by the chunked upload option above. We recommend a relatively small value around 4 to 20 Mb to prevent backup timeouts. The exact maximum value you can use depends on the speed of your server and its connection speed to OneDrive's server. Try starting high and lower it if the backup fails during transfer to OneDrive. You cannot set a chunk size lower than 1Mb or higher than 60Mb because of OneDrive's API restrictions. We recommend using 4, 10 or 20Mb (tested and found to be properly working).

Access Token
: This is the connection token to OneDrive. Normally, it is automatically sent to your site when clicking the blue button from the Authentication Step 1 popup described above. If you do not wish to click that button copy the (very, VERY long!) Access Token from that popup window into this box.

Unlike other engines, such as Dropbox, you CANNOT share OneDrive tokens between multiple site. Each site MUST go through the authentication process described above and use a different set of Access and Refresh tokens!

Refresh Token

This is the refresh token to OneDrive, used to get a fresh Access Token when the previous one expires. Normally, it is automatically sent to your site when clicking the blue button from the Authentication Step 1 popup described above. If you do not wish to click that button copy the (very, VERY long!) Refresh Token from that popup window into this box.

Unlike other engines, such as Dropbox, you CANNOT share OneDrive tokens between multiple site. Each site MUST go through the authentication process described above and use a different set of Access and Refresh tokens!

3.3.5.10. Upload to Remote FTP server

Note

This feature is available only to Akeeba Backup Professional.

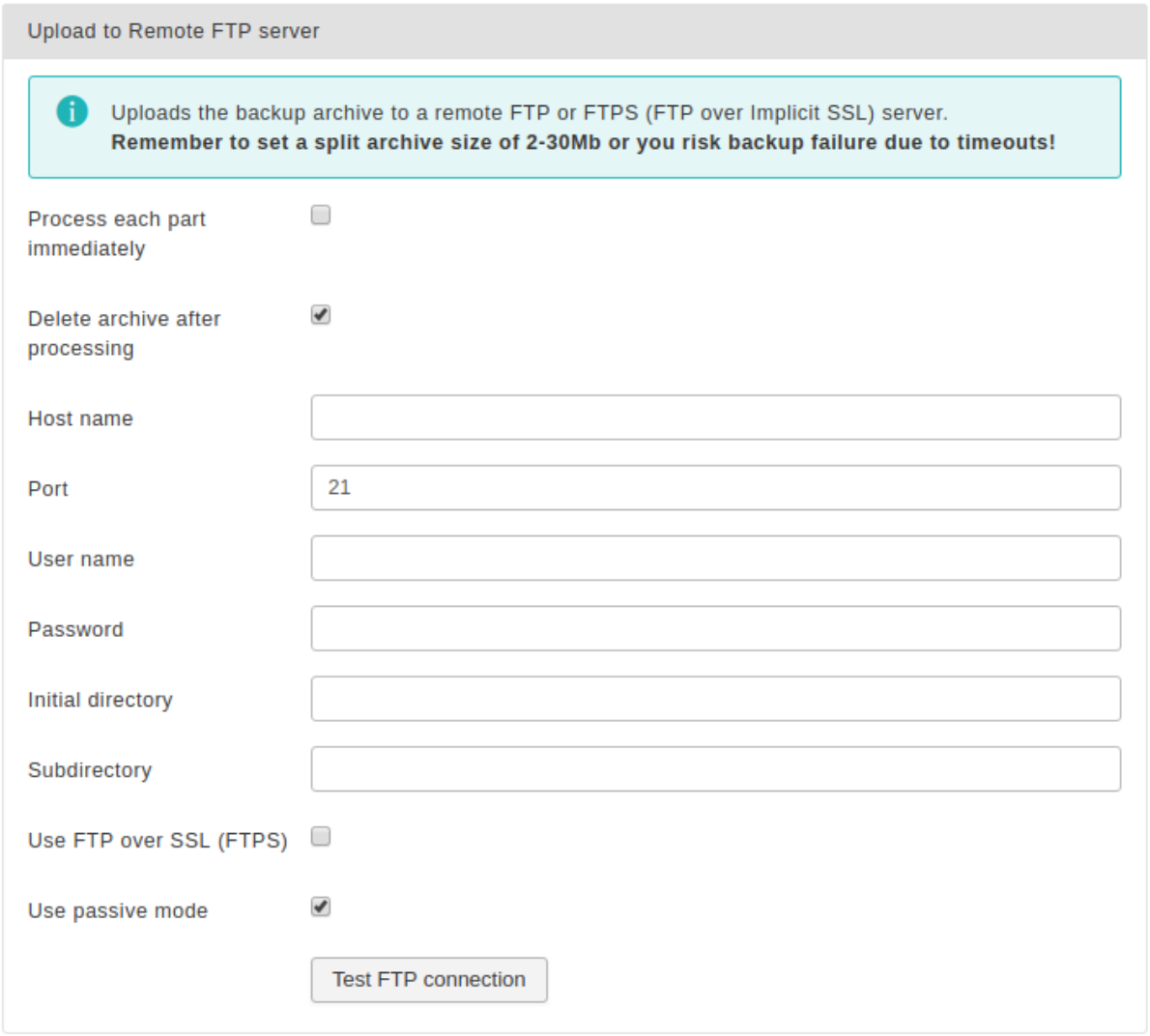
Using this engine, you can upload your backup archives to any FTP or FTPS (FTP over Implicit SSL) server. There are some "FTP" protocols and other file storage protocols which are not supported, such as SFTP, SCP, Secure FTP, FTP over Explicit SSL and SSH variants. The difference of this engine to the DirectFTP archiver engine is that this engine uploads backup archives to the server, whereas DirectFTP uploads the uncompressed files of your site. DirectFTP is designed for rapid migration, this engine is designed for easy moving of your backup archives to an off-server location.

This engine uses PHP's native FTP functions. This may not work if your host has disabled PHP's native FTP functions or if your remote FTP server is incompatible with them. In this case you may want to use the Upload to Remote FTP server over cURL engine instead.

Your originating server must support PHP's FTP extensions and not have its FTP functions blocked. Your originating server must not block FTP communication to the remote (target) server. Some hosts apply a firewall policy which requires you to specify to which hosts your server can connect. In such a case you might need to allow communication to your remote host.

Before you begin, you should know the limitations. Most servers do not allow resuming of uploads (or even if they do, PHP doesn't quite support this feature), so the archive has to be transferred in a single step. PHP has a time limit restriction we can't overlook. The time required to upload a file to FTP equals the size of the file divided by the available bandwidth. We want to time to upload a file to be less than PHP's time limit restriction so as to avoid timing out. Since the available bandwidth is finite and constant, the only thing we can reduce in order to avoid timeouts is the file size. To this end, you have to produce split archives, by setting the part size for archive splitting in ZIP's or JPA's engine configuration pane. The suggested values are between 10 and 20 MB.

Upload to Remote FTP Server



The available configuration options are:

Process each part immediately
: If you enable this, each backup part will be uploaded as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the upload fails, the backup fails. If you don't enable this option, the upload process will take place after the backup is complete and finalized. This ensures that if the upload process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.

Delete archive after processing
: If enabled, the archive files will be removed from your server after they are uploaded to the FTP server.

Host name
: The hostname of your remote (target) server, e.g. `ftp.example.com`. You must NOT enter the ftp:// protocol prefix. If you do, Akeeba Backup will try to remove it automatically and issue a warning about it.

Port
: The TCP/IP port of your remote host's FTP server. It's usually 21.

User name
: The username you have to use to connect to the remote FTP server.

Password	The password you have to use to connect to the remote FTP server.
Initial directory	The absolute FTP directory to your remote site's location where your archives will be stored. This is provided by your hosting company. Do not ask us to tell you what you should put in here because we can't possibly know. There is an easy way to find it, though. Connect to your target FTP server with FileZilla. Navigate to the intended directory. Above the right-hand folder pane you will see a text box with a path. Copy this path and paste it to Akeeba Backup's setting.
Use FTP over SSL	If your remote server supports secure FTP connections over SSL (they have to be Implicit SSL; explicit SSL is not supported), you can enable this feature. In such a case you will most probably *have* to change the port. Please ask your hosting company to provide you with more information on whether they support this feature and what port you should use. You must note that this feature must also be supported by your originating server as well.
Use passive mode	Normally you should enable it, as it is the most common and firewall-safe transfer mode supported by FTP servers. Sometimes, you remote server might require active FTP transfers. In such a case please disable this, but bear in mind that your originating server might not support active FTP transfers, which usually requires tweaking the firewall!

3.3.5.11. Upload to Remote FTP server over cURL

Note

This feature is available only to Akeeba Backup Professional.

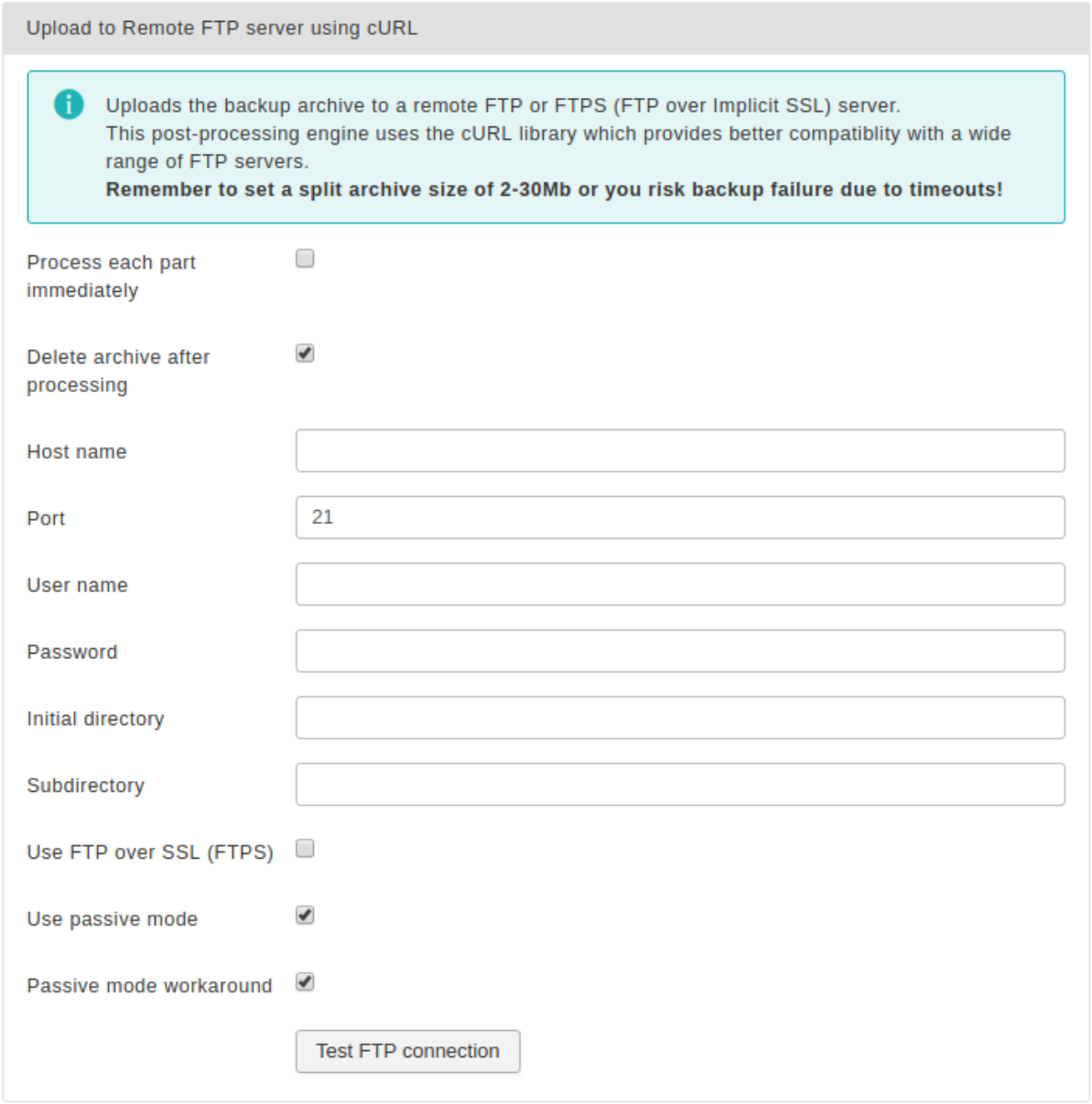
Using this engine, you can upload your backup archives to any FTP or FTPS (FTP over Implicit SSL) server. There are some "FTP" protocols and other file storage protocols which are not supported, such as SFTP, SCP, Secure FTP, FTP over Explicit SSL and SSH variants. The difference of this engine to the DirectFTP over cURL archiver engine is that this engine uploads backup archives to the server, whereas DirectFTP over cURL uploads the uncompressed files of your site. DirectFTP over cURL is designed for rapid migration, this engine is designed for easy moving of your backup archives to an off-server location.

This engine uses PHP's cURL functions. This may not work if your host has not installed or enabled the cURL functions. In this case you may want to use the Upload to Remote FTP server engine instead.

Your originating server must support PHP's cURL extension and not have its FTP functions blocked. Your originating server must not block FTP communication to the remote (target) server. Some hosts apply a firewall policy which requires you to specify to which hosts your server can connect. In such a case you might need to allow communication to your remote host.

Before you begin, you should know the limitations. Most servers do not allow resuming of uploads (or even if they do, PHP doesn't quite support this feature), so the archive has to be transferred in a single step. PHP has a time limit restriction we can't overlook. The time required to upload a file to FTP equals the size of the file divided by the available bandwidth. We want to time to upload a file to be less than PHP's time limit restriction so as to avoid timing out. Since the available bandwidth is finite and constant, the only thing we can reduce in order to avoid timeouts is the file size. To this end, you have to produce split archives, by setting the part size for archive splitting in ZIP's or JPA's engine configuration pane. The suggested values are between 10 and 20 MB.

Upload to Remote FTP Server over cURL



The available configuration options are:

Process each part immediately
: If you enable this, each backup part will be uploaded as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the upload fails, the backup fails. If you don't enable this option, the upload process will take place after the backup is complete and finalized. This ensures that if the upload process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.

Delete archive after processing
: If enabled, the archive files will be removed from your server after they are uploaded to the FTP server.

Host name
: The hostname of your remote (target) server, e.g. `ftp.example.com`. You must NOT enter the ftp:// protocol prefix. If you do, Akeeba Backup will try to remove it automatically and issue a warning about it.

Port	The TCP/IP port of your remote host's FTP server. It's usually 21.
User name	The username you have to use to connect to the remote FTP server.
Password	The password you have to use to connect to the remote FTP server.
Initial directory	The absolute FTP directory to your remote site's location where your archives will be stored. This is provided by your hosting company. Do not ask us to tell you what you should put in here because we can't possibly know. There is an easy way to find it, though. Connect to your target FTP server with FileZilla. Navigate to the intended directory. Above the right-hand folder pane you will see a text box with a path. Copy this path and paste it to Akeeba Backup's setting.
Use FTP over SSL	If your remote server supports secure FTP connections over SSL (they have to be Implicit SSL; explicit SSL is not supported), you can enable this feature. In such a case you will most probably *have* to change the port. Please ask your hosting company to provide you with more information on whether they support this feature and what port you should use. You must note that this feature must also be supported by your originating server as well.
Use passive mode	Normally you should enable it, as it is the most common and firewall-safe transfer mode supported by FTP servers. Sometimes, you remote server might require active FTP transfers. In such a case please disable this, but bear in mind that your originating server might not support active FTP transfers, which usually requires tweaking the firewall!
Passive mode workaround	Some badly configured / misbehaving servers report the wrong IP address when you enable the passive mode. Usually they report their internal network IP address (something like 127.0.0.1 or 192.168.1.123) instead of their public, Internet-accessible IP address. This erroneous information confuses FTP information, causing uploads to stall and eventually fail. Enabling this workaround option instructs cURL to ignore the IP address reported by the server and instead use the FTP server's public IP address, as seen by your web server. In most cases this works much better, therefore we recommend leaving this option turned on if you're not sure. You should only disable it in case of an exotic setup where the FTP server uses two different public IP addresses for the control and data channels.

3.3.5.12. Upload to Google Storage (Legacy S3 API)

Note

This feature is available only to Akeeba Backup Professional.

This is an old implementation which might stop working when Google drops support for the S3 API. We recommend using the more modern JSON API integration described later in the documentation.

Using this engine, you can upload your backup archives to the Google Storage cloud storage service using the interoperable API (Google Storage simulates the API of Amazon S3)

Please note that Google Storage is NOT the same thing as Google Drive. These are two separate products. If you want to upload files to Google Drive please look at the documentation for Upload to Google Drive.

Before you begin you have to go to the Google Developer's Console. After creating a storage bucket, in the left hand menu, go to Storage, Cloud Storage, Settings. Then go to the tab/option Interoperability. There you can go and enable interoperability and create the Access and Secret keys you need for Akeeba Backup.

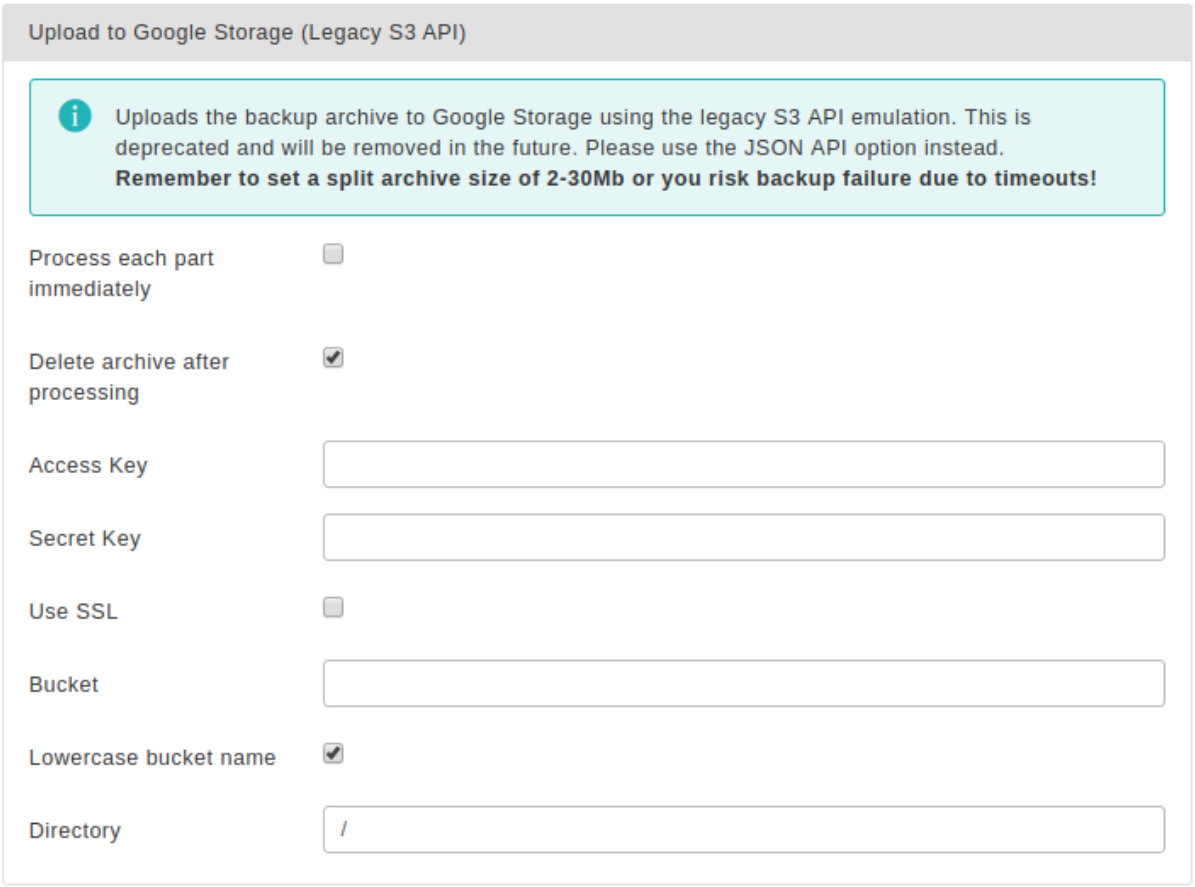
You should also know the limitations. Google Storage's interoperable API does not allow appending to files, so the archive has to be transferred in a single step. PHP has a time limit restriction we can't overlook. The time required to upload a file to Google Storage equals the size of the file divided by the available bandwidth. We want to time to upload a file to be less than PHP's time limit restriction so as to avoid timing out. Since the available bandwidth is finite and constant, the only thing we can reduce in order to avoid timeouts is the file size. To this end, you have

to produce split archives, by setting the part size for archive splitting in ZIP's or JPA's engine configuration pane. The suggested values are between 10 and 20 MB.

Tip

If you use the native CRON mode (akeeba-backup.php), there is usually no time limit - or there is a very high time limit in the area of 3 minutes or so. Ask your host about it. Setting up a profile for use only with the native CRON mode allows you to increase the part size and reduce the number of parts a complete backup consists of.

Upload to Google Storage (Legacy S3 API)



The required settings for this engine are:

Process each part immediately
: If you enable this, each backup part will be uploaded as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the upload fails, the backup fails. If you don't enable this option, the upload process will take place after the backup is complete and finalized. This ensures that if the upload process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.

Delete archive after processing
: If enabled, the archive files will be removed from your server after they are uploaded to Google Storage.

Access Key
: Your Google Storage Access Key, available from the Google Cloud Storage key management tool [https://code.google.com/apis/console#:storage:legacy].

Secret Key
: Your Google Storage Secret Key, available from the Google Cloud Storage key management tool [https://code.google.com/apis/console#:storage:legacy].

Use SSL
: If enabled, an encrypted connection will be used to upload your archives to Google Storage. In this case the upload will take longer, as encryption - what SSL does - is a resource intensive operation. You may have to lower your part size. We strongly recommend enabling this option for enhanced security.

> **Warning**
>
> Do not enable this option if your bucket name contains dots.

Bucket
: The name of your Google Storage bucket where your files will be stored in. The bucket must be already created; Akeeba Backup can not create buckets.

  DO NOT CREATE BUCKETS WITH NAMES CONTAINING UPPERCASE LETTERS. If you use a bucket with uppercase letters in its name it is very possible that Akeeba Backup will not be able to upload anything to it. Moreover you should not use dots in your bucket names as they are incompatible with the Use SSL option due to an Amazon S3 API limitation.

  Please note that this is a limitation of the API. It is not something we can "fix" in Akeeba Backup. If this is the case with your site, please **DO NOT** ask for support; simply create a new bucket whose name only consists of lowercase unaccented latin characters (a-z), numbers (0-9) and dashes.

Directory
: The directory inside your Google Storage bucket where your files will be stored in. If you want to use subdirectories, you have to use a forward slash, e.g. `directory/subdirectory/subsubdirectory`.

> **Tip**
>
> You can use Akeeba Backup's "variables" in the directory name in order to create it dynamically. These are the same variables as what you can use in the archive name, i.e. [DATE], [TIME], [HOST], [RANDOM].

Regarding the naming of buckets and directories, you have to be aware of the Google Storage rules:

- Folder names can not contain backward slashes (\). They are invalid characters.
- Bucket names can only contain lowercase letters, numbers, periods (.) and dashes (-). Accented characters, international characters, underscores and other punctuation marks are illegal characters.
- Bucket names must start with a number or a letter.
- Bucket names must be 3 to 63 characters long.
- Bucket names can't be in an IP format, e.g. 192.168.1.2
- Bucket names can't end with a dash.
- Bucket names can't have an adjacent dot and dash. For example, both `my.-bucket` and `my-.bucket` are invalid. It's best not to use dots at all as they are incompatible with the Use SSL option.

If any - or all - of those rules are broken, you'll end up with error messages that Akeeba Backup couldn't connect to Google Storage, that the calculated signature is wrong or that the bucket does not exist. This is normal and expected behaviour, as Google Storage drops the connection when it encounters invalid bucket or directory names.

3.3.5.13. Upload to Google Storage (JSON API)

Using this engine, you can upload your backup archives to the Google Storage cloud storage service using the official Google Cloud JSON API. This is the preferred method for using Google Storage.

Foreward and requirements

Setting up Google Storage is admittedly complicated. We did ask Google for permission to use the much simpler end-user OAuth2 authentication, a method which is more suitable for people who are not backend developers or IT managers. Unfortunately, their response on July 14th, 2017 was that we were not allowed to. They said in no uncertain terms that we MUST have our clients use Google Cloud Service Accounts. Unfortunately this comes with increased server requirements and more complicated setup instructions.

First the requirements. Google Storage support requires the openssl_sign() function to be available on your server and support the "sha256WithRSAEncryption" method (it must be compiled against the OpenSSL library version 0.9.8l or later). If you are not sure please ask your host. Please note that the versions of the software required for Google Storage integration have been around since early 2012 so they shouldn't be a problem for any decently up-to-date host.

Moreover, we are only allowed to give you the following quick start instructions as an indicative way to set up Google Storage. If you need support for creating a service account or granting Akeeba Backup the appropriate permissions via the IAM Policies, Google requested that we direct you to their Google Cloud Support page [https://cloud.google.com/support/]. We are afraid this means that we will not be able to provide you with support about any issues concerning the Google Cloud side of the setup at the request of Google.

We apologize for any inconvenience. We have no option but to abide by Google's terms. It's their service, their API and their rules.

Initial Setup

Before you begin you will need to create a JSON authorization file for Akeeba Backup / Akeeba Solo. Please follow the instructions below, step by step, to do this. Kindly note that you can reuse the same JSON authorization file on multiple sites and / or backup profiles.

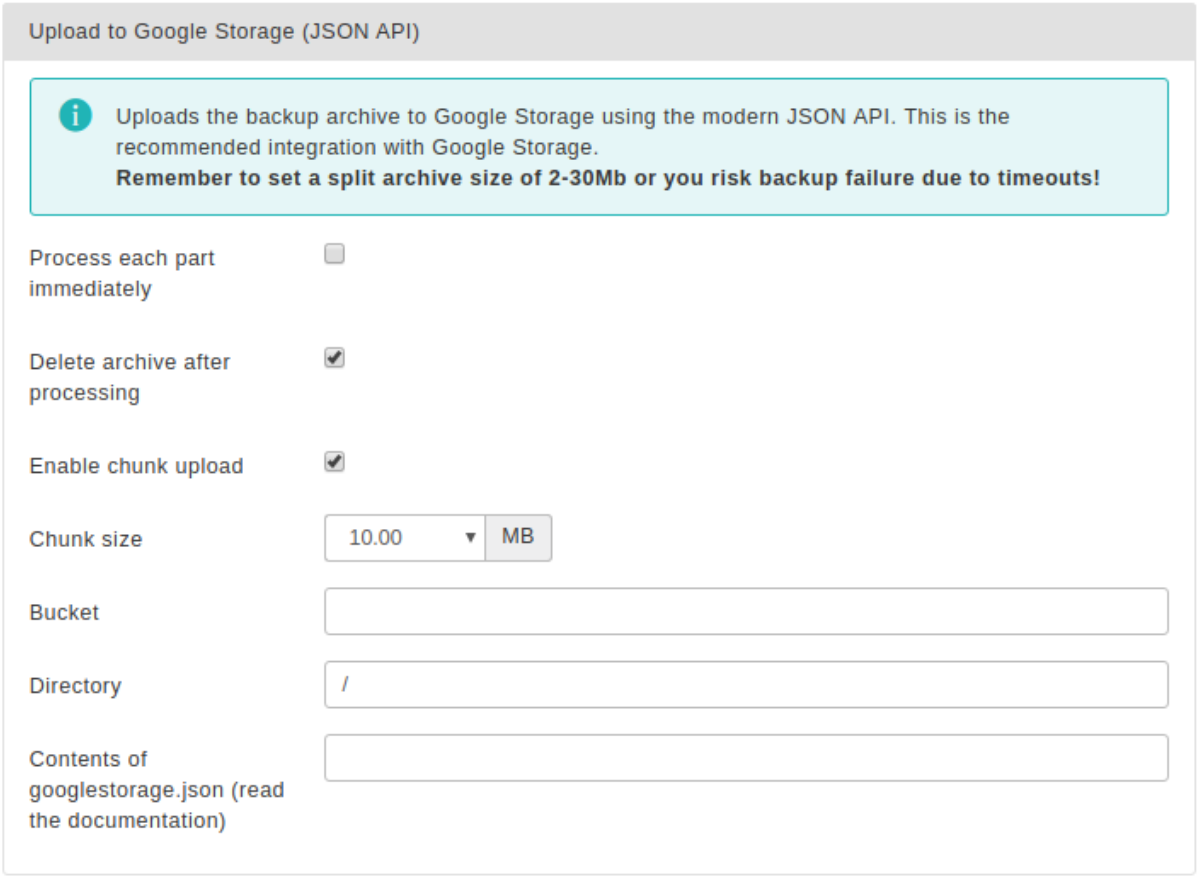
1. Go to https://console.developers.google.com/permissions/serviceaccounts?pli=1
2. Select the API Project where your Google Storage bucket is already located in.
3. Click on Create Service Account
4. Set the Service Account Name to `Akeeba Backup Service Account`
5. Click on Role and select Storage, Storage Object Admin
6. Check the Furnish a new private key checkbox.
7. The Key Type section appears. Make sure `JSON` is selected.
8. Click on the CREATE link at the bottom right.
9. Your server prompts you to download a file. Save it as `googlestorage.json` You will need to paste the contents of this file in the Contents of googlestorage.json (read the documentation) field in the Configuration page of Akeeba Backup / Akeeba Solo.

Important

If you lose the `googlestorage.json` file you will have to delete the Service Account and create it afresh. If you had any sites already set up with this `googlestorage.json` you will need to reconfigure them with the *new* file you created for the *new* Service Account. In short: don't lose that file, you *will* need it to (re)connect your sites with Google Storage.

Post-processing engine options

Upload to Google Storage (JSON API)



The settings for this engine are:

Process each part immediately
: If you enable this, each backup part will be uploaded as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the upload fails, the backup fails. If you don't enable this option, the upload process will take place after the backup is complete and finalized. This ensures that if the upload process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.

Delete archive after processing
: If enabled, the archive files will be removed from your server after they are uploaded to Google Storage.

Enabled chunk upload
: When enabled, Akeeba Backup / Akeeba Solo will upload your backup archives in 5Mb chunks. This is the recommended methods for larger (over 10Mb) archives and/or archive parts.

Bucket
: The name of your Google Storage bucket where your files will be stored in. The bucket must be already created; the application can not create buckets.

  DO NOT CREATE BUCKETS WITH NAMES CONTAINING UPPERCASE LETTERS. If you use a bucket with uppercase letters in its name it is very possible that the application will not be able to upload anything to it.

	Please note that this is a limitation of the API. It is not something we can "fix" in the application. If this is the case with your site, please simply create a new bucket whose name only consists of lowercase unaccented latin characters (a-z), numbers (0-9), dashes and dots.
Directory	The directory inside your Google Storage bucket where your files will be stored in. If you want to use subdirectories, you have to use a forward slash, e.g. `directory/subdirectory/subsubdirectory`. You can use the application's "variables" in the directory name in order to create it dynamically. These are the same variables as what you can use in the archive name, i.e. [DATE], [TIME], [HOST], [RANDOM].
Contents of googlestorage.json (read the documentation)	Open the JSON file you created in the Initial Setup stage outlined above. Copy all of its contents. Paste them in this field. Make sure you have included the curly braces, { and }, at the beginning and end of the file respectively. Don't worry about line breaks being "eaten up", they are NOT important.

Regarding the naming of buckets and directories, you have to be aware of the Google Storage rules:

- Folder names can not contain backward slashes (\). They are invalid characters.
- Bucket names can only contain lowercase letters, numbers, periods (.) and dashes (-). Accented characters, international characters, underscores and other punctuation marks are illegal characters.
- Bucket names must start with a number or a letter.
- Bucket names must be 3 to 63 characters long.
- Bucket names can't be in an IP format, e.g. 192.168.1.2
- Bucket names can't end with a dash.
- Bucket names can't have an adjacent dot and dash. For example, both `my.-bucket` and `my-.bucket` are invalid.

If any - or all - of those rules are broken, you'll end up with error messages that the application couldn't connect to Google Storage, that the calculated signature is wrong or that the bucket does not exist. This is normal and expected behaviour, as Google Storage drops the connection when it encounters invalid bucket or directory names.

3.3.5.14. Upload to Google Drive

Note

This feature is available only to Akeeba Backup Professional.

Using this engine you can upload your backup archives to Google Drive.

Important security and privacy information

Google Drive uses the OAuth 2 authentication method. This requires a fixed endpoint (URL) for each application which uses it, such as Akeeba Backup. Since Akeeba Backup is installed on your site it has a different endpoint URL for each installation, meaning you could not normally use Google Drive's API to upload files. We have solved it by creating a small script which lives on our own server and acts as an intermediary between your site and Google Drive. When you are linking Akeeba Backup to Google Drive you are going through the script on our site. Moreover, whenever the request token (a time-limited key given by Google Drive to your Akeeba Backup installation to access the service) expires your Akeeba Backup installation has to exchange it with a new token. This process also takes place through the script on our site. Please note that even though you are going through our site we DO NOT store this information and we DO NOT have access to your Google Drive account.

WE DO NOT STORE THE ACCESS CREDENTIALS TO YOUR GOOGLE DRIVE ACCOUNT. WE DO NOT HAVE ACCESS TO YOUR GOOGLE DRIVE ACCOUNT. SINCE CONNECTIONS TO OUR SITE ARE PROTECTED BY STRONG ENCRYPTION (HTTPS) NOBODY ELSE CAN SEE THE INFORMATION EXCHANGED BETWEEN YOUR SITE AND OUR SITE AND BETWEEN OUR SITE AND GOOGLE DRIVE. HOWEVER, AT THE FINAL STEP OF THE AUTHENTICATION PROCESS, YOUR BROWSER IS SENDING THE ACCESS TOKENS TO YOUR SITE. SOMEONE CAN STEAL THEM IN TRANSIT IF AND ONLY IF YOU ARE NOT USING HTTPS ON YOUR SITE'S ADMINISTRATOR.

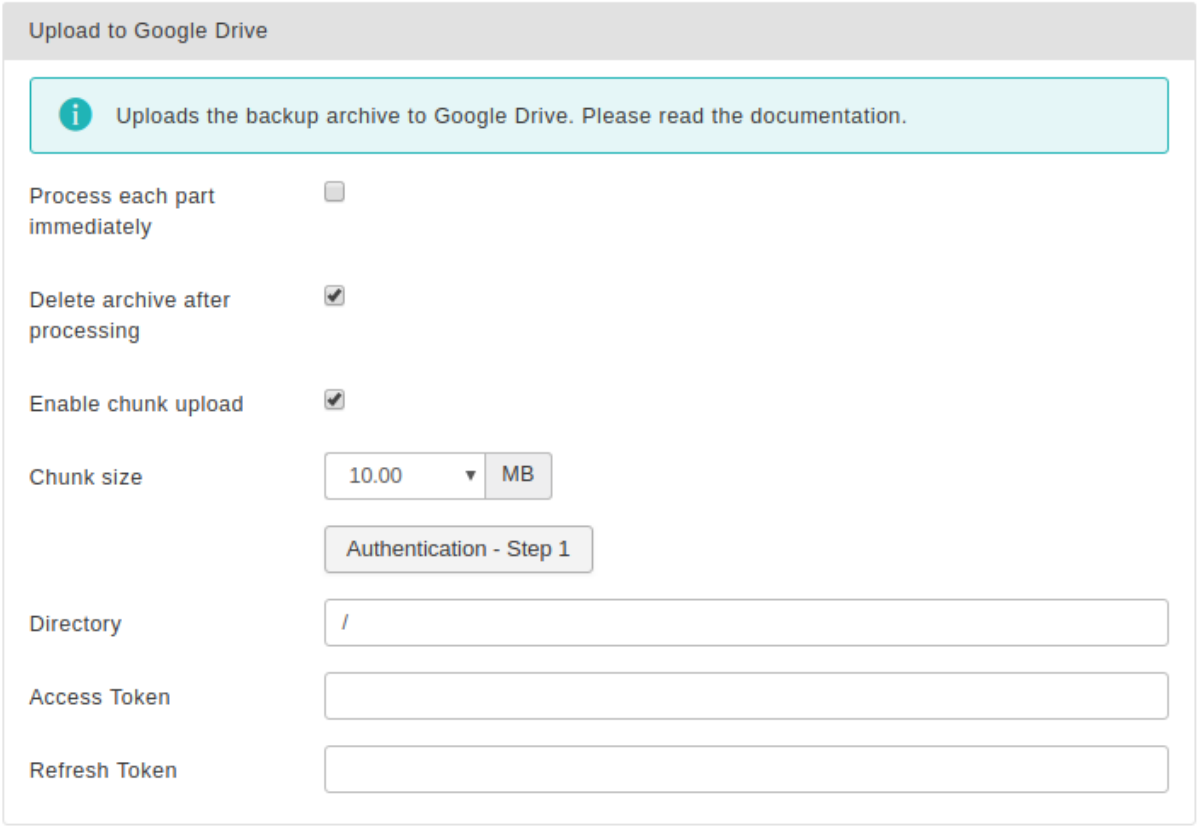
For this reason we DO NOT accept any responsibility whatsoever for any use, abuse or misuse of your connection information to Google Drive. If you do not accept this condition you are FORBIDDEN from using the intermediary script on our site which, simply put, means that you cannot use the Google Drive integration.

Moreover, the above means that there are additional requirements for using Google Drive integration on your Akeeba Backup installation:

- You need the PHP cURL extension to be loaded and enabled on your server. Most servers do that by default. If your server doesn't have it enabled the upload will fail and warn you that cURL is not enabled.
- Your server's firewall must allow outbound HTTPS connections to www.akeebabackup.com over port 443 (standard HTTPS port) to get new tokens every time the current access token expires.
- Your server's firewall must allow outbound HTTPS connections to Google Drive's domains over port 443 to allow the integration to work. These domain names are, unfortunately, not predefined. Most likely your server administrator will have to allow outbound HTTPS connections to any domain name matching *.googleapis.com to allow this integration to work. This is a restriction of how the Google Drive service is designed, not something we can modify (obviously, we're not Google).

Settings

Upload to Google Drive



The settings for this engine are:

Process each part immediately
: If you enable this, each backup part will be uploaded as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the upload fails, the backup fails. If you don't enable this option, the upload process will take place after the backup is complete and finalized. This ensures that if the upload process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.

Delete archive after processing
: If enabled, the archive files will be removed from your server after they are uploaded to Google Drive

Enabled chunked upload
: The application will always try to upload your backup archives / backup archive parts in small chunks and then ask Google Drive to assemble them back into one file. This allows you to transfer larger archives more reliably.

  When you enable this option every step of the chunked upload process will take place in a separate page load, reducing the risk of timeouts if you are transferring large archive part files (over 5Mb). When you disable this option the entire upload process has to take place in a single page load.

  ## Warning

  When you select Process each part immediately this option has no effect! In this case the entire upload operation for each part will be attempted *in a single page load*. For this reason we recommend that you use a Part Size for Split Archives of 5Mb or less to avoid timeouts.

Chunk size
: This option determines the size of the chunk which will be used by the chunked upload option above. You are recommended to use a relatively small value around 5 to 20 Mb to prevent backup timeouts. The exact maximum value you can use depends on the speed of your server and its connection speed to the Google Drive server. Try starting high and lower it if the backup fails during transfer to Google Drive.

Authentication – Step 1
: If this is the **FIRST** site you connect to Akeeba Backup click on this button and follow the instructions.

  On **EVERY SUBSEQUENT SITE** do NOT click on this button! Instead copy the Refresh Token from the first site into this new site's Refresh Token edit box further below the page.

  ## Warning

  Google imposes a limitation of 20 authorizations for a single application –like Akeeba Backup– with Google Drive. Simply put, every time you click on the Authentication – Step 1 button a new Refresh Token is generated. The 21st time you generate a new Refresh Token the one you had created the very first time becomes automatically invalid without warning. This is how Google Drive is designed to operate. For this reason we strongly recommend AGAINST using this button on subsequent sites. Instead, copy the Refresh Token.

Directory
: The directory inside your Google Drive where your files will be stored in. If you want to use subdirectories, you have to use a forward slash, e.g. `directory/subdirectory/subsubdirectory`.

  ## Tip

  You can use Akeeba Backup's "variables" in the directory name in order to create it dynamically. These are the same variables as what you can use in the archive name, i.e. [DATE], [TIME], [HOST], [RANDOM].

Warning

Object (file and folder) naming in Google Drive is ambiguous by design. This means that two or more files / folders with the same name can exist inside the same folder at the same time. In other words, a folder called My Files may contain ten *different* files all called "File 1"! Obviously this is problematic when you want to store backups which need to be uniquely named (otherwise you'd have no idea which backup is the one you want to use!). We work around this issue using the following conventions:

- If there are multiple folders by the same name we choose the first one returned by the Google Drive API. There are no guarantees which one it will be! **Please do NOT store backup archives in folders with ambiguous names** or the remote file operations (quota management, download to server, download to browser, delete) will most likely fail.
- If a folder in the path you specified does not exist we create it
- If a file by the same name exists in the folder you specified we delete it before uploading the new one.

Access Token
: This is the temporary Access Token generated by Google Drive. It has a lifetime of one hour (3600 seconds). After that Akeeba Backup will use the Refresh Token automatically to generate a new Access Token. Please do not touch that field and do NOT copy it to other sites.

Refresh Token
: This is essentially what connects your Akeeba Backup installation with your Google Drive. When you want to connect more sites to Google Drive please copy the Refresh Token from another site linked to the same Google Drive account to your site's Refresh Token field.

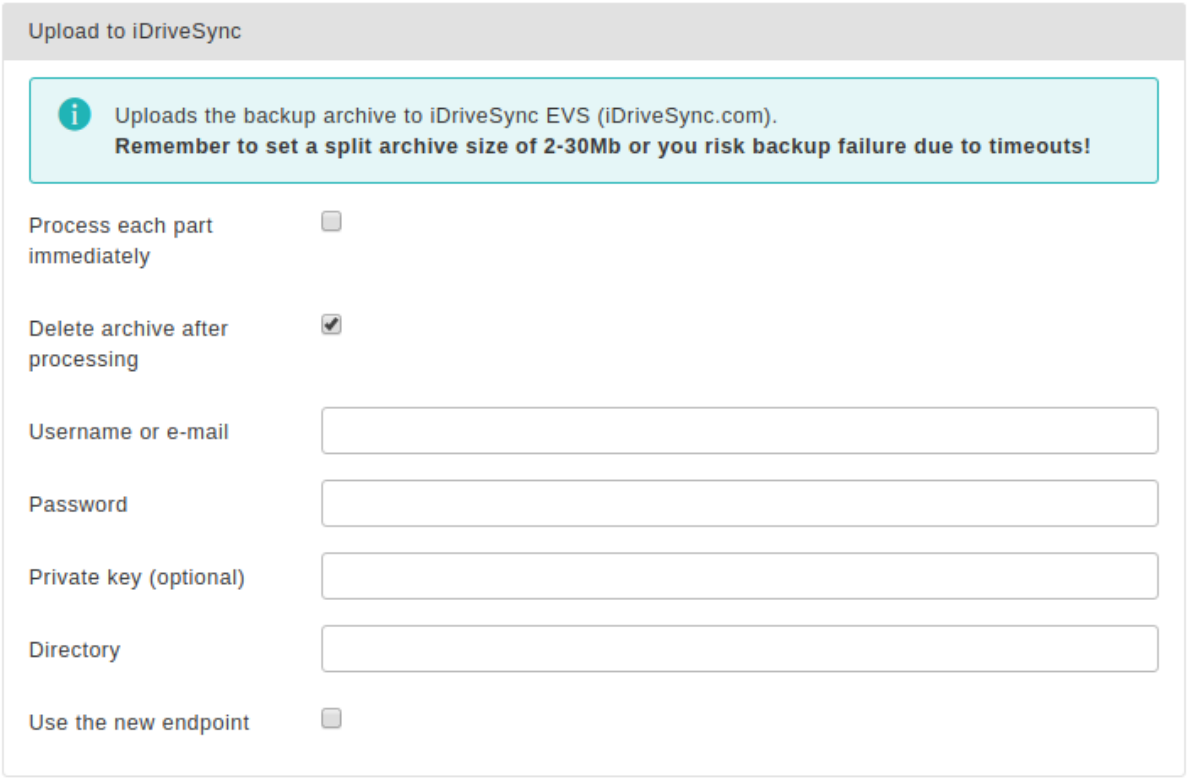
Warning

Since all of your sites are using the same Refresh Token to connect to Google Drive you must NOT run backups on multiple sites simultaneously. That would cause all backups to fail since one active instance of Akeeba Backup would be invalidating the Access Token generated by the other active instance of Akeeba Backup also trying to upload to Google Drive. This is an architectural limitation of Google Drive.

3.3.5.15. Upload to iDriveSync

Using this engine, you can upload your backup archives to the iDriveSync low-cost, encrypted, cloud storage service.

Upload to iDriveSync



The required settings for this engine are:

Process each part immediately	If you enable this, each backup part will be uploaded as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the upload fails, the backup fails. If you don't enable this option, the upload process will take place after the backup is complete and finalized. This ensures that if the upload process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.
Delete archive after processing	If enabled, the archive files will be removed from your server after they are uploaded to iDriveSync
Username or e-mail	Your iDriveSync username or email address
Password	Your iDriveSync password
Private key (optional)	If you have locked your account with a private key (which means that all your data is stored encrypted in iDriveSync) please enter your Private Key here. If you are not making use of this feature please leave this field blank.
Directory	The directory inside your iDriveSync where your files will be stored in. If you want to use subdirectories, you have to use a forward slash, e.g. `directory/subdirectory/subsubdirectory`.

Tip

You can use Akeeba Backup's "variables" in the directory name in order to create it dynamically. These are the same variables as what you can use in the archive name, i.e. [DATE], [TIME], [HOST], [RANDOM].

Use the new endpoint

This is required for iDriveSync accounts created after 2014. If you have entered your username/e-mail and password correctly but Akeeba Backup can't connect to iDriveSync please try checking this box.

Lengthier explanation. Sometime after 2014 iDriveSync started signing up new users through iDrive.com instead of iDriveSync.com. The new accounts need to access a new service endpoint (URL) to upload new files, delete existing files and so on. Meanwhile, accounts created before this change still need to access the old service endpoint (URL). The same service, two different interface implementations, making it impossible for us to automatically detect which method will work with your iDriveSync account. Therefore the only thing we could do was add this confusing checkbox. We're sorry about that.

3.3.5.16. Upload to Amazon S3 (Legacy API)

Note

This feature has been discontinued. If you were using it please upgrade your backup profiles to the Upload to Amazon S3 post-processing engine.

3.3.5.17. Upload to Amazon S3

Note

This feature is available only to Akeeba Backup Professional. Older versions of Akeeba Backup may not have all of the options discussed here.

Using this engine, you can upload your backup archives to the Amazon S3 cloud storage service and other storage services providing an S3-compatible API. With dirt cheap prices per Gigabyte, it is an ideal option for securing your backups. Even if your host's data center is annihilated by a natural disaster and your local PC and storage media are wiped out by an unlikely event, you will still have a copy of your site readily accessible and easy to restore.

This engine supports multi-part uploads to Amazon S3. This means that, unlike the other post-processing engines, even if you do not use split archives, Akeeba Backup will still be able to upload your files to Amazon S3! This new feature allows Akeeba Backup to upload your backup archive in 5Mb chunks so that it doesn't time out when uploading a very big archive file. That said, we STRONGLY suggest using a part size for archive splitting of 2000Mb. This is required to work around a PHP limitation which causes extraction to fail if the file size is over roughly 2Gb.

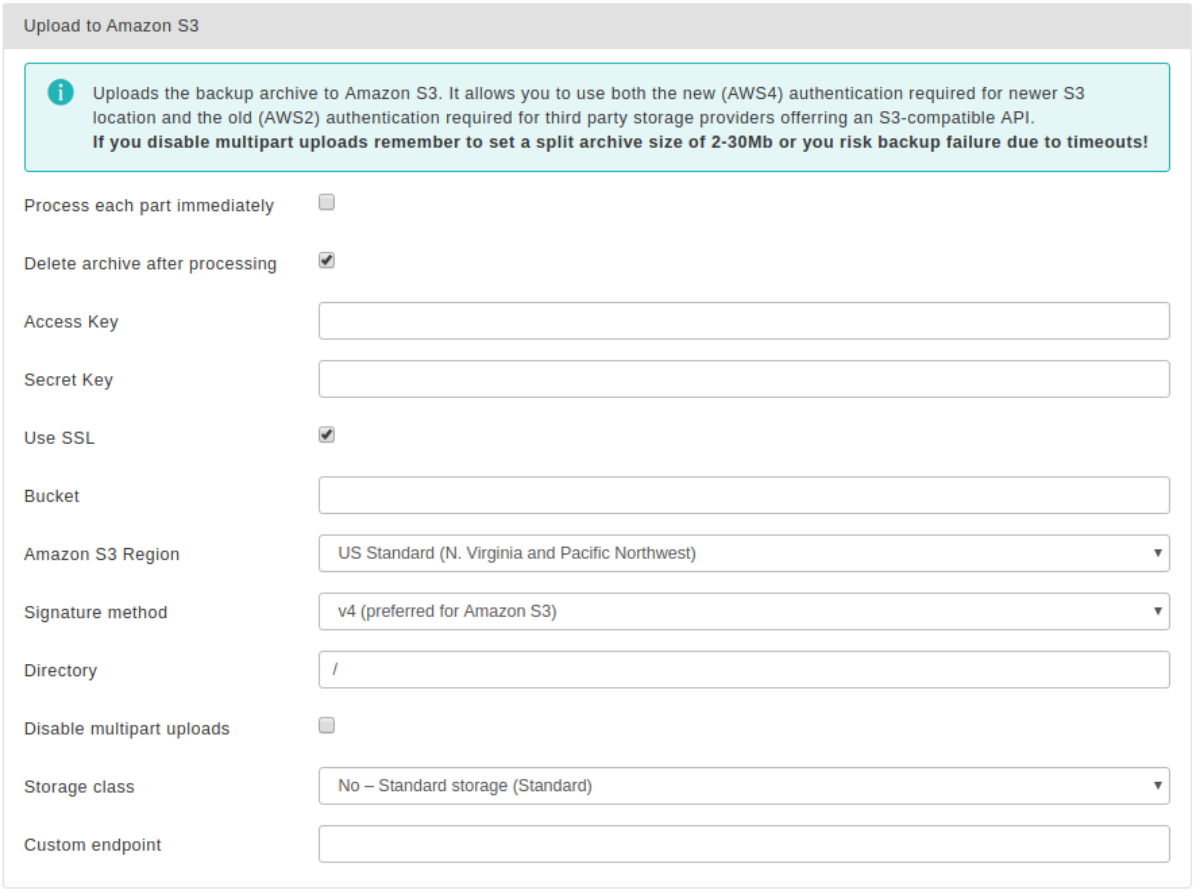
You can also specify a custom endpoint URL. This allows you to use this feature with third party cloud storage services offering an API compatible with Amazon S3 such as Cloudian, Riak CS, Ceph, Connectria, HostEurope, Dunkel, S3For.me, Nimbus, Walrus, GreenQloud, Scality Ring, CloudStack and so on. If a cloud solution (public or private) claims that it is compatible with S3 then you can use it with Akeeba Backup.

Note

Akeeba Backup 5.1.2 and later support the Beijing Amazon S3 region, i.e. storage buckets hosted in China. These buckets are only accessible from inside China and have a few caveats:

- You can only access buckets in the Beijing region from inside China.
- Download to browser is not supported unless you have a license by the Chinese government to share content from your Amazon S3 bucket. That's because downloading to browser requires a pre-signed URL which could, in theory, be used to disseminate material from your Amazon S3 bucket to others. So even though you see the Download button it will most likely result in an error.
- Sometimes deleting and trying to re-upload an object or trying to overwrite fails silently (without an error message). WE strongly recommend using unique names for your backup archives and testing them frequently.

Upload to Amazon S3



The required settings for this engine are:

Process each part immediately
: If you enable this, each backup part will be uploaded as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the upload fails, the backup fails. If you don't enable this option, the upload process will take place after the backup is complete and finalized. This ensures that if the upload process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.

Delete archive after processing
: If enabled, the archive files will be removed from your server after they are uploaded to Amazon S3.

Access Key
: Your Amazon S3 Access Key. Required unless you run Akeeba Backup inside an EC2 instance with an attached IAM Role. Please read about this below.

Secret Key
: Your Amazon S3 Secret Key. Required unless you run Akeeba Backup inside an EC2 instance with an attached IAM Role. Please read about this below.

Use SSL
: If enabled, an encrypted connection will be used to upload your archives to Amazon S3.

Warning

Do not use this option if your bucket name contains dots.

Bucket
: The name of your Amazon S3 bucket where your files will be stored in. The bucket must be already created; Akeeba Backup can not create buckets.

Warning

DO NOT CREATE BUCKETS WITH NAMES CONTAINING UPPERCASE LETTERS. AMAZON CLEARLY WARNS AGAINST DOING THAT. If you use a bucket with uppercase letters in its name it is very possible that Akeeba Backup will not be able to upload anything to it. More specifically, it seems that if your web server is located in Europe, you will be unable to use a bucket with uppercase letters in its name. If your server is in the US, you will most likely be able to use such a bucket. Your mileage may vary. The same applies if your bucket name contains dots and you try using the Use SSL option, for reasons that have to do with Amazon S3's setup.

Please note that this is a limitation imposed by Amazon itself. It is not something we can "fix" in Akeeba Backup. If this is the case with your site, please **DO NOT** ask for support; simply create a new bucket whose name only consists of lowercase unaccented Latin characters (a-z), numbers (0-9) and dashes.

Amazon S3 Region

Please select which S3 Region you have created your bucket in. This is MANDATORY for using the newer, more secure, v4 signature method. You can see the region of your bucket in your Amazon S3 management console. Right click on a bucket and click on Properties. A new pane opens to the left. The second row is labelled Region. This is where your bucket was created in. Go back to Akeeba Backup and select the corresponding option from the drop-down.

Important

If you choose the wrong region the connection WILL fail.

Please note that there are some reserved regions which have not been launched by Amazon at the time we wrote this engine. They are included for forward compatibility should and when Amazon launches those regions.

Signature method

This option determines the authentication API which will be used to "log in" the backup engine to your Amazon S3 bucket. You have two options:

- **v4 (preferred for Amazon S3)**. If you are using Amazon S3 (not a compatible third party storage service) and you are not sure, you need to choose this option. Moreover, you MUST specify the Amazon S3 Region in the option above. This option implements the newer AWS4 (v4) authentication API. Buckets created in Amazon S3 regions brought online after January 2014 (e.g. Frankfurt) will only accept this option. Older buckets will work with either option.

  ### Important

  v4 signatures are only compatible with Amazon S3 proper. If you are using a custom Endpoint this option will NOT work.

- **v2 (legacy mode, third party storage providers)**. If you are using an S3-compatible third party storage service (NOT Amazon S3) you MUST use this option. We do not recommend using this option with Amazon S3 as this authentication method is going to be phased out by Amazon itself in the future.

Bucket Access

This option determines how the API will access the Bucket. If unsure, use the `Virtual Hosting` setting.

The two available settings are:

- **Virtual Hosting (recommended)**. This is the recommended and supported method for Amazon S3. Buckets created after May 2019 will only support this method. Amazon has communicated that this method is the only available in Amazon S3's API starting September 2020.
- **Path Access (legacy)**. This is the older, no longer supported method. You should only need to use it with a custom endpoint and ONLY if your storage provider has told you that you need to enable it.

Directory — The directory inside your Amazon S3 bucket where your files will be stored in. If you want to use subdirectories, you have to use a forward slash, e.g. `directory/subdirectory/subsubdirectory`.

> **Tip**
>
> You can use Akeeba Backup's "variables" in the directory name in order to create it dynamically. These are the same variables as what you can use in the archive name, i.e. [DATE], [TIME], [HOST], [RANDOM].

Disable multipart uploads — Since Akeeba Backup 3.2, uploads to Amazon S3 of parts over 5Mb use Amazon's new multi-part upload feature. This allows Akeeba Backup to upload the backup archive in 5Mb chunks and then ask Amazon S3 to glue them together in one big file. However, some hosts time out while uploading archives using this method. In that case it's preferable to use a relatively small Part Size for Split Archive setting (around 10-20Mb, your mileage may vary) and upload the entire archive part in one go. Enabling this option ensures that, no matter how big or small your Part Size for Split Archives setting is, the upload of the backup archive happens in one go. You MUST use it if you get RequestTimeout warnings while Akeeba Backup is trying to upload the backup archives to Amazon S3.

Storage class — Select the storage class for your data. Standard is the regular storage for business critical data. Please consult the Amazon S3 documentation for the description of each storage class.

Custom endpoint — Enter the custom endpoint (connection URL) of a third party service which supports an Amazon S3 compatible API. Please remember to set the Signature method to v2 when using this option.

Regarding the naming of buckets and directories, you have to be aware of the Amazon S3 rules (these rules are a simplified form of the list S3Fox presents you with when you try to create a new bucket):

- Folder names can not contain backward slashes (\). They are invalid characters.
- Bucket names can only contain lowercase letters, numbers, periods (.) and dashes (-). Accented characters, international characters, underscores and other punctuation marks are illegal characters.

> **Important**
>
> Even if you created a bucket using uppercase letters, **you must type its name with lowercase letters**. Amazon S3 automatically converts the bucket name to all-lowercase. Also note that, as stated above, you may NOT be able to use at all under some circumstances. Generally, your should avoid using uppercase letters.

- Bucket names must start with a number or a letter.
- Bucket names must be 3 to 63 characters long.
- Bucket names can't be in an IP format, e.g. 192.168.1.2
- Bucket names can't end with a dash.
- Bucket names can't have an adjacent dot and dash. For example, both `my.-bucket` and `my-.bucket` are invalid. It's best to avoid dots at all as they are incompatible with the Use SSL option.

If any - or all - of those rules are broken, you'll end up with error messages that Akeeba Backup couldn't connect to S3, that the calculated signature is wrong or that the bucket does not exist. This is normal and expected behaviour, as Amazon S3 drops the connection when it encounters invalid bucket or directory names.

Automatic provisioning of Access and Secret Key on EC2 instances with an attached IAM Role

Starting with version 6.2.0, Akeeba Backup can automatically provision temporary credentials (Access and Secret Key) if you leave these fields blank. This feature is meant for advanced users who automatically deploy multiple sites to Amazon EC2. This feature has four requirements:

- Using Amazon S3, not a custom endpoint. Only Amazon S3 proper works with the temporary credentials issued by the EC2 instance.
- Using the v4 signature method. The old signature method (v2) does not work with temporary credentials issued by the EC2 instance. This is because Amazon requires that the requests authenticated with these credentials to also include the Security Token returned by the EC2 instance, something which is only possible with the v4 signature method.
- Running Akeeba Backup on a site which is hosted on an Amazon EC2 instance. It should be self understood that you can't use temporary credentials issued by the EC2 instance unless you use one. Therefore, don't expect this feature to work with regular hosting; it requires that your site runs on an Amazon EC2 server.
- Attaching an IAM Role to the Amazon EC2 instance. The IAM Role must allow access to the S3 bucket you have specified in Akeeba Backup's configuration.

When Akeeba Backup detects that both the Access and Secret Key fields are left blank (empty) it will try to query the EC2 instance's metadata server for an attached IAM Role. If a Role is attached it will make a second query to the EC2 instance's metadata server to retrieve its temporary credentials. It will then proceed to use them for accessing S3.

The temporary credentials are cached by Akeeba Backup for the duration of the backup process. If they are about to expire or expire during the backup process new credentials will be fetched from the EC2 instance's metadata server using the same process.

Creating and attaching IAM Roles to EC2 instances is beyond the scope of our documentation and our support services. Please refer to Amazon's documentation.

3.3.5.18. Upload to Remote SFTP server

> **Note**
>
> This feature is available only to Akeeba Backup Professional.

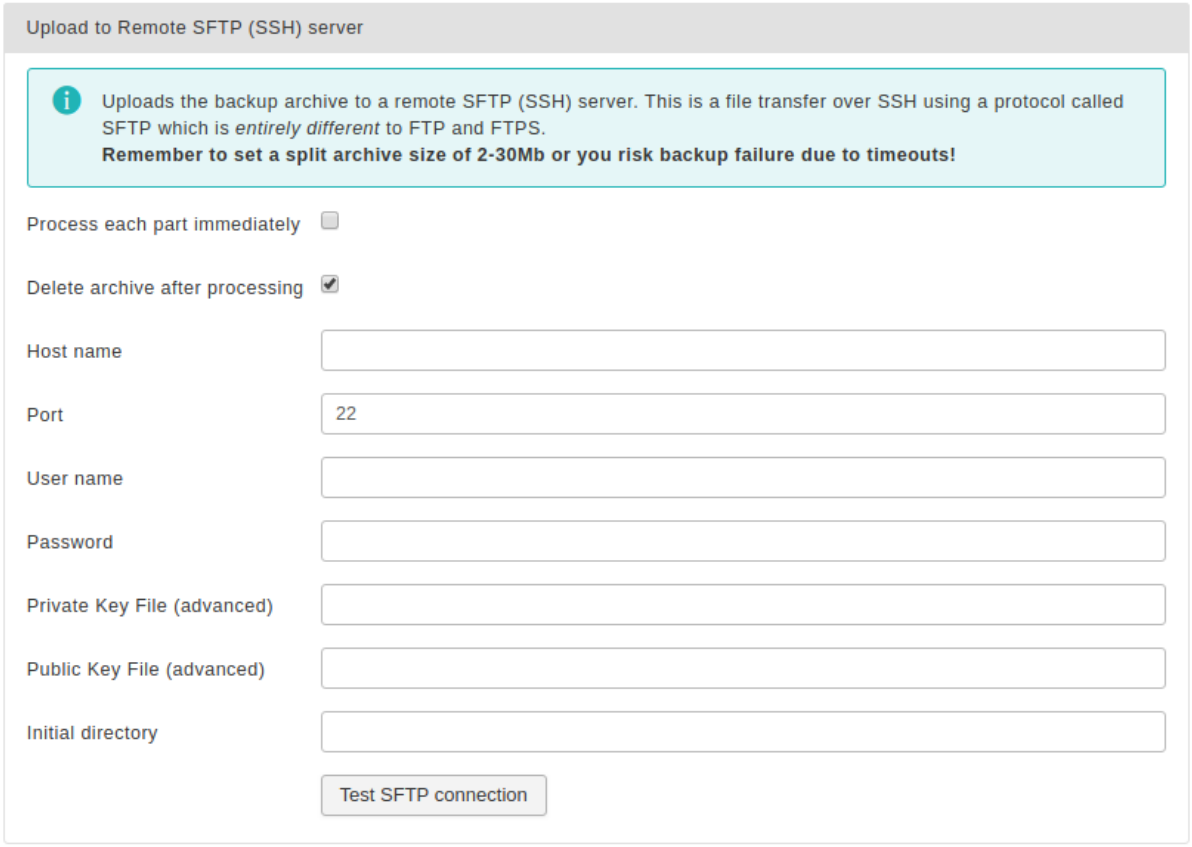
> **Note**
>
> This engine uses the PHP extension called SSH2. The SSH2 extension is still marked as an alpha and is not enabled by default or even provided by many commercial hosts. In this case you may want to use the Upload to Remote SFTP server over cURL engine instead which uses PHP's cURL extension, available on most hosts.

Using this engine, you can upload your backup archives to any SFTP (Secure File Transfer Protocol) server. Please note that SFTP is the *encrypted* file transfer protocol provided by SSH servers. Even though the name is close, it has nothing to do with plain old FTP or FTP over SSL. Not all servers support this but for those which do this is the most secure file transfer option.

The difference of this engine to the DirectSFTP archiver engine is that this engine uploads backup archives to the server, whereas DirectSFTP uploads the uncompressed files of your site. DirectSFTP is designed for rapid migration, this engine is designed for easy moving of your backup archives to an off-server location. Moreover, this engine also supports connecting to your SFTP server using cryptographic key files instead of passwords, a much safer (and much harder and geekier) user authentication method.

Your originating server must have PHP's SSH2 module installed and activated and its functions unblocked. Your originating server must also not block SFTP communication to the remote (target) server. Some hosts apply a firewall policy which requires you to specify to which hosts your server can connect. In such a case you might need to allow communication to your remote host over TCP port 22 (or whatever port you are using).

Upload to Remote SFTP Server



Before you begin, you should know the limitations. SFTP does not allow resuming of uploads so the archive has to be transferred in a single step. PHP has a time limit restriction we can't overlook. The time required to upload a file to SFTP equals the size of the file divided by the available bandwidth. We want to time to upload a file to be less than PHP's time limit restriction to avoid timing out. Since the available bandwidth is finite and constant, the only thing we can reduce in order to avoid timeouts is the file size. To this end, you have to produce split archives, by setting the part size for archive splitting in ZIP's or JPA's engine configuration pane. The suggested values are between 10Mb and 20Mb. Most servers have a bandwidth cap of 20Mbits, which equals to roughly 2Mb/sec (1 byte is 8 bits, plus there's some traffic overhead, lost packets, etc). With a time limit of 10 seconds, we can upload at most 2 Mb/sec * 10 sec = 20Mb without timing out. If you get timeouts during post-processing lower the part size.

The available configuration options are:

Process each part immediately
: If you enable this, each backup part will be uploaded as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the upload fails, the backup fails. If you don't enable this option, the upload process will take place after the backup is complete and finalized. This ensures that if the upload process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.

Delete archive after processing
: If enabled, the archive files will be removed from your server after they are uploaded to the SFTP server.

Host name
: The hostname of your remote (target) server, e.g. `secure.example.com`. You must NOT enter the sftp:// or ssh:// protocol prefix. If you do, Akeeba Backup will try to remove it automatically and issue a warning about it.

Port
: The TCP/IP port of your remote host's SFTP (SSH) server. It's usually 22. If unsure, please ask your host.

User name
: The username you have to use to connect to the remote SFTP server. This must be always provided

Password
: The password you have to use to connect to the remote SFTP server.

Private key file (advanced)
: Many (but not all) SSH/SFTP servers allow you to connect to them using cryptographic key files for user authentication. This method is far more secure than using a password. Passwords can be guessed within some degree of feasibility because of their relatively short length and complexity. Cryptographic keys are night impossible to guess with the current technology due to their complexity (on average, more than 100 times as complex as a typical password).

  If you want to use this kind of authentication you will need to provide a set of two files, your public and private key files. In this field you have to enter the full filesystem path to your private key file. The private key file must be in RSA or DSA format and has to be configured to be accepted by your remote host. The exact configuration depends on your SSH/SFTP server and is beyond the scope of this documentation. If you are a curious geek we strongly advise you to search for "ssh certificate authentication" in your favourite search engine for more information.

  If you are using encrypted private key files enter the passphrase in the Password field above. If it is not encrypted, which is a bad security practice, leave the Password field blank.

  ### Important

  If the libssh2 library that the SSH2 extension of PHP is using is compiled against GnuTLS (instead of OpenSSL) you will NOT be able to use encrypted private key files. This has to do with bugs / missing features of GnuTLS, not our code. If you can't get certificate authentication to work please try providing an unencrypted private key file and leave the Password field blank.

Public Key File (advanced)
: If you are using the key file authentication method described above you will also have to supply the public key file. Enter here the full filesystem path to the public key file. The public key file must be in RSA or DSA format and, of course, unencrypted (as it's a public key).

Initial directory
: The absolute filesystem path to your remote site's location where your archives will be stored. This is provided by your hosting company. Do not ask us to tell you what you should put in here because we can't possibly know. There is an easy way to find it, though. Connect to your target SFTP server with FileZilla. Navigate to the intended directory. Above the right-hand folder pane you will see a text box with a path. Copy this path and paste it to Akeeba Backup's setting.

3.3.5.19. Upload to Remote SFTP server over cURL

Note

This feature is available only to Akeeba Backup Professional.

Note

This engine uses the PHP cURL extension. If your host has disabled the cURL extension but has enabled the SSH2 PHP extension you may want to use the Upload to Remote SFTP server engine instead which uses PHP's SSH2 extension.

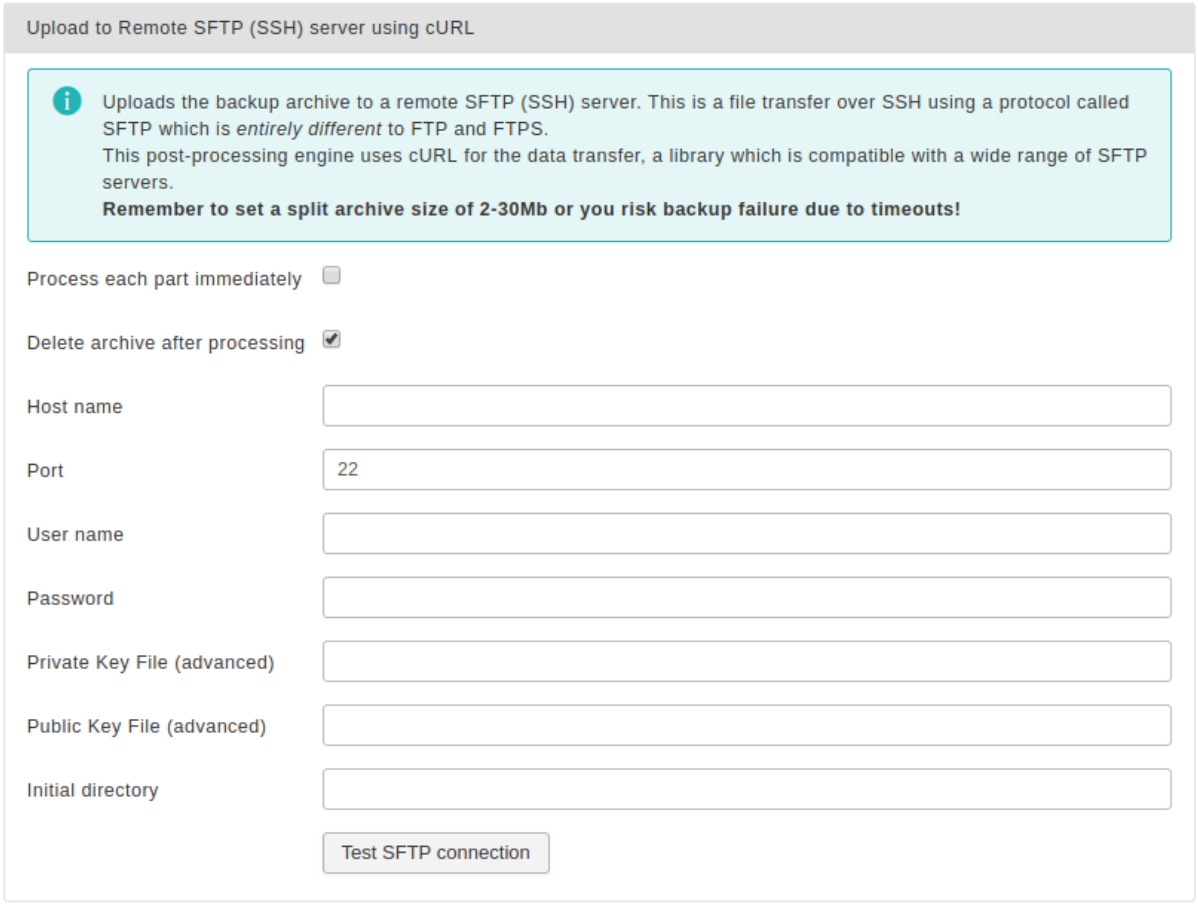
Using this engine, you can upload your backup archives to any SFTP (Secure File Transfer Protocol) server. Please note that SFTP is the *encrypted* file transfer protocol provided by SSH servers. Even though the name is close, it has nothing to do with plain old FTP or FTP over SSL. Not all servers support this but for those which do this is the most secure file transfer option.

The difference of this engine to the DirectSFTP over cURL archiver engine is that this engine uploads backup archives to the server, whereas DirectSFTP over cURL uploads the uncompressed files of your site. DirectSFTP over cURL is designed for rapid migration, this engine is designed for easy moving of your backup archives to an off-server location.

Your originating server (where you are backing up *from*) must a. have PHP's cURL extension installed and activated, b. have the cURL extension compiled with SFTP support and c. allow outbound TCP/IP connections to your target host's SSH port. Please note that some hosts provide the cURL extension without SFTP support. This feature will NOT work on these hosts. Moreover, some hosts apply a firewall policy which requires you to specify to which hosts your server can connect. In such a case you might need to allow communication to your remote host.

Before you begin, you should know the limitations. SFTP does not allow resuming of uploads so the archive has to be transferred in a single step. PHP has a time limit restriction we can't overlook. The time required to upload a file to SFTP equals the size of the file divided by the available bandwidth. We want to time to upload a file to be less than PHP's time limit restriction to avoid timing out. Since the available bandwidth is finite and constant, the only thing we can reduce in order to avoid timeouts is the file size. To this end, you have to produce split archives, by setting the part size for archive splitting in ZIP's or JPA's engine configuration pane. The suggested values are between 10Mb and 20Mb. Most servers have a bandwidth cap of 20Mbits, which equals to roughly 2Mb/sec (1 byte is 8 bits, plus there's some traffic overhead, lost packets, etc). With a time limit of 10 seconds, we can upload at most 2 Mb/sec * 10 sec = 20Mb without timing out. If you get timeouts during post-processing lower the part size.

Upload to Remote SFTP Server over cURL



The available configuration options are:

Process each part immediately
: If you enable this, each backup part will be uploaded as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the upload fails, the backup fails. If you don't enable this option, the upload process will take place after the backup is complete and finalized. This ensures that if the upload process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.

Delete archive after processing
: If enabled, the archive files will be removed from your server after they are uploaded to the SFTP server.

Host name
: The hostname of your remote (target) server, e.g. `secure.example.com`. You must NOT enter the sftp:// or ssh:// protocol prefix. If you do, Akeeba Backup will try to remove it automatically and issue a warning about it.

Port
: The TCP/IP port of your remote host's SFTP (SSH) server. It's usually 22. If unsure, please ask your host.

User name
: The username you have to use to connect to the remote SFTP server. This must be always provided

Password
: The password you have to use to connect to the remote SFTP server.

Private key file (advanced)
: Many (but not all) SSH/SFTP servers allow you to connect to them using cryptographic key files for user authentication. This method is far more secure than using a password. Passwords can be guessed within some degree of feasibility because of their relatively short length and complexity. Cryptographic keys are night impossible to guess with the current technology due to their complexity (on average, more than 100 times as complex as a typical password).

  If you want to use this kind of authentication you will need to provide a set of two files, your public and private key files. In this field you have to enter the full filesystem path to your private key file. The private key file must be in RSA or DSA format and has to be configured to be accepted by your remote host. The exact configuration depends on your SSH/SFTP server and is beyond the scope of this documentation. If you are a curious geek we strongly advise you to search for "ssh certificate authentication" in your favourite search engine for more information.

  If you are using encrypted private key files enter the passphrase in the Password field above. If it is not encrypted, which is a bad security practice, leave the Password field blank.

  ## Important

  If cURL is compiled against GnuTLS (instead of OpenSSL) you will NOT be able to use encrypted private key files. This has to do with bugs / missing features of GnuTLS, not our code. If you can't get certificate authentication to work please try providing an unencrypted private key file and leave the Password field blank.

Public Key File (advanced)
: If you are using the key file authentication method described above you will also have to supply the public key file. Enter here the full filesystem path to the public key file. The public key file must be in RSA or DSA format and, of course, unencrypted (as it's a public key). Some newer versions of cURL allow you to leave this blank, in which case they will derive the public key information from the private key file. We do not recommend this approach.

Initial directory
: The absolute filesystem path to your remote site's location where your archives will be stored. This is provided by your hosting company. Do not ask us to tell you what you should put in here because we can't possibly know. There is an easy way to find it, though. Connect to your target SFTP server with FileZilla. Navigate to the intended directory. Above the right-hand folder pane you will see a text box with a path. Copy this path and paste it to Akeeba Backup's setting.

3.3.5.20. Upload to SugarSync

Note

This feature is available only to Akeeba Backup Professional 3.5.a1 and later.

Using this engine, you can upload your backup archives to the SugarSync [http://www.sugarsync.com] cloud storage service. SugarSync has a free tier (with 5Gb of free space) and a paid tier. Akeeba Backup can work with either one.

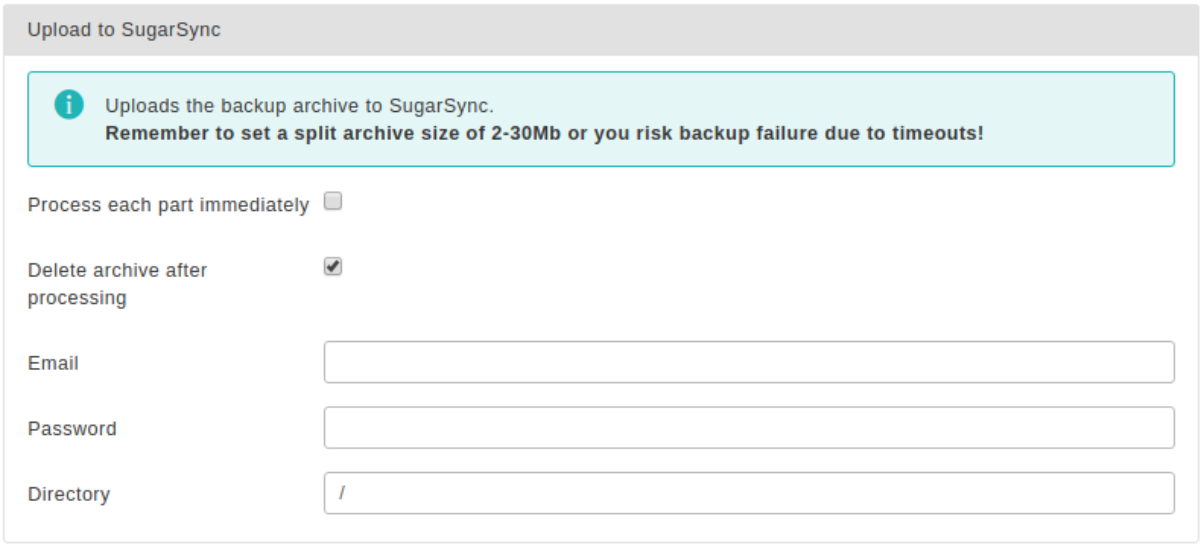
Please note that Akeeba Backup can only upload files to Sync Folders, it can not upload files directly to a Workspace (a single device). You have to set up your Sync Folders in SugarSync before using Akeeba Backup. If you have not created or specified any Sync Folder, Akeeba Backup will upload the backup archives to your Magic Briefcase, the default Sync Folder which syncs between all of your devices, including your mobile devices (iPhone, iPad, Android phones, ...).

Before you begin, you should know the limitations. As most cloud storage providers, SugarSync does not allow appending to files, so the archive has to be transferred in a single step. PHP has a time limit restriction we can't overlook. The time required to upload a file to SugarSync equals the size of the file divided by the available bandwidth. We want to time to upload a file to be less than PHP's time limit restriction so as to avoid timing out. Since the available bandwidth is finite and constant, the only thing we can reduce in order to avoid timeouts is the file size. To this end, you have to produce split archives, by setting the part size for archive splitting in ZIP's or JPA's engine configuration pane. The suggested values are between 10Mb and 20Mb. Most servers have a bandwidth cap of 20Mbits, which equals to roughly 2Mb/sec (1 byte is 8 bits, plus there's some traffic overhead, lost packets, etc). With a time limit of 10 seconds, we can upload at most 2 Mb/sec * 10 sec = 20Mb without timing out. If you get timeouts during post-processing lower the part size.

Tip

If you use the native CRON mode (akeeba-backup.php), there is usually no time limit - or there is a very high time limit in the area of 3 minutes or so. Ask your host about it. Setting up a profile for use only with the native CRON mode allows you to increase the part size and reduce the number of parts a complete backup consists of.

Upload to SugarSync



The required settings for this engine are:

Process each part immediately	If you enable this, each backup part will be uploaded as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after

processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the upload fails, the backup fails. If you don't enable this option, the upload process will take place after the backup is complete and finalized. This ensures that if the upload process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.

Delete archive after processing — If enabled, the archive files will be removed from your server after they are uploaded to SugarSync.

Email — The email used by your SugarSync account.

Password — The password used by your SugarSync account.

Directory — The directory inside SugarSync where your files will be stored in. If you want to use subdirectories, you have to use a forward slash, e.g. `/directory/subdirectory/subsubdirectory`. You may use the same variables used in archive naming, e.g. [HOST] for the site's host name or [DATE] for the current date.

Please note that the first part of your directory should be the name of your shared folder. For example, if you have a shared folder named `backups` and you want to create a subdirectory inside it based on the site's name, you need to enter `backups/[HOST]` in the directory box. If a Sync Folder by the name "backups" is not found, a directory named "backups" will be created inside your Magic Briefcase folder. Yes, it's more complicated than, say, DropBox – but that's also why SugarSync is more powerful.

3.3.5.21. Upload to WebDAV

Note

This feature is available only to Akeeba Backup Professional 3.10.1 and later.

Using this engine, you can upload your backup archives to any server which supports the WebDAV (Web Distributed Authoring and Versioning) protocol. Examples of storage services supporting WebDAV:

- OwnCloud [http://doc.owncloud.org/server/5.0/user_manual/files/files.html] is a software solution that you can install on your own servers to provide a private cloud.
- CloudDAV [http://storagemadeeasy.com/CloudDav/] is a service which gives you WebDAV access to a plethora of cloud storage providers: Amazon S3, GMail, RackSpace CloudFiles, Microsoft OneDrive (formerly: SkyDrive), Windows Azure BLOB Storage, iCloud, LiveMesh, Box.com, FTP servers, Email (which, unlike the Send by email engine in Akeeba Backup, does support large files), Google Docs, Mezeo, Zimbra, FilesAnywhere, Dropbox, Google Storage, CloudMe, Microsoft SharePoint, Trend Micro, OpenStack Swift (supported by several providers), Google sites, HP cloud, Alfresco cloud, Open S3, Eucalyptus Walrus, Microsoft Office 365, EMC Atmos, iKoula - iKeepinCloud, PogoPlug, Ubuntu One, SugarSync, Hosting Solutions, BaseCamp, Huddle, IBM Files Cloud, Scality, Google Drive, Memset Memstore, DumpTruck, ThinkOn, Evernote, Cloudian, Copy.com, Salesforce. [TESTED with Amazon S3 as the storage provider]
- Apache web server (when the optional WebDAV support is enabled – recommended for advanced users only).
- 4Shared [http://www.4shared.com/].
- ADrive [http://www.adrive.com/].
- Amazon Cloud Drive [http://www.amazon.com/gp/feature.html/ref=cd_def?ie=UTF8&*Version*=1&*entries*=0&docId=1000828861].
- Box.com [https://www.box.com/].

- CloudSafe [https://secure.cloudsafe.com/login/].
- DriveHQ [https://www.drivehq.com/].
- DumpTruck [http://www.goldenfrog.com/].
- FilesAnywhere [https://www.filesanywhere.com/].
- MyDrive [http://www.mydrive.net/].
- MyDisk.se. [https://mydisk.com/web/main.php?show=home]
- PowerFolder [https://www.powerfolder.com/].
- OVH.net [http://ovh.net/]
- Safecopy Backup [http://safecopybackup.com/].
- Strato HiDrive [https://www.free-hidrive.com/index.html].
- Telekom Mediencenter [http://mediencenter.telekom.de/].
- Pretty much every storage provider which claims to support WebDAV

Tip

You can find more information for WebDAV access of each of these providers in http://www.free-online-backup-services.com/features/webdav.html

Note

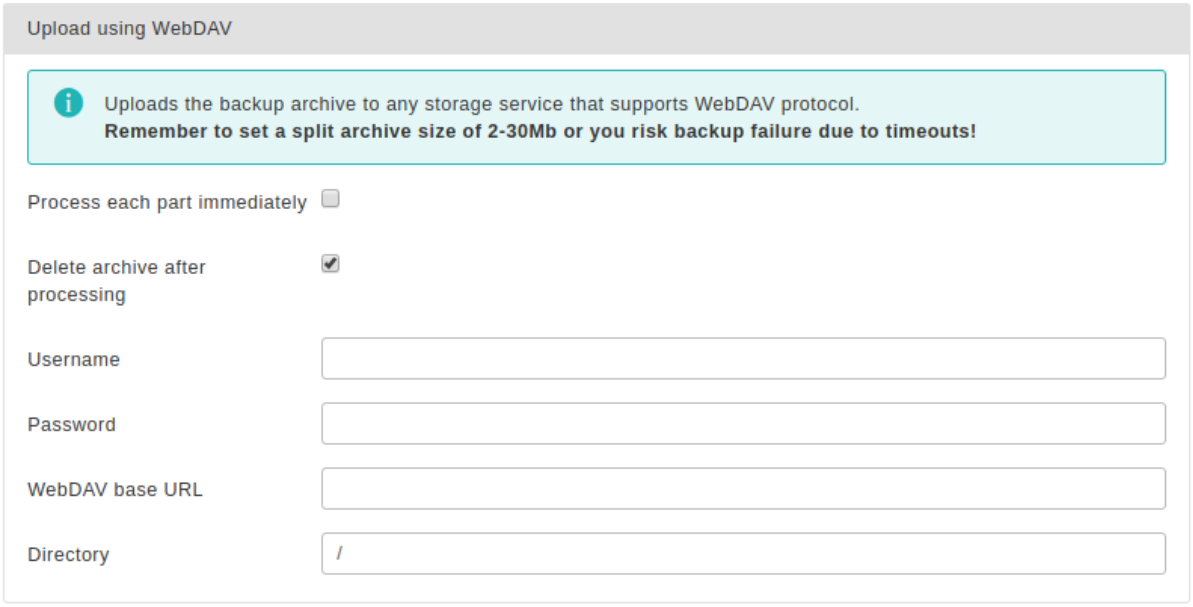
We have not thoroughly tested and do not guarantee that any of the above providers will work smoothly with Akeeba Backup unless you see the notive [TESTED] next to it.

Before you begin, you should know the limitations. As most remote storage technologies, WebDAV does not allow appending to files, so the archive has to be transferred in a single step. PHP has a time limit restriction we can't overlook. The time required to upload a file to WebDAV equals the size of the file divided by the available bandwidth. We want to time to upload a file to be less than PHP's time limit restriction so as to avoid timing out. Since the available bandwidth is finite and constant, the only thing we can reduce in order to avoid timeouts is the file size. To this end, you have to produce split archives, by setting the part size for archive splitting in ZIP's or JPA's engine configuration pane. The suggested values are between 10Mb and 20Mb. Most servers have a bandwidth cap of 20Mbits, which equals to roughly 2Mb/sec (1 byte is 8 bits, plus there's some traffic overhead, lost packets, etc). With a time limit of 10 seconds, we can upload at most 2 Mb/sec * 10 sec = 20Mb without timing out. If you get timeouts during post-processing lower the part size.

Tip

If you use the native CRON mode (akeeba-backup.php), there is usually no time limit - or there is a very high time limit in the area of 3 minutes or so. Ask your host about it. Setting up a profile for use only with the native CRON mode allows you to increase the part size and reduce the number of parts a complete backup consists of.

Upload to WebDAV



The required settings for this engine are:

Process each part immediately
: If you enable this, each backup part will be uploaded as soon as it's ready. This is useful if you are low on disk space (disk quota) when used in conjunction with Delete archive after processing. When using this feature we suggest having 10Mb plus the size of your part for split archives free in your account. The drawback with enabling this option is that if the upload fails, the backup fails. If you don't enable this option, the upload process will take place after the backup is complete and finalized. This ensures that if the upload process fails a valid backup will still be stored on your server. The drawback is that it requires more available disk space.

Delete archive after processing
: If enabled, the archive files will be removed from your server after they are uploaded to SugarSync.

Username
: The username you use to connect to your WebDAV server

Password
: The password you use to connect to your WebDAV server

WebDAV base URL
: The base URL of your WebDAV server's endpoint. It might be a directory such as `http://www.example.com/mydav/` or even a script endpoint such as `http://www.example.com/webdav.php`. If unsure please ask your WebDAV provider for more information.

Warning

If the base URL of your WebDAV server's endpoint is a directory (almost always) you MUST use a trailing slash, e.g. `http://www.example.com/mydav/` (correct) but not `http://www.example.com/mydav` (WRONG!)

Directory
: The directory inside the WebDAV folder where your files will be stored in. If you want to use subdirectories, you have to use a forward slash, e.g. `/directory/subdirectory/subsubdirectory`. You may use the same variables used in archive naming, e.g. [HOST] for the site's host name or [DATE] for the current date.

Warning

You MUST always use a directory. Most WebDAV servers, e.g. Box.com, allow you to use the root directory which is denoted by `/` (a single forward slash). Other

WebDAV servers, such as CloudDAV, DO NOT allow you to use the root directory. In this case you MUST use a non-empty directory, e.g. `/backups` for the upload to WebDAV to work at all.

3.3.5.22. Upload to Box.net / Box.com

As of Akeeba Backup 3.10.1 you can use the Upload to WebDAV option to upload your backup archives to Box.com. You will need to use the following parameters:

Username	Your box.com email address
Password	Your box.com password
WebDAV base URL	`https://dav.box.com/dav`

For more information please check the official Box.com page explaining the Box.com over WebDAV feature: https://support.box.com/hc/en-us/articles/200519748-Does-Box-support-WebDAV-

Important

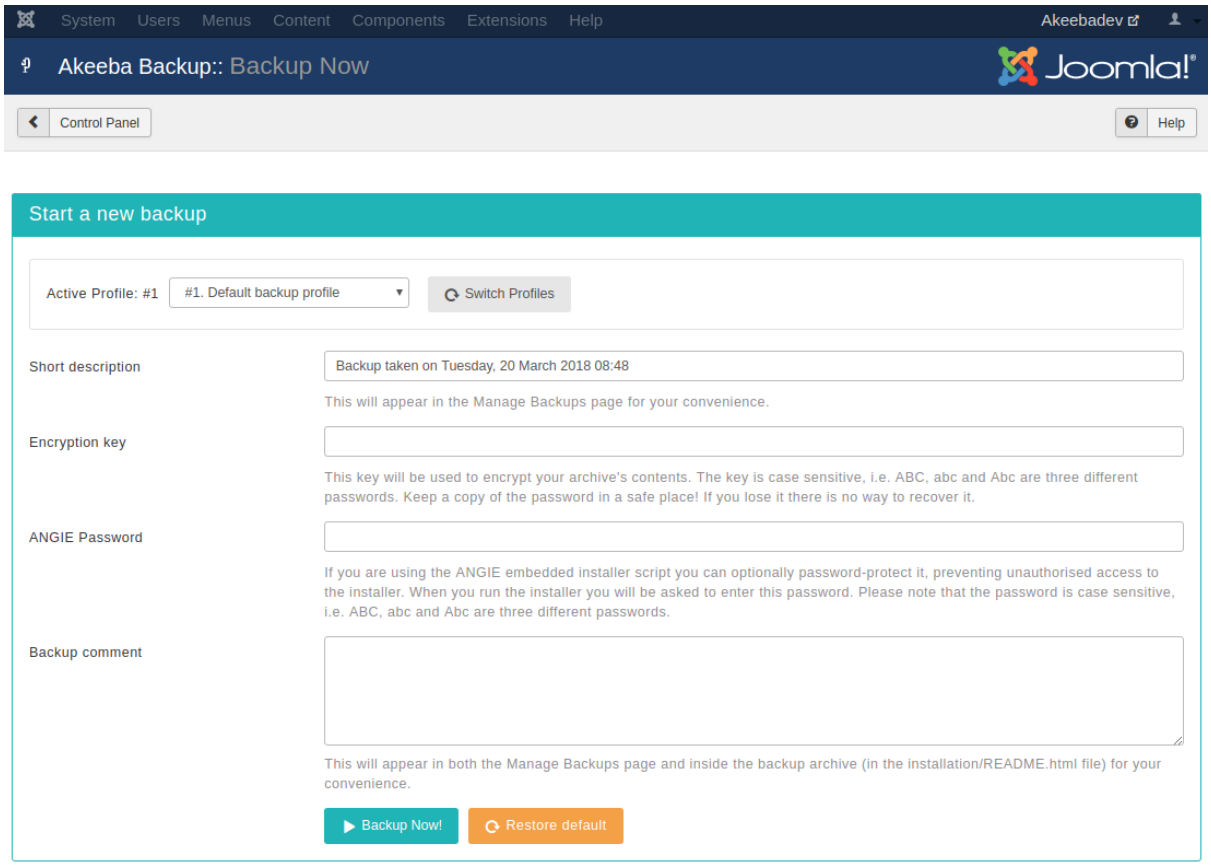
Due to limitations in the Box.com implementation of WebDAV we strongly recommend using a Part Size for Split Archives smaller than 50Mb at all times.

3.4. Backup now

Before we go on describing the Backup Now page, we have to discuss something important pertaining to the overall backup and restoration process. In order for the restoration to work properly, the original site must have a readable and valid configuration.php on its root. This means that a 'trick' some very few webmasters use, providing a configuration.php which includes an off-server-root PHP file, is incompatible with the restoration procedure. If the 'trick' has been effective on the original site, the installer will have blanks in its options and if the user proceeds with the restoration/installation procedure the site will not work as expected, as crucial options will have the default or no value at all!

Moreover we would like to remind you that restoring to a temporary URL (something like http://www.yourhost.com/~youruser) will NOT work. This has to do with the way Joomla!, Apache and PHP session management works. It has nothing to do with Akeeba Backup. Even if you install straight up Joomla! on a temporary URL you'll have problems logging in or, at the very least, with SEF URLs.

Backup start



The initial backup page lets you define a short description (required) and an optional lengthy comment for this backup attempt. This information will be presented to you in the backup administration page to help you identify different backups. The default description contains the date and time of backup. Both the description and comment will be stored in a file named `README.html` inside your archive's `installation` directory, but only if the backup mode is full backup.

Since Akeeba Backup 3.1.b1 both the description and the comment support Akeeba Backup's file naming "variables", e.g. `[SITE]`, `[DATE]` and `[TIME]`. These variables are documented in the Output Directory configuration option's description. It goes without saying, but these variables can also be used in the case of automated backups, e.g. CRON-mode backups.

There are two more fields which may be displayed on this page:

- Encryption key. When you are using the JPS (encrypted archive) format the contents of the archive are encrypted using the AES-256 algorithm and this password. In order to extract the archive you will need to enter this password. If you had entered a default password for JPS files in the Configuration page this field is pre-filled with that password.

  ### Important

  The password is case-sensitive. ABC, abc and Abc are three different passwords! Also note that the password is non-recoverable. If you lose or forget your password you will not be able to extract your JPS archive.

- ANGIE Password. As of Akeeba Backup 3.7.5 the ANGIE installer (embedded in the backup archive) allows you to password protect it. This means that you will have to enter this password before you can restore your site. This feature is designed to prevent unauthorised users from "stumbling" on your site while it's still undergoing restoration and copy your database passwords or obtain other information about your site.

We STRONGLY advise that you always use an ANGIE Password if you intend to restore your site on a live server. This is the only way to prevent accidental information leak.

Important

The password is case-sensitive. ABC, abc and Abc are three different passwords! Unlike the JPS password, setting an ANGIE password will not prevent anyone from extracting the archive and looking at its contents. It will only prevent people attempting to browse your site while you're restoring a backup from seeing potentially confidential information in the installer.

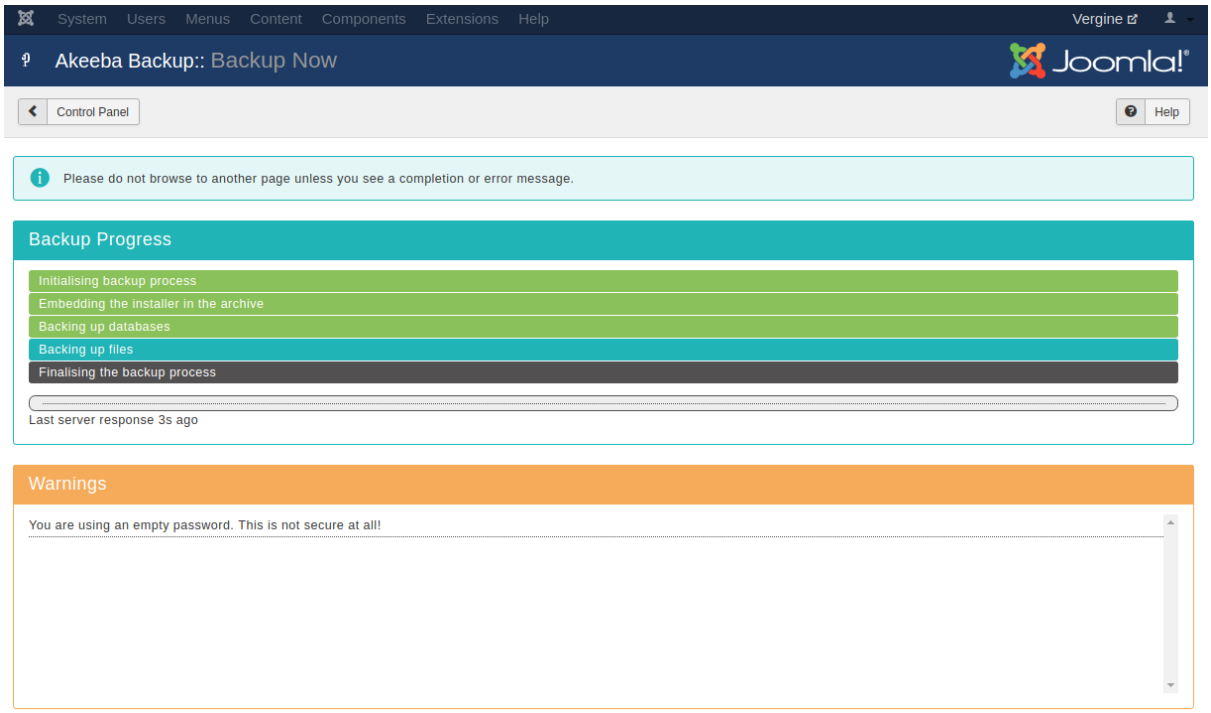
Whenever you are ready to start the backup, just click the Backup Now button. Do note that above the description field, there might be one or more warnings. These are the same warnings appearing in the Control Panel's right-hand pane and act as a reminder.

Important

Default output directory is in use *is not an error message*! It's just a reminder that the default output directory is a well known location on your site. In theory, a malicious user could figure out the name of the backup archive and download it directly over the web. In order to deter that, Akeeba Backup places a .htaccess file (compatible with virtually all Apache installations) and a web.config file (compatible only with IIS 7) to deter that. If you are using a host which doesn't support the directives of those two files, the contents of that directory may be inadvertently available over the web to malicious users. If in doubt, ask your host. Do not ask us, please, we are not your host.

Our recommendation: consult your host about the proper way to create a backup output directory above your site's root and make it writable by PHP. Then, use that directory as the Output Directory in all of your backup profiles. This method offers the best protection.

Backup progress page



Once you click on the Backup Now button, the backup progress page appears. You must not navigate away from this page or close your browser window until the backup is complete. Otherwise, the backup process will be interrupted and no backup file will be created (or you'll end up with an incomplete backup file). Akeeba Backup disables the Joomla! menu during backup to prevent accidentally switching to a different page.

The backup progress page consists of a large pane. The top section of the pane lists the steps Akeeba Backup has to take in order to complete your backup. Steps in gray background have not been dealt with yet. Steps in green background have been successfully completed. The step in blue background is the one being currently processed.

Below that, you will find two lines. The first line will show you which table or directory has **been backed up in the previous step**. This is very important. When the backup crashes, it hasn't necessarily crashed backing up the table or directory you see on the screen. Most likely that the table/directory which has been *successfully* backed up. The real problem appears in the log file and this is why we are adamant in asking for a backup log to be posted with your support request. The line below is normally used for messages of lesser importance, such as noting the percentage of a table already completed (especially useful when backing up huge tables) and the name of the archive part which was processed by a data processing engine.

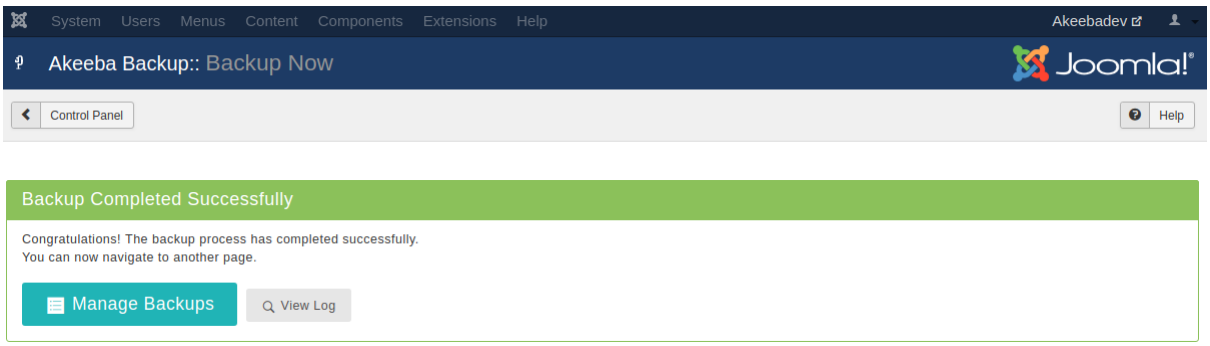
The big bar is the overall progress bar and displays an *approximation* of the backup progress. Do note that during file backup you may see this bar jump back and forth. This is normal and, please, do not report it as a bug. It is exactly how it is supposed to behave. The reason is rather simple. Before your site is backed up, Akeeba Backup doesn't know how many files and directories it contains. As a result, it tries to do an educated guess and display an approximate backup progress. Guesswork is never accurate, which causes some jumping back and forth. Nothing to worry about, your backup is working without a problem.

The next thing you see is time elapsed since the last server response. This resets to 0 when a new backup step is started. If you see a last server response over 300 seconds –except when the application is uploading your backup archives– you can assume that your backup has crashed. Only in this case you should navigate away from the backup page and take a look at the log file for any error messages. Always try different configuration options, especially changing the minimum and maximum execution time, before filing a support request.

Should a minor (non fatal) error occur, Akeeba Backup displays a new Warnings pane with yellow background. This box holds the warnings which have occurred during the backup process, in chronological order. These are also logged in the Akeeba Backup Debug Log and marked with the WARNING label, that is if your log level is at least Errors and Warnings. Usual causes of warnings are unreadable files and directories. Akeeba Backup regards them as minor errors because, even though the backup process can go through, what you get might be a partial backup which doesn't meet your expectations. In case warnings appear on your screen you are advised to review them and assess their importance.

Sometimes your backup may halt with an AJAX error. This means that there was a communications error between the browser and your server. In most cases this is a temporary server or network issue. Depending on your configuration preferences, Akeeba Backup may try to resume the backup after a while. By default, Akeeba Backup will retry resuming the backup at most three consecutive times and after waiting 10 seconds after each error. If the backup cannot be resumed you will receive an error page, at which point your backup has positively failed.

Backup completion page



After the whole process is complete, Akeeba Backup will clean up any temporary files it has created. Akeeba Backup will also clean temporary files and delete incomplete archive files upon detecting a backup failure. Please note that log files are not removed by default. You will have to go to the Manage Backups page, select the failed backup attempt(s) and then click on Delete Files or Delete to have it remove the log files of failed backups.

By that point, your site backup file has been created. You can now navigate out of the backup page and possibly into the backup administration page, clicking on the handy button which appears below the backup completion message.

Frequently asked questions

Where are my backup files? [https://www.akeebabackup.com/documentation/troubleshooter/abwherearemyfiles.html]

How can I download my backup files? [https://www.akeebabackup.com/documentation/troubleshooter/abwherearemyfiles.html]

I got an "AJAX loading error" when backing up. What should I do? [???]

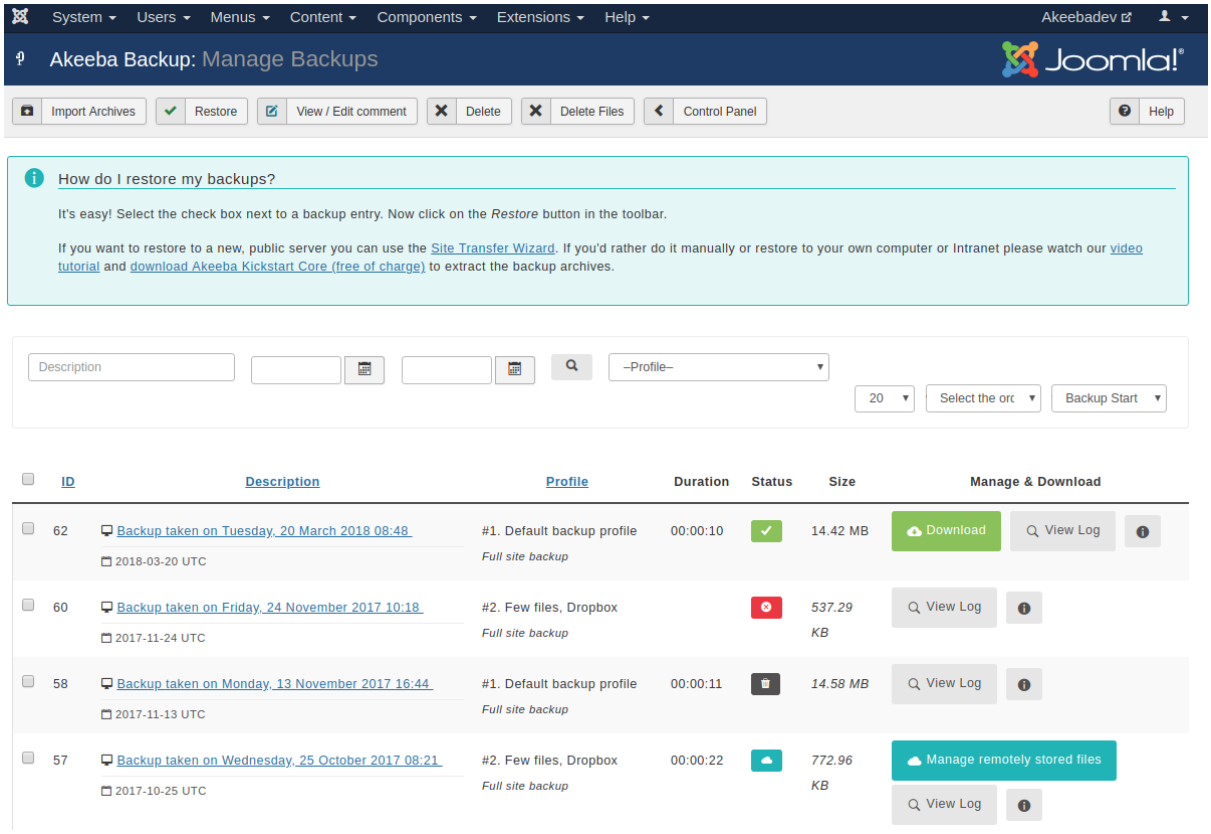
How do I know that my backup archive works? [https://www.akeebabackup.com/documentation/troubleshooter/abtestsupport.html]

What happens if I have a backup problem? [https://www.akeebabackup.com/documentation/troubleshooter/abtestsupport.html]

How do I get support? [https://www.akeebabackup.com/documentation/troubleshooter/abtestsupport.html]

3.5. Manage Backups

Manage Backups



This page is the single place you can review all your Akeeba Backup backup history, as well as administer the backup files. The bulk of the page consists of a standard Joomla!™ list table. Each row represents a backup attempt and displays a whole lot of information:

The check box column
: Clicking the check box on the leftmost cell of a row selects this backup for an operation to be applied to it. Operations are activated by clicking on tool bar buttons. In case of an operation allowing a single row to be selected, the topmost selected row is considered as the sole selection.

Description
: Displays the description you have set when you started the backup. If your backup has a comment attached to it, an info icon will also appear. Hovering your mouse over the info icon will show you a preview of that comment.

  To the left of the description there's an icon indicating the backup origin, e.g. Backend, Frontend, JSON API, CLI and so on. Hover over it to see what each icon means.

  Below the description you will see the date and time of the backup. The date / time format, timezone and timezone suffix are configured in the component's Options page.

Profile
: Displays the numeric identifier (and description, if available) of the backup profile used during the backup. It is possible that since the time of the backup the profile may have been modified or even deleted!

  Below it you will see the backup type. It indicates the backup type. A backup type may not be provided if the backup profile has been deleted in the meantime.

Duration
: The duration of the backup in hours : minutes : seconds format. This information is not available for failed backups!

Status
: Indicates the status of the backup. Hover over the icon to see what it means. It will be one of:

  OK (Green)
  : A complete backup whose backup archive is available for download.

  Obsolete (Gray)
  : A complete backup whose backup archive is either deleted, or was overwritten by another backup attempt.

    ## Note

    If you move your backup output directory's location, all your previous backups will appear as "Obsolete", even though you might have moved these backup files as well. This is not a bug. Akeeba Backup internally stores the absolute path to the backup files. When you move the output directory its absolute path changes, so Akeeba Backup is unable to locate the old backup files.

    ## Important

    If your host uses MySQL 4.0 the status will always appear as Obsolete and you will be unable to download the backup archive through your browser, as the result of limitations of this *ancient, obsolete and unsupported* MySQL version. You can still use your favorite FTP client to download the backup archives, though.

  Remote (Blue)
  : Indicates a complete backup which has been uploaded to remote storage (e.g. Dropbox, Amazon S3, CloudFiles and so on), but it is no longer stored on your server. You can fetch the backup archive backup to your server any time (as long as you haven't manually removed the file from the remote storage) in order to restore it, clicking the Manage Remote Files link on the right-hand column.

    ## Note

    Not all remote storage engines support fetching back backup archives. Currently, only FTP, Amazon S3, CloudFiles and Dropbox support this feature.

	Pending (Yellow)	A backup attempt which is still running. You should not see any such record, unless a backup attempt started while you were loading this page. In this case, you should not navigate to the Control Panel page! Doing so would invalidate the backup and wreck havoc. You have been warned! Another reason to see such an entry is a backup attempt which failed with a PHP fatal error, or which was abruptly interrupted (by the user or a PHP error). In this case, you can safely delete the entry and get rid of the backup file as well.
	Failed (Red)	A backup attempt which failed with a catchable error condition.
Size	The total size of the backup archive in MB. If the files are not available on your server, i.e. the record is marked as "obsolete" or "remote", the size appears inside parentheses to let you know that the files are not available for download.	
Manage and Download	Depending on the status of the backup it will show two or more buttons: • Download. Opens a popup which allows you to download the backup archive file(s) directly from your browser. However, this is NOT recommended. The only guaranteed method of downloading your backup archives error-free is using FTP or SFTP in BINARY transfer mode. Anything else has the potential to CORRUPT your backup archives for reasons beyond our control! • Manage remotely stored files. If the file is stored on a remote storage location, e.g. Amazon S3 or a remote FTP server, you will also see this button. Clicking on it will allow you to transfer the files back to your server, download them directly from the remote location or remove them from the remote storage. • Upload to <remote storage name>. If Akeeba Backup failed to upload your backup archive to remote storage you will be shown this button. Clicking it will have Akeeba Backup retry the upload to remote storage. • View Log. If your backup archive has a backup ID you will also see this button. Clicking it takes you to the View Log page to see the backup log file. If the backup status is anything other than OK this button will be grayed over as Akeeba Backup can't guarantee that the log file is present. Hover your mouse over the button to get the Log file ID which you'll need in the View Log page to look for this log file. • Info. Clicking this button tells you if the backup archive is currently present on your server, where to find it (relative to your site's root directory) and what is the name of the backup archive file. This allows you to download the backup archive over FTP/SFTP as discussed above.	

Clicking on the label of each column allows you to sort the backup entries by the contents of that column. By default, Akeeba Backup sorts the records by the time of backup descending, so that the newest backup attempts will appear on top. Below the header there are four filter boxes. The first one allows you to filter by the backup description. The other two allow you to select a date range so that only backups attempted within this date range will be displayed. You can leave either of these boxes empty to allow an open start or end date respectively. The final box allows you to filter by backup profile.

On the top of the page you can find a tool bar with operations buttons. The Delete button will remove the selected backup attempt entries along with their backup archives (if applicable), whereas the Delete Files button will only remove the files (if found on your server). The Restore button (Akeeba Backup Professional only) will run the integrated restoration feature for the selected archive file. This feature can be used to restore your backup archive on the same server you backed up from or even a different server (live transfer of your site to another host!). The Discover and Import Archives (available since Akeeba Backup Professional 3.2) allows you to import any ZIP, JPA or JPS file, located anywhere in your server or Amazon S3, in the Manage Backups (formerly "Administer Backup Files") page in order to restore it on this or any other site.

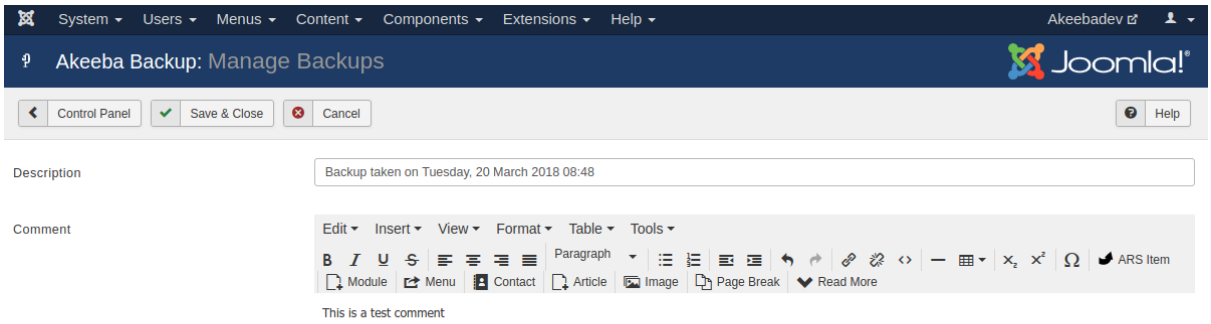
Note

If you are interested in restoring your backup archives and your site is inaccessible or you're using the free Akeeba Backup Core edition, you can use Akeeba Kickstart to extract the archive and restore it on their server. The procedure is detailed in our Video Tutorials.

Important

Integrated restoration is only supported for Full Site and Files Only backup archives. Trying to use it with any other type of backup files will ultimately result in an error. This feature is available only to Akeeba Backup Professional - the paid version. Users of the Akeeba Backup Core version can follow our video tutorials to easily restore their backups using Kickstart.

Backup description / comment editor



The View / Edit Comment button will open a page showing the description and comment of the currently selected backup row. You can freely edit both the description and the comment on that page and save your changes using the Save & Close button. The same page will open if you click on a backup record's description (appearing as a link).

3.5.1. Integrated restoration

Warning

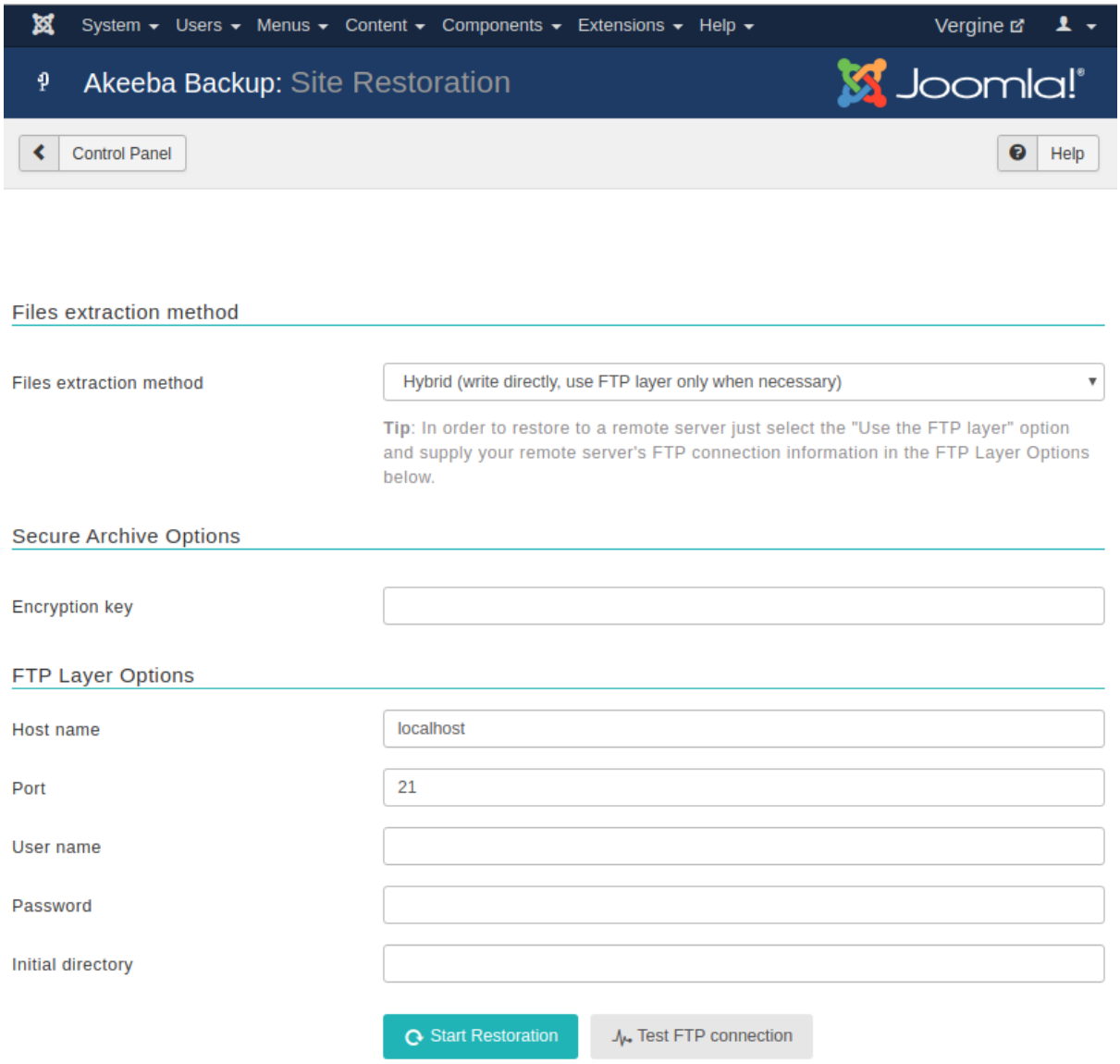
Using the integrated restoration you are OVERWRITING your site with the one contained in the backup archive. The same precautions apply as with any backup restoration.

The integrated restoration feature allows you to easily restore a previous backup directly on your server, as long as your backup archive still exists on your server of course. The whole idea behind this feature is that it is not necessary to manually download Kickstart, place it in your site's root and move the backup archive from the output directory to the site's root in order to perform the restoration. Instead, the integrated restoration feature takes care of extracting your backup archive directly from the backup output folder into your site's root and then allow you to run the embedded installer (Akeeba Backup Installer) to complete the restoration procedure.

The communication between your browser and the archive extraction script is encrypted with the AES128 (Rijndael) encryption method, using a random key produced as soon as you initiate the restoration of a backup archive. This ensures that a malicious user can't exploit the restoration script to mischievously extract your backup archive in your site's root with the intent to steal your database password. The encryption/decryption algorithm is implemented with standard PHP and Javascript code, eliminating the need for third party cryptography libraries and ensuring that under no circumstances unencrypted data will be exchanged between the browser and the server.

In order to start an integrated restoration begin by going to the Manage Backups page of the component. In that page check the checkbox next to the backup you want to restore and click the Restore button in the toolbar to will run the integrated restoration feature for the selected archive file.

The integrated restoration setup page



When you first start the integrated restoration feature, you are presented with a few settings. The first setting, appearing above the Start Restoration button, determines how the file extraction will be performed. The two available options are:

Write directly to files
: All files will be extracted directly to their final location using direct PHP file writes. If your permissions settings do not allow some files or directories to be created/overwritten the process will fail and your site will be left in a half-restored state.

Use FTP uploads
: Using this method, each file is first extracted to the temporary directory specified by the current profile and then moved to its final location using FTP. This is a "best effort" approach and can work with most servers. Do note that only unencrypted FTP (plain FTP) is supported. If you choose this option, you'll also have to specify the FTP connection settings.

Tip

You can use this option to restore a backup on a different site. Just select this option and provide the FTP connection details to the other site before clicking on Start Restoration.

Hybrid — This mode combines the previous two in an intelligent manner. When selected, Akeeba Backup will first attempt to write to the files directly. If this is not possible, i.e. due to permissions or ownership of the file or folder being exracted, it will automatically make use of the FTP mode to overcome the permissions / ownership problem. It effectively works around a situation commonly called "permissions hell", where different files and folders are owned by different users, making it extremely difficult to overwrite them. This is a situation which happens very commonly on shared hosting. Therefore **we strongly advise clients on shared hosting environments to use the Hybrid option**.

Note

You MUST supply your FTP information for this mode to have any effect. If you do not do that the Hybrid mode will function exactly as the "Write directly to files" mode.

The default mode is writing directly to files, unless your site's Global Configuration indicates that the FTP layer should be used in which case the Hybrid mode is selected by default.

In the event that a partial restoration happens, your site will be left in a semi-restored state. Trying to access it will pop up the restoration script (ANGIE). If you want to retry the restoration using different settings, please remove the `installation` directory from your site's root manually, for example using FTP, before trying to access your site's administrator back-end.

If you chose to use the FTP mode, there are some connection settings you have to take care of. Do note that they are filled in with Joomla!'s FTP layer settings by default. Unless you chose not to store your FTP password in Joomla!'s configuration or if you have not configured the FTP layer yet, there is no need to change them. The settings are:

Host name — The host name of your site's FTP server, without the protocol. For example, `ftp.example.com` is valid, `ftp://ftp.example.com` is *invalid*.

Port — The TCP/IP port of your site's FTP server. The default and standard value is 21. Please only use a different setting if your host explicitly specifies a non-standard port.

User name — The username used to connect to the FTP server.

Password — The password used to connect to the FTP server.

Initial directory — The FTP directory to your web site's root. This *is not the same as the filesystem directory* and can't be determined automatically. The easiest way to determine it is to connect to your site using your favourite FTP client, such as FileZilla. Navigate inside your web site's root directory. You'll know you are there when you see the file `configuration.php` and directories such as `administrator`, `components`, `language` in that directory. Copy (in FileZilla it appears on the right hand column, above the directory tree) and paste that path in Akeeba Backup's setting.

Test FTP connection — Clicking on this button will tell you if the FTP connection could be established or not. If the connection is not successful you should not proceed with a restoration in FTP mode as it will fail immediately.

The whole process is fully automated, so there is not much to tell you about it. However, you must not that in order for the restoration procedure to work properly you must take care of the following:

1. This feature is directly calling the `administrator/components/com_akeeba/restore.php` script. If you have a server-side protection, i.e. .htaccess rules, or permissions settings which prevent this file from being called directly the process will fail.

   Security note: The restore.php file is of no use to potential hackers. In order for it to work at all, it requires the `restoration.php` file (more on that on the next point of this list) to load. Even then, it expects a key which is not predefined and is only known to the `restore.php` script and the integrated restoration page of Akeeba Backup. As a result, it can't be used as a potential attack vector.

2. Before the restoration begins, Akeeba Backup needs to create the `administrator/components/com_akeeba/restoration.php` file with all the archive extraction setup parameters. It is intelligent enough to use Joomla!'s FTP mode if it is enabled so as to overcome any permission problems, but you are ultimately responsible for ensuring that the permission settings are adequate for Akeeba Backup to create this file.

   If you have disabled Joomla!'s FTP layer, the permissions of the `administrator/components/com_akeeba` directory should be 0777 for the integrated restoration to work, or 0755 on hosts which use suPHP. You can change these permissions after the restoration is over, of course.

   If you are using Joomla!'s FTP layer and it was active when you were installing Akeeba Backup, you'll need to give this directory at least 0755 permissions, but you may have to manually remove `restoration.php` (**but NOT** `restore.php`!!!) after the site restoration is over.

3. When the extraction of the backup archive finishes, you will be automatically forwarded to the Akeeba Backup Installer page on a new tab or window. DO NOT CLOSE THE INTEGRATED RESTORATION PAGE'S TAB/WINDOW! After you have competed the Akeeba Backup Installer process you are supposed to return to the Integrated Restoration page and click on the Finalize button to:

   - remove the `installation` directory from your site's root, and
   - remove the `administrator/components/com_akeeba/restoration.php` setup file to nullify the, already non-existent, potential risk of a malicious user abusing this script.

3.5.2. Manage remotely stored files

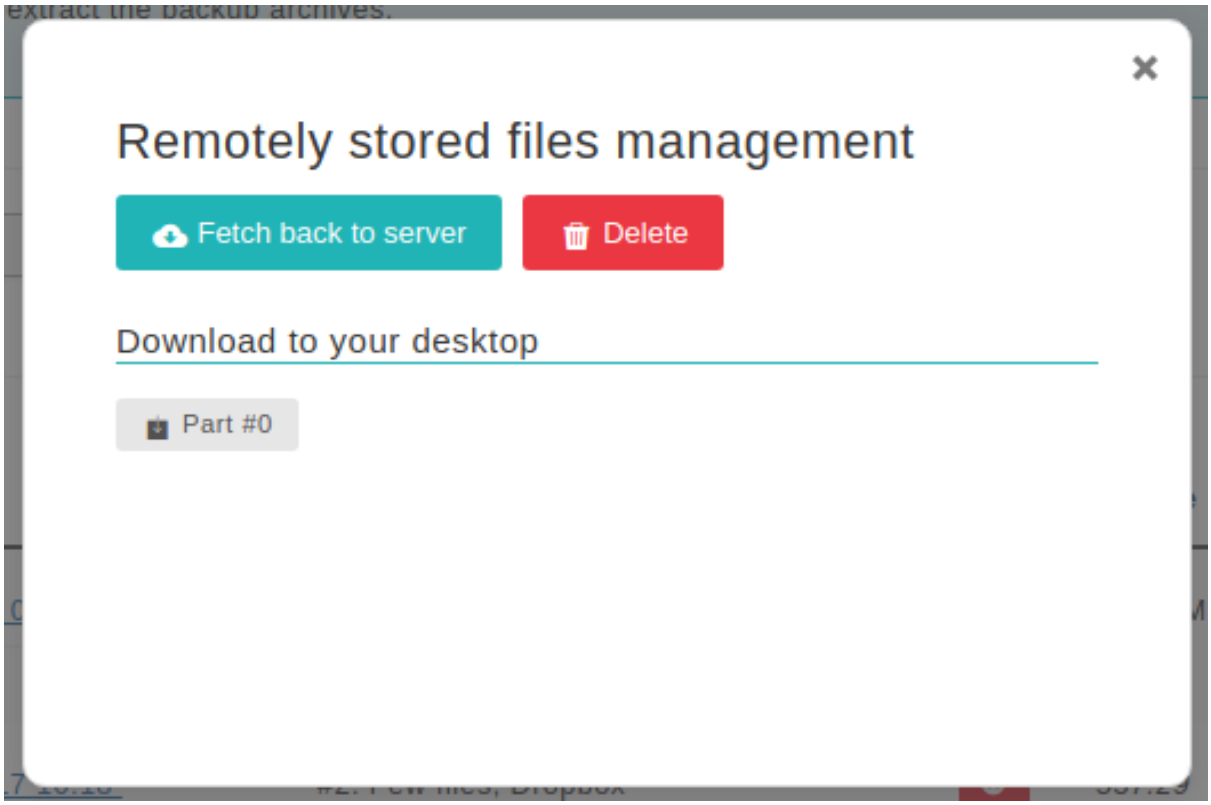
Note

This feature is only available in the Akeeba Backup Professional edition

Since Akeeba Backup 3.2 you have the option to manage backup archives stored in a remote storage location, for example Amazon S3 or a remote FTP server. You can do that by clicking on the Manage remotely stored files link on the far right of supported backup records in the Manage Backups (formerly "Administer Backup Files") page. Clicking on that link opens a modal dialog with the options compatible with your backup archive.

Please note that not all of the following features may appear in the dialog. It depends on the remote storage engine used for the backup record. All options currently appear only for files stored on Amazon S3 and remote FTP.

The "Manage Remotely Stored Files" page



The Fetch back to server button will automatically download the backup archive from the remote location and store it again on your server. This allows you to easily import backup archives stored on a remote location back to your server's storage so that you can easily restore them on the same or a different site. If you are using S3, please make sure that the user credentials you have supplied have enough privileges for the files to be downloaded (i.e. they don't grant write-only access to the bucket). Also make sure that you have adequate free disk space on your server for the operation to complete.

The Delete button will permanently delete the archives from the remote storage. There is no confirmation. Once you click this button, your remotely stored files will be removed.

Finally, there are links under the Download to your desktop header. Clicking on them will instruct your browser to download the respective backup archive's part directly to your PC. Currently, only Amazon S3, CloudFiles, Dropbox and remote FTP support this feature. Do note that the backup archives are transferred directly from the remote storage to your PC. They are not stored to your site's server. If you want to store them to your server, use the Fetch back to server button instead.

If none of the above options are available, Akeeba Backup will display an error message. In that case, just close the modal dialog.

After finishing your remote files administration, please close the modal dialog by clicking on the X button on its top-right corner and *reload the Manage Backups (formerly "Administer Backup Files") page*. Until you reload the page the changes you made WILL NOT be visible. This is not a bug, it is the way it is meant to be.

3.5.3. Discover and import archives

Note

This feature is only available in the Akeeba Backup Professional edition

Sometimes you may have accidentally deleted a backup record from the Manage Backups (formerly "Administer Backup Files") page, or simply want to restore a backup file taken from another site. Normally, the only way to do

that is to upload the archive file and Kickstart to your site and launch the restoration process from there. However, some users insisted that they are better off doing that from inside Akeeba Backup itself. In order to accommodate for their needs, we introduced the Discover and Import Archives features in Akeeba Backup 3.2.

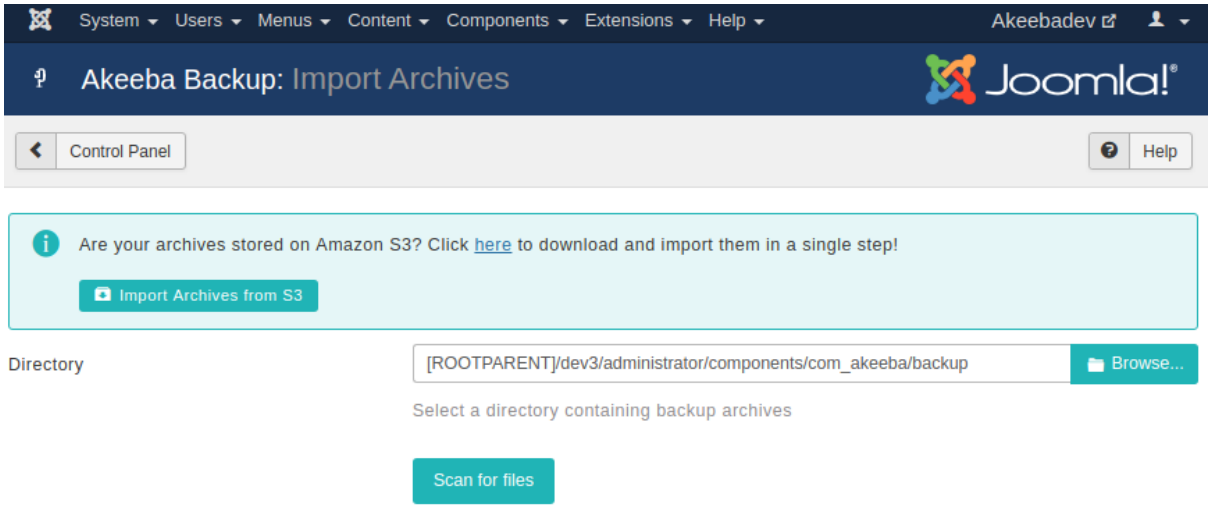
This feature allows you to automatically find and import archives stored anywhere on your account. This means that you can upload backup archives anywhere in your site's folder structure, or even on a private off-site directory and Akeeba Backup will be able to import them. All backup archives are imported as backup records of the default backup profile (profile with ID #1) and can be restored just like any other backup archive.

In order to launch this feature, go to the Manage Backups (formerly "Administer Backup Files") page and click on the Discover and import archives button on the toolbar. A new page appears which lets you select a directory.

Tip

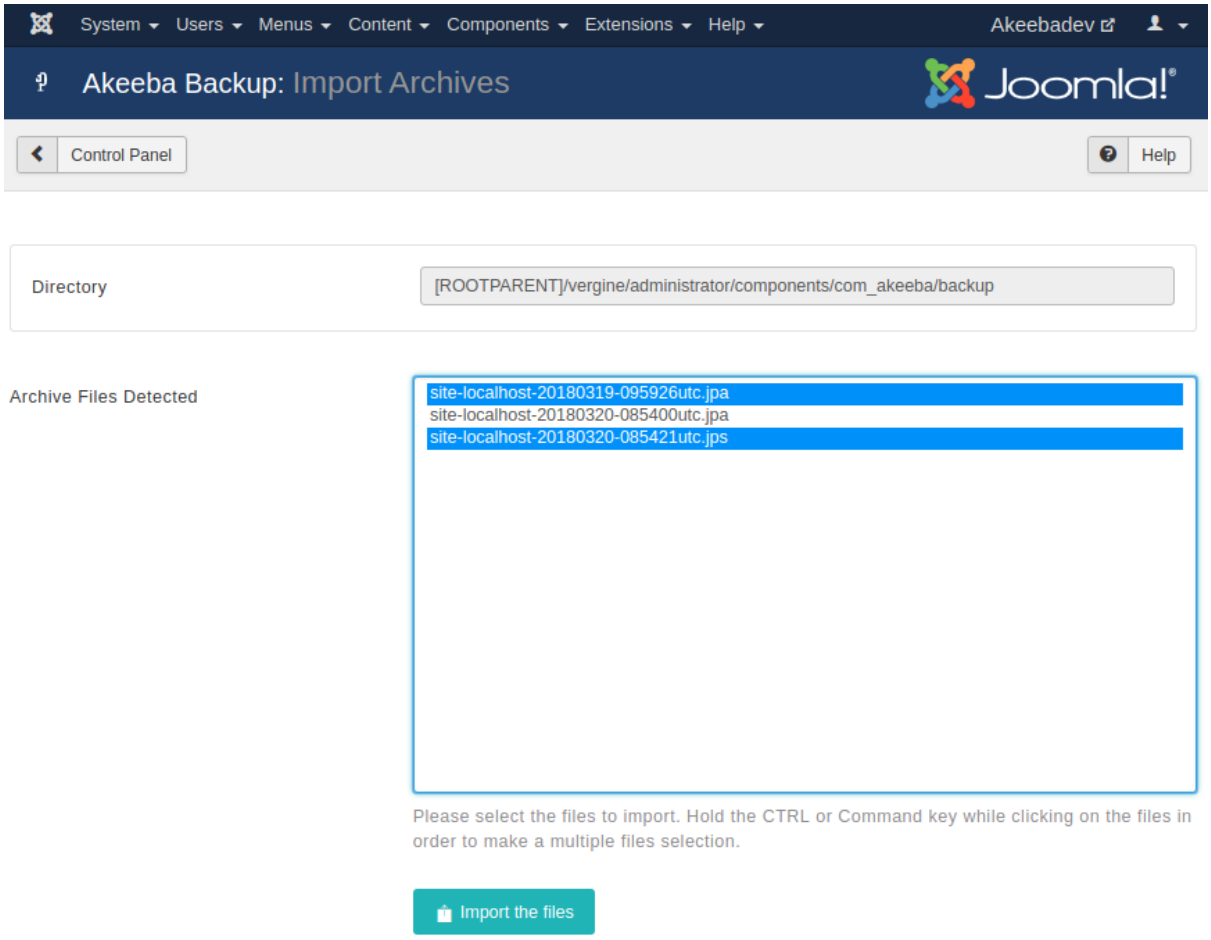
Since Akeeba Backup 3.4.a1 you have the option to import archives from Amazon S3. Click the link directly above the directory selection box. It will take you to a slightly different page where you can enter the connection credentials to your S3 account and allow you to browse for ZIP and JPA files to import.

The "Discover and Import archives" page



Use the Browse... button to open an interactive folder browser in a modal dialog. Navigate to the directory which contains the uploaded backup archives and click on the Use button. The dialog closes and you can now click on the Scan for files button to let Akeeba Backup search for backup archives inside that directory. You are presented with a new page, listing the discovered backup archives.

Importing discovered archives

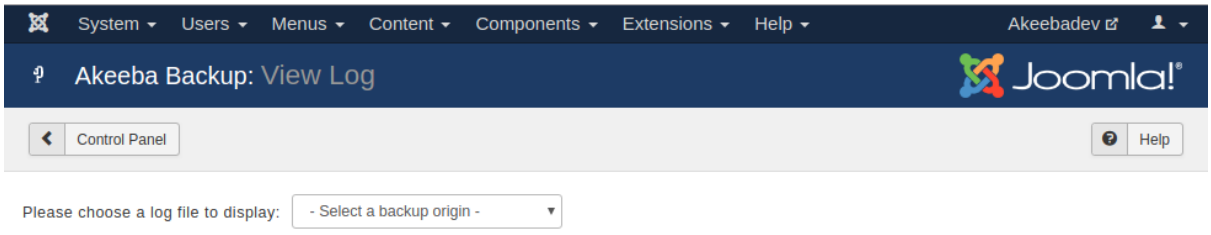


Select the backup archive you want to import by clicking on them. If you want to select multiple files, Control-click (Windows, Linux) or Command-click (Mac OS X) the archive you want to import. After that, click on the Import the files button. After a short while Akeeba Backup takes you back to the Control Panel page with a message that the import operation completed successfully. You can now click on the Manage Backups (formerly "Administer Backup Files") button to view the newly imported backup archives. You can now download or restore the imported backup archives.

3.6. View Log

The View Log option allows you to download or view the log from a recent backup operation. This information may be useful in diagnosing problems if you are having a problem completing a backup.

Selecting an origin

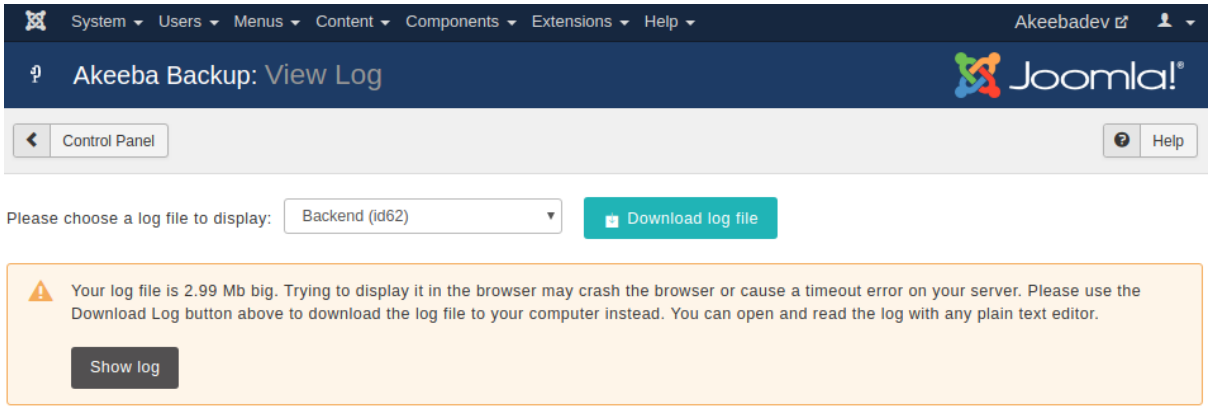


The first page allows you to select a log file to display. Each option in the drop-down is a backup origin followed by the backup ID in parentheses. If you are not sure which backup is the one you're looking for you can always

go to the Manage Backups page, find the backup there and click the View Log button next to it. This way you will bypass this selection drop-down.

This takes you to the View Log visualization page.

View Log



If you wish to ask for support, you must download the raw log (a text file). Just click on the Download Log File button. Please do not copy and paste the text appearing in the log viewer. We actually need the log file to help you.

Warning

When asking for support, make sure that the Log Level was set to "All Information and Debug" in the Basic section of the Configuration page *before* backing up. Otherwise the log will be of no use.

If the backup log file is too big you will see warning about it and you will need to click on the Show log button to display the log file. Once you do that the message will be replaced by the log viewer areas. Each line is preceded by a time stamp, in the format YYMMDD hh:mm:ss (that's year, month, date with two digits, a space and time in 24-hour format). The time stamp is in the GMT timezone. Each line is colour coded, for your convenience. Debug information is in smaller, grey type. Normal information is in black type. Warnings appear in bold yellow letters. It is important to read them as they convey information about skipped directories or other things that will be missing from the backup archive. If any errors occurred, these appear in bold red type.

Whenever you report bugs, all of the information in the log is absolutely necessary. In order to reveal as little sensitive information as possible, whenever a file path has to be logged, your site's root folder is replaced with the string '<root>'. Keep this in mind when reading warnings and errors.

Tip

If you have a failed backup but do not understand what the log file tells you, Akeeba Backup can try reading it and tell you what it thinks is going on. Just go to the Control Panel page, click on Troubleshooter - ALICE and select the log file with the same ID as the one you were viewing in the View Log page. Akeeba Backup will figure out what the log means and give you a better idea of what is going on. Please note that you should only do that for failed backups. Trying to process successful backups will confuse the troubleshooter and the response you get will be useless and invalid.

4. Include data to the backup

Note

This feature is available only in Akeeba Backup Professional.

By default, Akeeba Backup automatically includes the whole database of your Joomla!™ installation as well as all the files under your site's root in the backup set. Sometimes you want to include a different database or files

you have placed above your site's root for increased security. Akeeba Backup Professional can cope with that need by providing you with handy data inclusion filters.

4.1. Multiple Databases Definitions

Note

This feature is available only in Akeeba Backup Professional

Sometimes your site may use more than one databases. Taking a full site backup requires backing up those external databases too. Normally, Akeeba Backup only backs up the tables from your Joomla! database. The solution to this problem is the Multiple databases definitions option of Akeeba Backup. You can define an unlimited number of additional MySQL databases which will get to be backed up (and restored) along with your regular Joomla! database.

Please note that you do not need to and, in fact, must not use this feature to add the main database of your site (the one used by Joomla! itself). The main database of your site is backed up automatically. If you ignore this warning and add your main database as an additional database in this page you **will cause errors during the restoration of your site**.

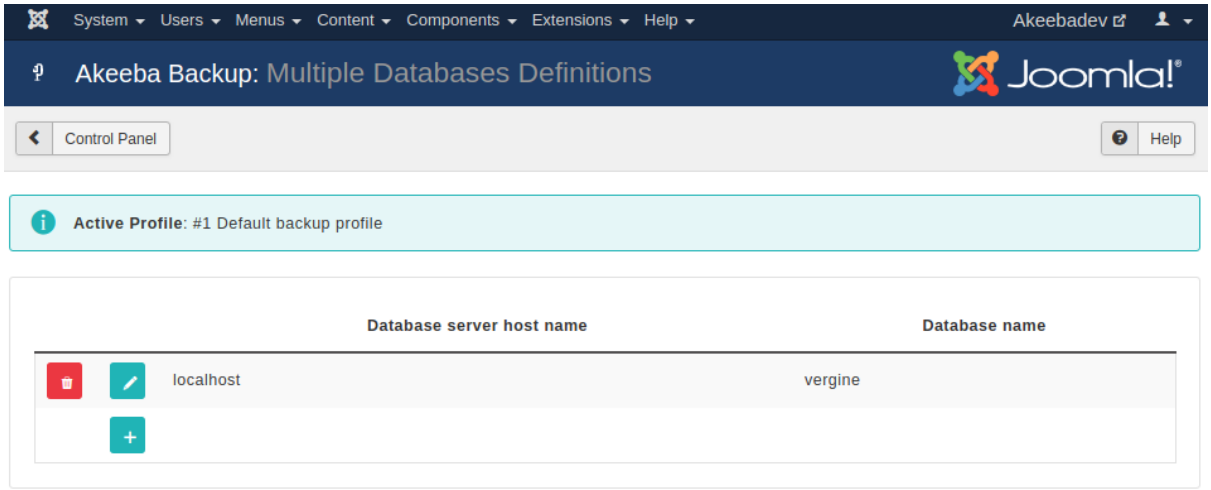
Moreover, you should not confuse the term "database" with your Joomla!™ tables. It is possible that a single *database* contains tables for the current Joomla!™ site, tables from a third party script, tables from another Joomla!™ site on the same server (e.g. a subdomain) and so on and so forth. As far as Akeeba Backup is concerned, all of those tables exist **in the same database** regardless of their *prefix*. Unless you tell it otherwise, it will backup ALL tables of the database.

Finally note that if you add an empty database (one which has no tables) it will result in backup errors!

Note

The settings on this page are defined *per profile* . Make sure you have selected the desired profile in the Control Panel page.

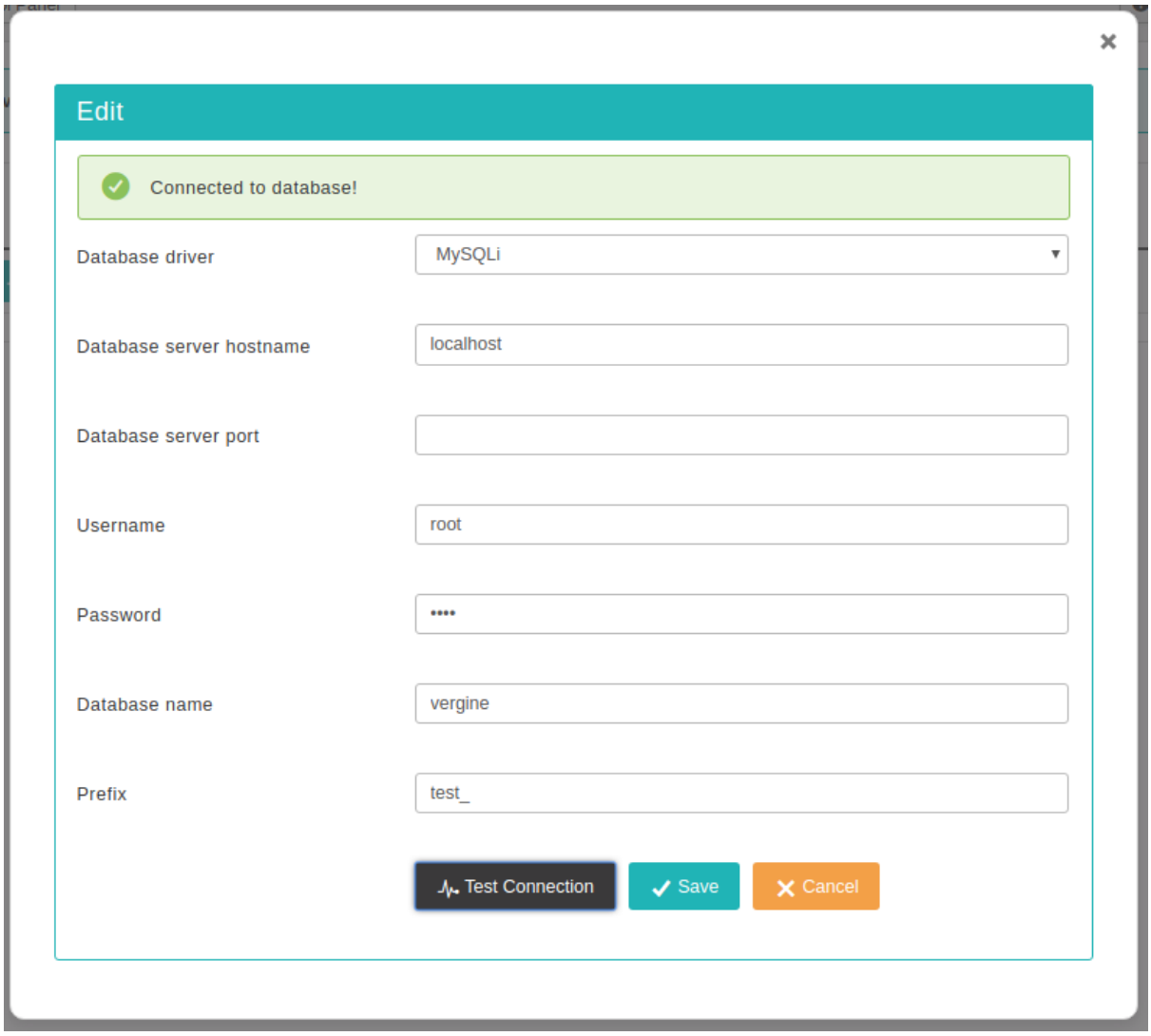
Multiple Databases Definitions



At first, you are presented with a grid view, listing all database definitions. On the left of each entry, there are two icons:

- **The trashcan**. Clicking on this icon will remove the current database definition from the backup set.
- **Pencil** or **Add**. Both will open the database definition editor: the former to edit the database definition, the latter to create a new one.

Multiple Databases Definitions - The editor



The database definition editor opens as a dialog box inside the multiple databases definitions page. The options you can select for each database are:

- **Database driver**. You can select which database driver Akeeba Backup will use to connect to the database. Your options are:
  - **MySQLi**. This is the modern MySQL connection driver. We recommend using it for MySQL, Percona and MariaDB databases.
  - **MySQL**. This is the old, obsolete MySQL connection driver for PHP. It has bad performance and has been removed in PHP 7.
  - **MySQL (PDO)**. This is the second modern MySQL connection driver.
- **Database server hostname**. The host of your database server. Usually it's `localhost`, but many hosts use something different. If in doubt, ask your host. Please remember that for MySQL servers the settings `localhost` and `127.0.0.1` are NOT the same. The first means "connect using a socket or named pipe" the second means "connect using TCP/IP networking". If you are on Windows they have a massive performance difference as well.
- **Database server port**. Leave it blank, unless your host has told you to use a non standard port for connecting to their database server.

- **Username**. The username of the database user needed to connect to the database.
- **Password**. The password of the database user needed to connect to the database.
- **Database name**. The name of the database you are connecting to.
- **Prefix**. The prefix used in the table name's prefixes. **MAJOR PITFALL**: Please do not leave the Prefix field blank if you intend to use the Database Table Exclusion feature to exclude tables or table data of this extra database from the backup. If you don't want to use a real prefix, please use a "fake" prefix, e.g. `thisIsAFakePrefix_` , to keep the Database Table Exclusion feature happy and functional.

Some hosts use your account name as a prefix for the database and username. **This is not the same as the Prefix setting above.** That database and username prefix is actually part of the actual database and username that you need to fill into this page. For example, you're hosted under the account name `foobar` and you create a database `mydata` and a user `myuser`. Your host displays a prefix `foobar_` on the left of the edit boxes where you entered the database and user names. This means that your REAL database name is `foobar_mydata` and your real username is `foobar_myuser`. This is especially true for accounts hosted in cPanel and Plesk powered hosts. It goes without saying that your password does NOT take a prefix. If in doubt, please contact your host. We can't guess the right values for you because we are not your host. If you ask your host to give you the connection information to your database, they must be able to do so - except for the password which they obviously cannot see for security reasons.

When you think you have all the connection information ready, click on Test Connection. This will check all settings except the Prefix. The connection test will tell you if it succeeded or failed.

If your connection works properly, it's time to save your changes by clicking the Save & Close button. The top panel will briefly display a "loading" message and the dialog box will go away. That was it, your extra database definition is now saved.

4.2. Off-site Directories Inclusion

Note

This feature is available only in Akeeba Backup Professional

It's very likely that advanced site owners will place files outside the site's root to prevent web visitors from having direct access to those files. These directories typically contain files that need complex access control and are, therefore, only made available for download through PHP code, e.g. a download manager extension for Joomla!. Akeeba Backup Core will only backup files under the site's root, which would make these files impossible to backup.

The solution to that problem is the Off-site Directories Inclusion feature of Akeeba Backup Professional. Using this feature you can tell Akeeba Backup to look for files in arbitrary locations outside the site's root and include them in the backup archive. All the directories included with this filter will be placed in the archive as subdirectories of another folder, in order to avoid directory name clashes. We call this parent folder the "virtual folder".

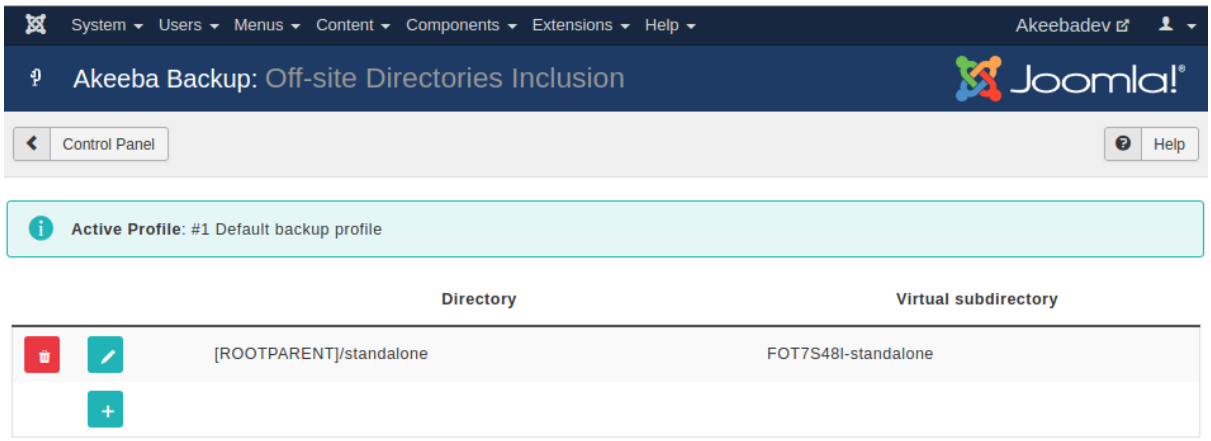
For example, let's say you want to backup an off-site directory named `images`. If we weren't using the virtual folder its contents would end up being backed up inside the Joomla! `images` directory. This is not desirable. If your virtual folder is called `my_offsite_includes` , this directory would end up being backed up as something like `my_offsite_includes\123ABC-images` . Notice the stuff and the dash before the actual directory name? This is a smart feature which allows you to backup many directories that have the same name. You could, for instance, backup two directories named `images` , confident that there would be no name clash inside the archive.

Since keeping track of these folders is a pain, Akeeba Backup includes a `readme.txt` text file inside the virtual folder which tells you which backed up folder corresponds to which physical folder, making it easy for you to restore these directories to their rightful place.

Moreover, ANGIE -the restoration script included in the Akeeba Backup archives- can semi-automatically restore the off-site directories to their original location. You will need to confirm the destination directory or, if you don't want to do this, just tell it to skip over that directory.

Finally note that **you MUST NOT add your site's root as an off-site directory inclusion**. Akeeba Backup already adds the contents of your site's root to the backup. If you manually add your site's main directory as an off-site directory inclusion you will be backing up the same files twice, doubling your backup size. For the same reason you must not add an folder already under the site's root as an off-site directory inclusion: you'd be backing up files already backed up, bloating the backup size.

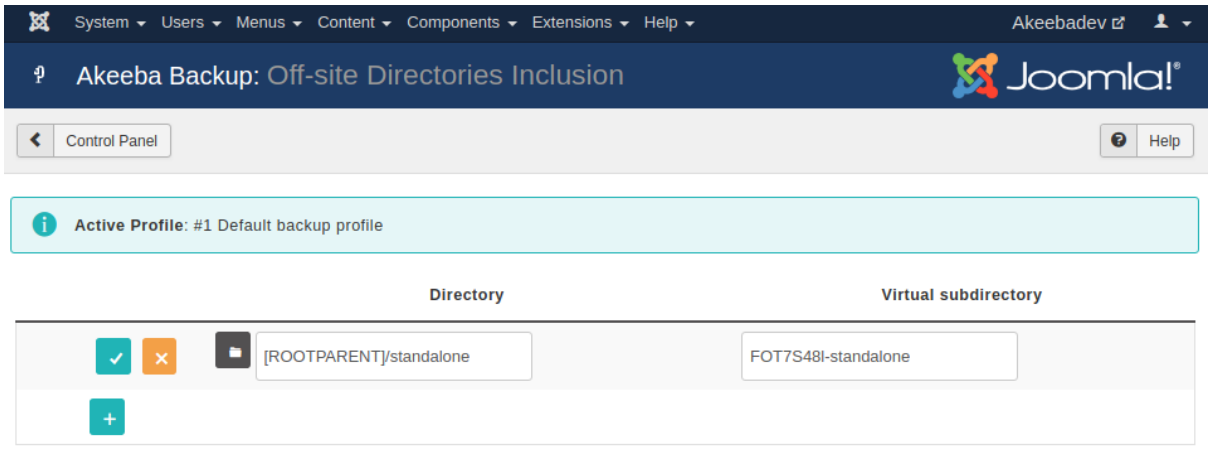
Off-site Directories Inclusion



At first you are presented with a grid view, listing all the off-site inclusions you may have already added. Next to each row and on the left hand side of it you will find two icons:

- **The trashcan**. Clicking on this icon will remove the current directory definition from the backup set.
- **Pencil** or **Add**. Both will toggle the row to edit mode: the former to edit the directory definition, the latter to create a new one.

Off-site Directories Inclusion - Edit mode



When a row enters the edit mode, the pencil icon changes to two different icons:

- **The disk**. Clicking on this icon will save any changes you have made.
- **Cancel**. Clicking it will abort any changes you have made.

You will also observe that the path to the external directory has also turned to an edit box with a folder icon on its left. You can type in the absolute path to the external directory using the edit box, or click on the folder icon

to launch a visual folder browser, much like the one you use to select an output directory in the component's Configuration page. If you choose to use the edit box, you can use the following variables:

- **[SITEROOT]** is the absolute path to your site's root. You should never use this for the reasons explained earlier in this section.

- **[ROOTPARENT]** is the absolute path to your site root's parent directory, i.e. one level above your site's root.

To the right of the directory you will see another field called Virtual Directory. This is the name of the subdirectory where Akeeba Backup stores the files and folders of these off-site directory's files. Normally, the subdirectory is placed inside the virtual directory for external files, as defined in your backup profile's configuration. If you do not enter a directory name Akeeba Backup will use a predetermined name. This name is a random value followed by a dash and the name of the off-site directory you are defining.

Sometimes you want to include off-site files directly inside the archive's root. Two very useful cases are overriding your regular configuration.php file with another one –presumably one tuned for use on your dev site– as well as overriding files in the installation directory, for example in order to customise the appearance of the installer. In those cases you don't want the off-site files to be included inside the virtual directory for off-site files. With Akeeba Backup 3.7.5 and later this is very easy to accomplish. Just set the Virtual Directory to a single forward slash (it's this character: `/`) and Akeeba Backup will copy the off-site files inside the archive's root.

5. Exclude data from the backup

More often than not you have data on your site you don't want to include in the backup set. This can be host-specific directories (e.g. `cgi-bin`, `stats`, etc), log files, temporary data, an huge but immutable collection of large media files, click tracking tables, download log database records and so forth. The exclusion filters allow you to fine tune what should be left out of the backup set.

5.1. Files and Directories Exclusion

Very often our sites have files or folders which don't really belong to the backup. A few examples are:

- Additional sites whose root folders are subdirectories of your site's root. As explained elsewhere in this documentation, backing them up and restoring them would end up overwriting more sites than you bargained for.

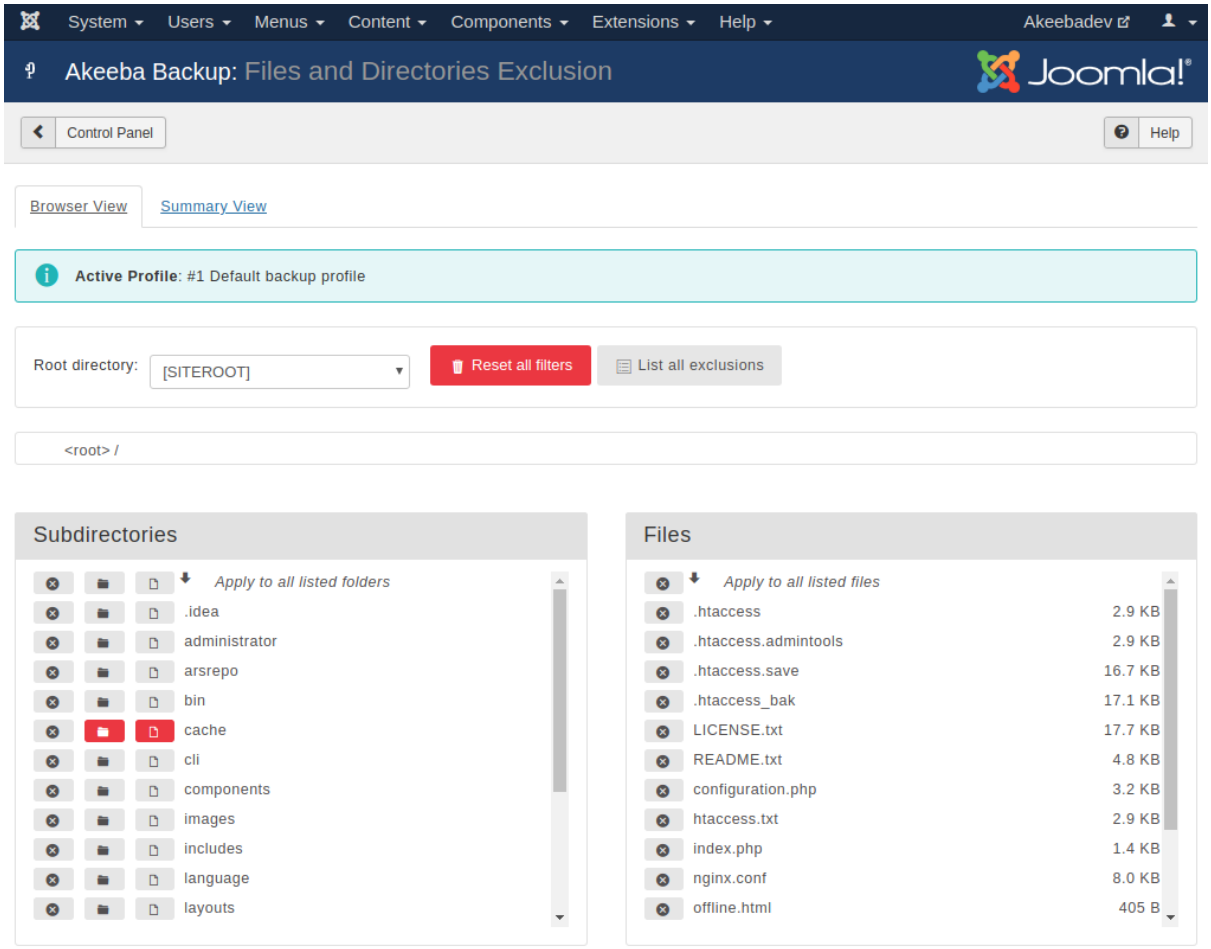
- Directories with large amounts of videos, images, download repositories or other infrequently changing files. In most cases it makes sense to exclude them from your daily backups and only include them in a separate weekly or monthly backup profile.

- Leftover files you had forgotten about until the time came to back up your site. For example, that really big ZIP file with the previous version of your site you meant to delete two years ago.

Akeeba Backup lets you exclude files and folders to solve these problems.

Before discussing this feature, you should be aware of some automatic file and folder exclusions applied by Akeeba Backup. Akeeba Backup will automatically exclude your site's temp-folder and logs folder as configured in your site's Global Configuration; the "cache" directories under your site's root and administrator directory; and all files and directories inside the Akeeba Backup's output directory. This means that you should **never use a folder whose contents you intend to back up as your backup output directory, your site's temp-folder or your site's logs folder**. Moreover, do not leave the temp-folder and / or log folder blank or set them to your site's root in your site's Global Configuration. Doing so will result in your backup archive NOT having any of your site's files since the site's root will be automatically excluded by the automatic filters as explained above.

Files and Directories Exclusion - Browser View



At the top of the page there are two tabs, allowing you to switch between the Browser and Summary views.

The middle area contains a few controls you need to know about:

- The Root Directory drop-down menu. Akeeba Backup can define filters for the site's files or for each of the off-site directories separately. The default selection, [SITEROOT], contains all filters pertaining to the main site's files. If you have defined off-site directories, you can select the appropriate directory from the drop-down list in order to define filters for that directory.
- The Reset all filters button will, as the name implies, remove all of the file and directory exclusion filters *for the selected root directory*.
- The List all exclusions button takes you to the Summary View page.
- The Current directory bread crumb list. It shows the current path relative to the Root directory above. Clicking on a subdirectory allows you to quickly navigate to it.

The lower area consists of two panes, showing the folders and files in the current directory. The icons next to each item are an exclusion type each. You can use them to enable / disable filters on each folder or file. The top row of each panes has controls (icons) which apply the filters to all of the listed folders or files below it.

Each icon can have three states: on (yellow background), off (gray background), or force enabled (red background). You can toggle between the on and off states by clicking on the icon. The force enabled state means that this exclusion type is active (on) and forcibly enabled by another feature of Akeeba Backup, such as the automatic exclusions discussed above, the regular expressions filters or a programmatic filter (plug-in) by a third-party developer. Force enabled filters cannot be changed through this page.

The available filters for directories are:

- **Exclude Directory**. When enabled, the folder and all of its contents (subdirectories and files) will not be included in the backup. This filter overrides the Skip Subdirectories and Skip Files filters.
- **Skip subdirectories**. When enabled, the subdirectories of this directory will not be included in the backup. However, the directory itself and its files will be included in the backup.
- **Skip files**. When enabled, the files inside this directory will not be included in the backup. However, the directory itself and its folders (and the files inside these folders) will be included in the backup.

If both Skip Subdirectories and Skip Files filters are enabled on a folder then an empty folder will be included in the backup. If you do not want the folder to be included *at all* use the Exclude Directory filter.

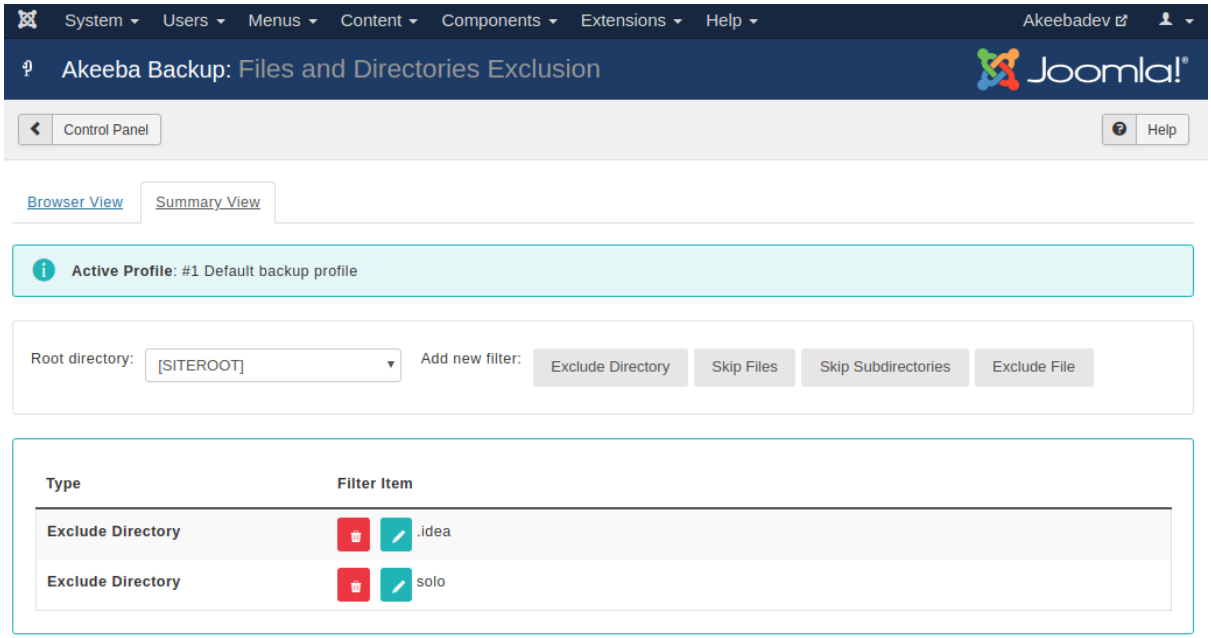
Clicking on a folder name in the Folders pane will navigate inside it.

The available filters for files are:

- **Exclude File**. When enabled, the file will not be included in the backup.
- The file name.

Each file name displays its size to the right. The file size will be displayed in the unit which is more convenient, i.e. bytes, KB, MB or GB. If you see no unit of measurement, the size is displayed in bytes.

Files and Directories Exclusion - Summary View



The Summary View displays a list of filters instead of a directory browser.

At the top you have the Root Directory drop-down menu. Akeeba Backup can define filters for the site's files or for each of the off-site directories separately. The default selection, [SITEROOT], contains all filters pertaining to the main site's files. If you have defined off-site directories, you can select the appropriate directory from the drop-down list in order to define filters for that directory.

On the top side of the grid you have the Add new filter buttons:

- **Exclude Directory**. The named folder and all of its contents (subdirectories and files) will not be included in the backup. This filter overrides the Skip Subdirectories and Skip Files filters.
- **Skip Subdirectories**. The subdirectories of the named folder will not be included in the backup. However, the folder itself and its files will be included in the backup.

- **Skip Files**. The files inside the named folder will not be included in the backup. However, the folder itself and its subfolders (and the files inside these folders) will be included in the backup
- **Exclude File**. The file will not be included in the backup.

The same notes regarding use of folder filters described in the Browser View apply.

Each line of the grid displays the following information:

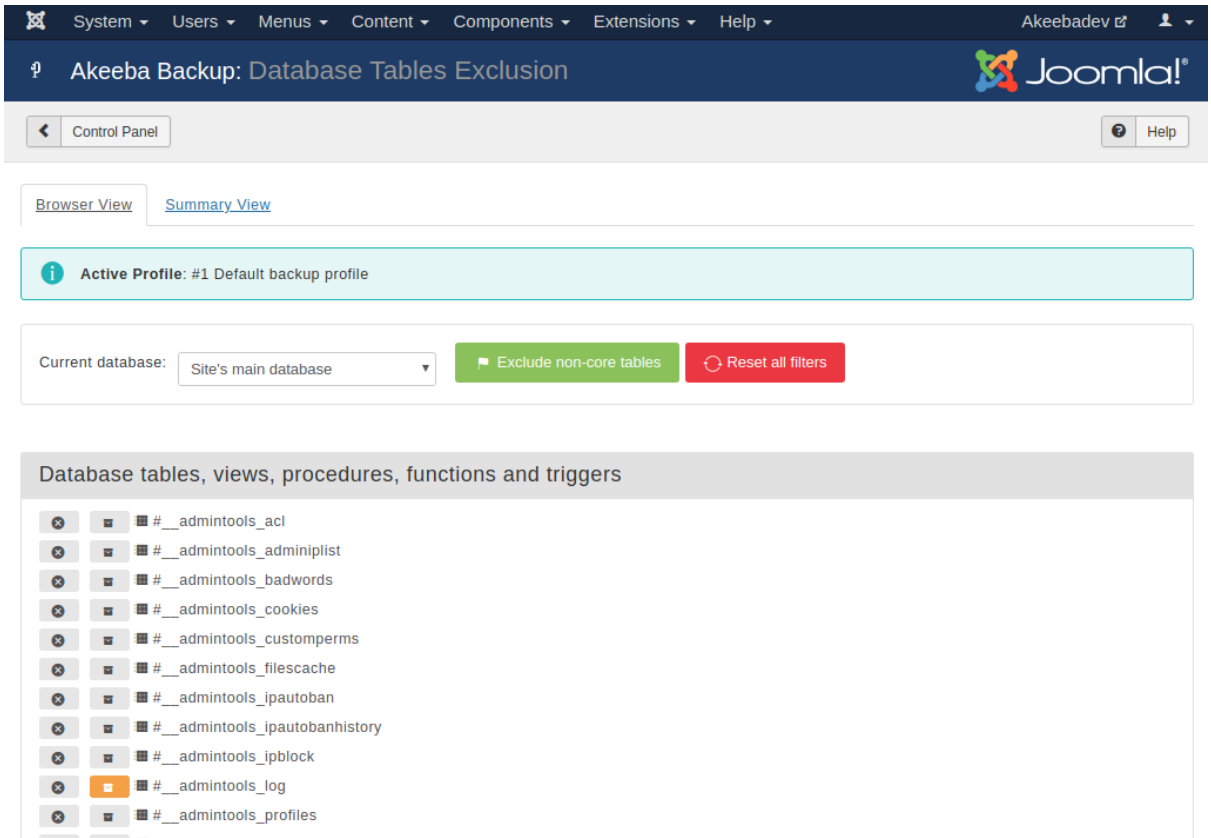
- **The filter type**. As described above in the add new filter buttons.
- **Trashcan**. When you click it, the filter row will be removed.
- **Pencil**. When you click it, the row switches to edit mode
- The **filter item** itself. It is the relative path to the directory or file which the filter row applies to. The path is relative to the Root directory displayed on the selection box on top.

When you click on the pencil icon, the filter item becomes an edit box. You can type in the new relative path and then click outside the edit box, or press Tab on your keyboard, to immediately save the changes.

5.2. Database Tables Exclusion

There are cases where you need to exclude either entire database tables or their contents from the backup. For example, if you are using a single database for the tables of two or more sites you will want to exclude all tables not belonging to the site you're backing up to prevent accidental overwriting of the wrong site when restoring the backup. Moreover, if you have large tables with not very important data, such as log entries, you may want to exclude their contents -but not the entire table- from the backup for performance reasons. This is what the Database tables exclusion feature lets you do.

Database Tables Exclusion - Browser View



At the very top of the page you can see two tabs which let you switch between the Browser View and the Summary View.

Below the tabs you will see the backup profile number and title as a reminder. Remember that database table exclusion filters, just like all Akeeba Backup filters, are set up per backup profile.

Further down there is the Current Database drop-down list. Akeeba Backup can define filters for the site's main database or for each of the extra database definitions separately. The default selection, Site's main database, contains all filters pertaining to the main site's database, i.e. the one your Joomla!™ site runs on. If you have defined additional databases you can select the appropriate database from the drop-down list to define filters for that database.

There are another two buttons here. The **Exclude non-core tables** button. Clicking it will automatically apply the Exclude This filte on all tables whose name doesn't begin with your site's prefix. These are usually tables which do not belong to the current Joomla! installation. Be warned of a major pitfall: the effects of this button are static. That is to say, if new tables with a different prefix are added in the future (e.g. tables are added in the other sites using the same database) you will have to come back here and click on this button again. Instead of that and if you have the Akeeba Backup Professional version you can use the Regular Expressions Database Tables feature to automatically deal with such configurations, without having to click this button.

The **Reset all filters** button will remove all database table filters for the currently selected database.

The main area of the page displays the contents of the database: tables, views, triggers, stored procedures and functions. Each row represents one database entity. The two leftmost icons represent an exclusion type, explained below. The third icon tells you what kind of database entity (table, view, trigger, ...) it is; hover over it to find out.

Each of the exclusion type icons may have one of three states: on (yellow background), off (gray background), or force enabled (red background). You can toggle between the on and off states by clicking on the icon. The force enabled state cannot be changed. It means that it is active (on) because another feature of Akeeba Backup, such as regular expressions, have it turned on. In case of non-table database entities the red filter type means that this operation is not applicable to this entity. For example, there is no point excluding the contents of a view since only its structure is being backed up anyway.

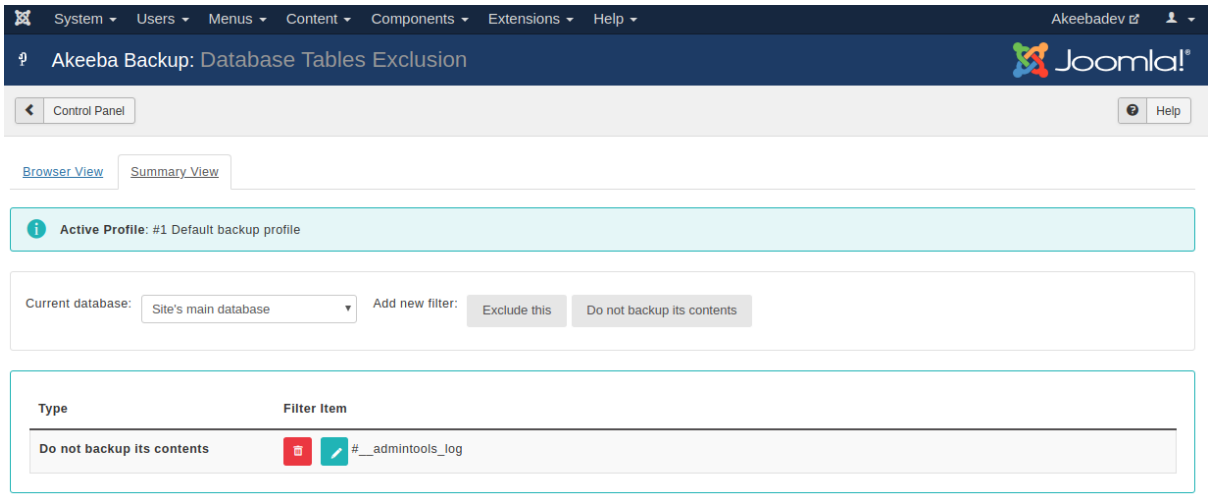
Important

The prefixes of the entities' names appear "abstracted". If your site's prefix is `abc1_` , the table `abc1_users` will appear as `#__users`. This helps you quickly identify the tables your site runs on.

The available filters are:

- **Exclude This**. This database entity will not be backed up at all.
- **Do not backup its contents**. Only the structure of the database entity will be backed up, but not its contents. When restoring, this table will be created empty.

Database Tables Exclusion - Summary View



The Summary View displays a list of all active filters, allowing to quickly modify them.

At the very top of the page you can see two tabs which let you switch between the Browser View and the Summary View.

Below the tabs you will see the backup profile number and title as a reminder. Remember that database table exclusion filters, just like all Akeeba Backup filters, are set up per backup profile.

Further down there is the Current Database drop-down list. Akeeba Backup can define filters for the site's main database or for each of the extra database definitions separately. The default selection, Site's main database, contains all filters pertaining to the main site's database, i.e. the one your Joomla!™ site runs on. If you have defined additional databases you can select the appropriate database from the drop-down list to define filters for that database.

Above the grid you have the Add new filter buttons. The filter types correspond to the icons in the Browser View, as discussed further above.

Each line of the grid displays the following information:

- **The filter type**. As discussed above.
- **Trashcan**. When you click it, the filter will be removed.
- **Pencil**. When you click it, the row switches to edit mode
- The **filter item** itself. It is the abstracted database entity name which the filter row applies to. When we say "abstracted" we mean that the site's prefix has to be replaced by #__ as discussed above.

When you click on the pencil icon, the filter item becomes an edit box. You can type in the new (abstracted) database entity name and then click outside the edit box, or press Tab on your keyboard, to immediately save the changes. There is no way to undo your changes.

5.3. RegEx Files and Directories Exclusion

Note

This feature is available only in Akeeba Backup Professional

Sometimes you know that you have to exclude files or directories following a specific naming pattern, but they are so many that it's impractical going to the normal exclusion filters page and click them one by one. Or they are scattered around the file system tree, making it too complicated tracking them down and excluding them one by

one. Regular expression filters let you create pattern-based filters to deal with that. What are regular expressions? Let's consult Wikipedia:

> In computing, regular expressions, also referred to as regex or regexp, provide a concise and flexible means for matching strings of text, such as particular characters, words, or patterns of characters. A regular expression is written in a formal language that can be interpreted by a regular expression processor, a program that either serves as a parser generator or examines text and identifies parts that match the provided specification.
>
> —"Regular expression" article [http://en.wikipedia.org/wiki/Regular_expression] from Wikipedia

In a nutshell, regular expressions allow you to quickly define filters which span multiple subdirectories and match file or directory names based on a number of criteria. If you want a quick cheatsheet you can use, we recommend taking a look at the Regular Expressions Cheat Sheet (V2) [https://www.cheatography.com/davechild/cheat-sheets/regular-expressions/] from Cheatography. Some practical examples will be presented at the end of this section.

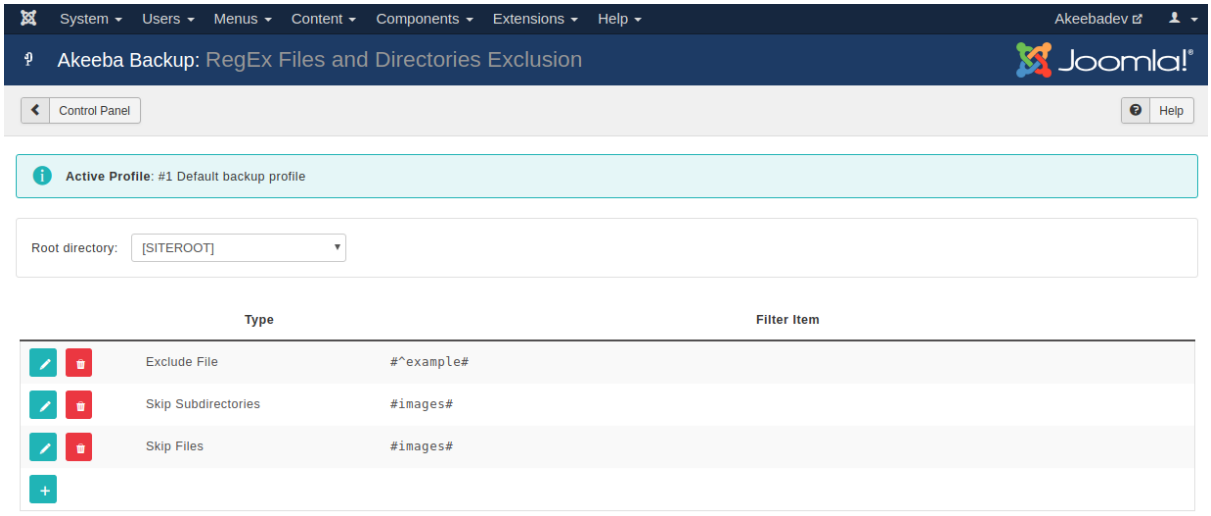
There are some special considerations experienced regular expressions users must have in mind:

- You are supposed to specify a full regular expression, including its opening and ending separators. So `^foo` is invalid, but `/^foo/` and `#^foo#` are valid.
- Akeeba Backup supports an extension to the PCRE syntax. If you prefix the regex with an exclamation mark you negate its meaning. So `/^foo/` will match all entities starting with *`foo`*, whereas `!/^foo/` will match all entities NOT starting with *`foo`*.
- Akeeba Backup stores and parses your data as raw Unicode (UTF-8), provided that your database meets the minimum requirement of site database server version. This eliminates the need to use the `u` suffix of regular expressions in order to reference Unicode characters.

When it comes to files and directories exclusion filters in particular, you have to bear in mind:

- The path separator is always the forward slash, even on Windows. This means that `c:\wamp\www\index.php` is internally represented as `c:/wamp/www/index.php`. Therefore, all regular expressions must use the forward slash whenever referencing a path separator.
- The filenames are always relative to the root. That's why you have to select a root before entering a regex filter. For instance, the images/stories directory on the root of your Joomla!™ site is internally referenced as `images/stories`. You have to take this into account when writing regular expressions.

RegEx Files and Directories Exclusion



At the very top of the page you will see the backup profile number and title as a reminder. Remember that database table exclusion filters, just like all Akeeba Backup filters, are set up per backup profile.

Right below it is the Root Directory drop-down menu. Akeeba Backup can define filters for the site's files or for each of the off-site directories separately. The default selection, [SITEROOT], contains all filters pertaining to the main site's files. If you have defined off-site directories, you can select the appropriate directory from the drop-down list in order to define filters for that directory.

Each row in the grid below represents a filter. The three columns on each row are:

Icons column
: You can perform the basic operation by clicking on this column's icons:

  - **Trashcan**. When you click it, the filter row will be removed.
  - **Pencil**. When you click it, the row switches to edit mode
  - **Add** (only on the last row). Clicking this icon adds a new row at the end of the list and switches it to edit mode. You can select the type of the newly added filter.

Type
: The filter type defines what will happen when a directory or file matches the regex filter and can be one of:

  - **Exclude directory**. Completely skips backing up the given subdirectory.
  - **Exclude file**. Completely skips backing up the given file.
  - **Skip subdirectories**. Skips backing up all the subdirectories inside the given directory.
  - **Skip files**. Skips backing up all the files inside the given directory.

Filter Item
: This is the actual regular expression you have to write.

RegEx Files and Directories Exclusion - Edit Mode

When you click on the pencil or add icons, the respective row enters the edit mode. In this mode, the filter type becomes a drop-down list where you can select the type of this filter row. The filter item column also turns into an edit box so that you can enter your filter definition. The icon column now contains two different icons:

- **Disk**. When you click it, the changes will be saved.
- **Cancel**. When you click it, any changes will be cancelled and the row will resume its previous state.

You can easily make sure that your filters match the directories and/or files you meant to. Just go back to the Control Panel and click on the Files and Directory Exclusion button. The items filtered out by the regular expressions filters will be automatically highlighted in red. You can browse through the file system structure to make sure that only the items you really meant are being excluded.

5.3.1. Regular Expressions recipes for files and directories

No matter how good you are on writing regular expressions, it's always a good idea to have some recipes which serve as a starting point for cooking your own.

1. Exclude AVI files in all directories (note: the i at the end causes the regex to match .avi, .Avi, .AVI, etc without discriminating lower or upper case):

   ```
   #\.avi$#i
   ```

2. Exclude AVI files in your site's `images` directory and all of its subdirectories:

   ```
   #^images/(.*).avi$#i
   ```

3. Exclude AVI files in your site's `images` directory but *not* its subdirectories

```
#^images/[^/]*.avi$#i
```

4. Exclude AVI files in your site's `images/video` subdirectory but *not* its subdirectories

   ```
   #^images/video/[^/]*.avi$#i
   ```

5. Exclude all files *except* for files ending in .php (note: the exclamation mark in the beginning is a custom Akeeba Backup notation which negates the meaning of the following regular expression)

   ```
   !#(?>\.php$)#
   ```

6. Exclude all `.svn` subdirectories anywhere and everywhere in your site. The idea is to match everything which ends in a slash (directory separator) and `.svn`, therefore it's a .svn subdirectory.

   ```
   #/\.svn$#
   ```

   However, this won't match the `.svn` directory in your site's root, so you will have to add yet another filter:

   ```
   #^\.svn$#
   ```

   This second filter matches only the `.svn` directory in your site's root.

5.4. RegEx Database Tables Exclusion

Note

This feature is available only in Akeeba Backup Professional

Sometimes you know that you have to exclude database tables following a specific naming pattern, but they are so many that it's impractical going to the normal exclusion filters page and click them one by one. Or, more frequently, you want to exclude database tables not following a specific pattern, e.g. tables whose name doesn't begin with your site's table naming prefix. Regular expression filters let you create pattern-based filters to deal with that. What are regular expressions? Let's consult Wikipedia:

> In computing, regular expressions, also referred to as regex or regexp, provide a concise and flexible means for matching strings of text, such as particular characters, words, or patterns of characters. A regular expression is written in a formal language that can be interpreted by a regular expression processor, a program that either serves as a parser generator or examines text and identifies parts that match the provided specification.
>
> —"Regular expression" article [http://en.wikipedia.org/wiki/Regular_expression] from Wikipedia

In a nutshell, regular expressions allow you to quickly define filters which span multiple subdirectories and match file or directory names based on a number of criteria. If you want a quick cheatsheet you can use, we recommend taking a look at the Regular Expressions Cheat Sheet (V2) [https://www.cheatography.com/davechild/cheat-sheets/regular-expressions/] from Cheatography. Some practical examples will be presented at the end of this section.

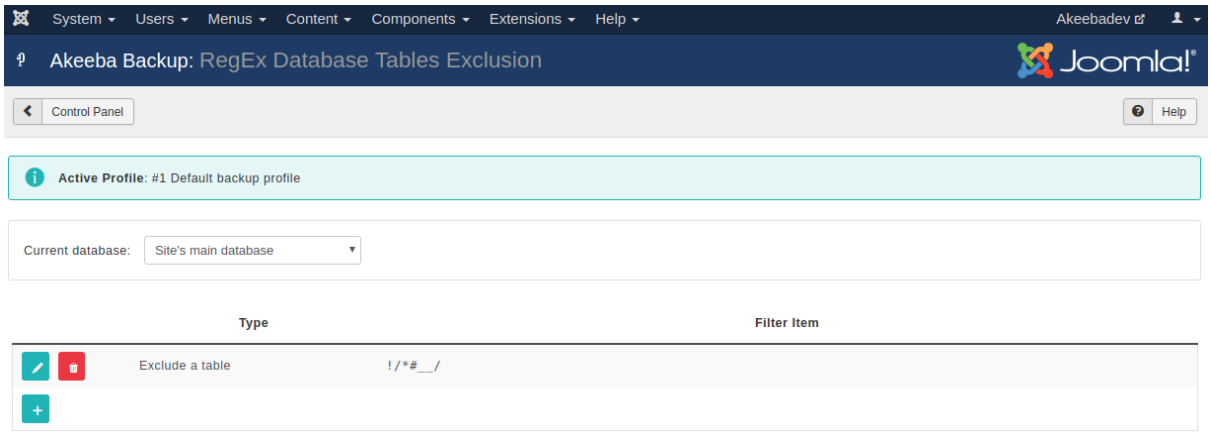
There are some special considerations experienced regular expressions users must have in mind:

- You are supposed to specify a full regular expression, including its opening and ending separators. So `^foo` is invalid, but `/^foo/` and `#^foo#` are valid.
- Akeeba Backup supports an extension to the PCRE syntax. If you prefix the regex with an exclamation mark you negate its meaning. So `/^foo/` will match all entities starting with *`foo`*, whereas `!/^foo/` will match all entities NOT starting with *`foo`*.
- Akeeba Backup stores and parses your data as raw Unicode (UTF-8), provided that your database meets the minimum requirement of site database server version. This eliminates the need to use the `u` suffix of regular expressions in order to reference Unicode characters.

When it comes to database table filters in particular, you have to bear in mind:

- All Joomla!™ tables have their prefix stripped and replaced by the standard #__ placeholder. So, if your database prefix is `abc1_`, the table `abc1_users` is internally referenced as `#__users`. This is called the "abstracted" name in Akeeba Backup's documentation. You must take this into account when writing regex filters. The abstracted name of the table is the name you will have to match with your regular expressions!
- The prefix replacement described above takes place in Full Site and All Configured Databases backup modes. However, it *does not* take place in the Database Only backup mode. As a result, you have to reference the tables by their full, normal name, e.g. `abc1_users`.
- The examples at the end of this section apply to a full site backup scenario, where the replacement does take place.

RegEx Database Tables Exclusion



At the very top of the page you will see the backup profile number and title as a reminder. Remember that database table exclusion filters, just like all Akeeba Backup filters, are set up per backup profile.

Below that you can find the Current Database drop-down menu. Akeeba Backup can define filters for the site's main database or for each of the extra databases you may have defined. The default selection, Site's main database, contains all filters pertaining to the main site's database, of course. If you have defined extra databases, you can select the appropriate database from the drop-down list in order to define filters for that database.

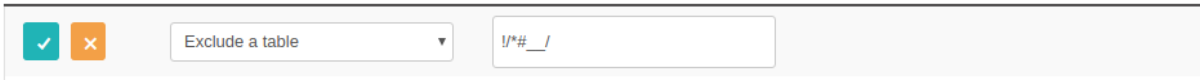
Each row represents a filter. It has three columns:

Icons column
: You can perform the basic operations by clicking on this column's icons:
  - **Trashcan**. When you click it, the filter row will be removed.
  - **Pencil**. When you click it, the row switches to edit mode
  - **Add** (only on the last row). Clicking this icon adds a new row at the end of the list and switches it to edit mode. You can select the type of the newly added filter.

Type
: The filter type defines what will happen when a directory or file matches the regex filter and can be one of:
  - **Exclude a table**. Completely skips backing up tables whose names match the regular expression.
  - **Do not backup a table's contents**. Only backs up the structure of tables whose names match the regular expression, but not their contents.

Filter Item
: This is the actual regular expression you have to write.

RegEx Database Tables Exclusion - Edit Mode



When you click on the pencil or add icons, the respective row enters the edit mode. In this mode, the filter type becomes a drop-down list where you can select the type of this filter row. The filter item column also turns into an edit box so that you can enter your filter definition. The icon column now contains two different icons:

- **Disk**. When you click it, the changes will be saved.
- **Cancel**. When you click it, any changes will be cancelled and the row will resume its previous state.

You can make sure that your filters match the tables you meant to. Just go back to the Control Panel and click on the Database Tables Exclusion button. The items filtered out by the regular expressions filters will be automatically highlighted in red. You can browse through the database structure to make sure that only the items you really meant are being excluded.

5.4.1. Regular Expressions recipes for database tables

No matter how good you are on writing regular expressions, it's always a good idea to have some recipes which serve as a starting point for cooking your own.

1. Exclude non-Joomla! database tables

   ```
   !/^#__/
   ```

2. Exclude Akeeba Backup tables. We know that these tables have `ak_` in their name after the table prefix, e.g. `abc1_ak_foobar` becomes `#__ak_foobar`, so you only need to filter `#__ak`.

   ```
   /^#__ak_/
   ```

6. Automating your backup

6.1. Taking backups automatically

Even though Akeeba Backup makes it very easy to take a backup of your Joomla!™ site, doing so manually is boring. Moreover, experience tells us that if you don't take backups automatically the chances are that when you need one you'll have not taken a backup for a long period of time. To this end, Akeeba Backup offers multiple methods to let you run backups automatically, without further human interaction.

Tip

Go to Akeeba Backup's Schedule Automatic Backups page. You will get a condensed version of these instructions, personalized -to the degree possible- for your own site and *the currently active backup profile*. Trust us, this page will save you a LOT of headache.

6.1.1. Front-end backup, for use with CRON

Tip

This option is available in both the Akeeba Backup Core and Akeeba Backup Professional releases. You don't need to subscribe to the Professional edition to use it.

The front-end backup feature is intended to provide the capability to perform an unattended, scheduled backup of your site.

The front-end backup URL performs a single backup step and sends a redirection (HTTP 302) header to force the client to advance to the next page, which performs the next step and so forth. You will only see a message upon completion, should it be successful or not. There are a few limitations, though:

- **It is not designed to be run from a normal web browser**, but from an unattended cron script, utilizing **wget** or **cron** as a means of accessing the function.
- The script is not capable of showing progress messages.
- Normal web browsers tend to be "impatient". If a web page returns a bunch of redirection headers, the web browser thinks that the web server has had some sort of malfunction and stop loading the page. It will also show some kind of "destination unreachable" message. Remember, these browsers are meant to be used on web pages which are supposed to show some content to a human. This behaviour is normal. Most browsers will quit after they encounter the twentieth page redirect response, which is bound to happen. Do not report a "bug" stating that Firefox, Internet Explorer, Chrome, Safari, Opera or another browser doesn't work with the front-end backup feature. It was NOT meant to work by design and you've been sufficiently warned.
- Command line utilities, by default, will also give up loading a page after it has been redirected a number of times. For example, **wget** gives up after 20 redirects, **curl** does so after 50 redirects. Since Akeeba Backup redirects once for every step, it is advisable to configure your command line utility with a large number of redirects; about 10000 should be more than enough for virtually all sites.

Tip

If you want to automate your backups despite your host not supporting proper CRON jobs you can use a third party service, such as Webcron.org [http://webcron.org/]. Just make sure you set up the time limit to be at least 10% more than the time it takes for Akeeba Backup to backup your site. Don't know how much is that? Just take a regular backup from your site's back-end, then go to the Manage Backups page and take a look at the Duration column.

We **VERY STRONGLY** recommend using the Front-End Backup feature only with sites configured to use HTTPS with a properly signed SSL certificate *for security reasons*: plain HTTP sites and self-signed HTTPS certificates can, under certain circumstances, lead to your Secret Word leaking. If a malicious user obtains the Secret Word they can launch a Denial of Service attack on your site and / or abuse Akeeba Backup's feature to obtain a copy of your site, including all privileged information. Getting a properly signed SSL certificate no longer costs any money. The Let's Encrypt certificate authority [https://letsencrypt.org/] offers free of charge SSL certificates. Most likely your hosting control panel already supports automatically acquiring and installing SSL certificates from Let's Encrypt. For example two of our favorite hosts, SiteGround and Rochen, have supported this since late 2015. If you are not sure, ask your host. Using HTTPS not only makes your site safer, it will even make it more popular with search engines. It's a win-win proposition!

Before beginning to use this feature, you must set up Akeeba Backup to support the front-end backup option. First, go to Akeeba Backup's main page and click on the Options button in the toolbar. Find the option titled Enable front-end and remote backup and set it to `Yes`. Below it, you will find the option named Secret key. In that box you have to enter a password which will allow your CRON job to convince Akeeba Backup that it has the right to request a backup to be taken. Think of it as the password required to enter the VIP area of a night club. After you're done, click the Save & Close button on top to save the settings and close the dialog.

Tip

Use only lower- and upper-case alphanumeric characters (0-9, a-z, A-Z) in your secret key. Do not use symbols, accented characters, non-Latin character sets (like Green or Cyrillic letters) etc. Such characters may need to be manually URL-encoded in the CRON job's command line. This is error prone and can cause the backup to never start even though you'll be quite sure that you have done everything correctly.

Most hosts offer a CPanel of some kind. There has to be a section for something like "CRON Jobs", "scheduled tasks" and the like. The help screen in there describes how to set up a scheduled job. One missing part for you would be the command to issue. Simply putting the URL in there is not going to work.

Warning

If your host only supports entering a URL in their "CRON" feature, this will most likely not work with Akeeba Backup. There is no workaround. It is a hard limitation imposed by your host: they do NOT follow redirections. In these cases you can schedule a CRON job on your own computer. The downside is that your computer will need to be powered on (not turned off or even in sleep / hibernate) at the time you've set up the backup to run and for the entire length of time it take to run the backup.

If you are on a UNIX-style OS host (usually, a Linux host) you most probably have access to a command line utility called **wget**. It's almost trivial to use:

wget --max-redirect=10000 "http://*www.yoursite.com*/index.php?option=com_akeeba&

view=backup&key=*YourSecretKey*"

Of course, the line breaks are included for formatting clarity only. You should not have a line break in your command line!

Do not miss the **--max-redirect=10000** part of the **wget** command! If you fail to include it, the backup will not work with **wget** complaining that the maximum number of redirections has been reached. This is normal behavior, it is not a bug.

YourSecretKey must be URL-encoded. You can use an online tool like http://www.url-encode-decode.com or simply consult the Schedule Automatic Backups page.

Do not forget to surround the URL in double quotes. If you don't the backup will fail. The reason is the way operating systems parse command lines. Special characters such as question marks and ampersands have special meanings.

If you're unsure whether your command line makes sense please check with your host. Sometimes you have to get from them the full path to wget in order for CRON to work. For example:

/usr/bin/wget --max-redirect=10000 "http://*www.yoursite.com*/index.php?option=com_akeeba&

view=backup&key=*YourSecretKey*"

Again, please do contact your host; they usually have a help page for all this stuff. Read also the section on CRON jobs below.

Optionally, you can also include an extra parameter to the above URL, `&profile=`*`profile_id`*, where *`profile_id`* is the numeric ID of the profile you want to use for the backup. If you don't specify this parameter, the default backup profile (ID=1) will be used. In this sense, the aforementioned URL becomes:

/usr/bin/wget --max-redirect=10000 "http://*www.yoursite.com*/index.php?option=com_akeeba&

view=backup&key=*YourSecretKey*&profile=*profile_id*"

`wget` is multi-platform command line utility program which is not included with all operating systems. If your system does not include the `wget` command, it can be downloaded at this address: http://wget.addictivecode.org/FrequentlyAskedQuestions#download. The `wget` homepage is here: http://www.gnu.org/software/wget/wget.html. Please note that the option `--max-redirect` is available on `wget` version 1.11 and above. If you are on an incredibly outdated server with `wget` version 1.10 and earlier the backup will most probably result into an error message concerning the maximum redirections limit being exceeded. This is *not* a bug in our software, it's a limitation of an ancient version of the third party wget software. Kindly note that version 1.11 which lifts that limitation was released ages ago, *in 2008 (two thousand eight!)* to be more precise.

Warning

The ampersands above should be written as a single ampersand, not as an HTML entity (&). Failure to do so will result in a 403: Forbidden error message and no backup will occur. This is not a bug in our software, it's how the Internet works.

We would like to note that some SEO or SEF URL extensions for Joomla! may get in the way of front-end backups, *especially on multi-language sites*. If you get inexplicable 403 or 404 errors towards the beginning of a front-end backup right after a redirection (HTTP code 301, 302 or 307) please consult with the developers of the SEO / SEF URL extensions you are using. Usually you can add an exception for Akeeba Backup's front-end backup URLs.

Furthermore, some hosts with very finicky web server firewalls may automatically block the front-end backup URL. Typically you get a 403 error at the very beginning of the backup process or after 2-3 redirections (at which point the front-end backup will no longer work). This typically happens when they misunderstand the front-end backup Secret Word as a security threat. Try changing your Secret Word to something else. If the problem persists please contact your host and ask them to take a look and add an exception for the front-end backup Secret Word you are using.

Using webcron.org to automate your backups

Assuming that you have already bought some credits on webcron.org, here's how to automate your backup using their service.

First, go to Akeeba Backup's main page (Control Panel) and click on the Options button in the toolbar. Find the option titled Enable front-end and remote backup and set it to `Yes`. Below it, you will find the option named Secret key. Type in a secret key. We strongly recommend using only alphanumeric characters, i.e. 0-9, a-z and A-Z. For the sake of this example, we will assume that you have entered `ak33b4s3cRet` in that field. We will also assume that your site is accessible through the URL `http://www.example.com`.

Log in to webcron.org. In the CRON area, click on the New Cron button. Here's what you have to enter at webcron.org's interface:

- **Name of cronjob**: anything you like, e.g. "Backup www.example.com"
- **Timeout**: 180sec; if the backup doesn't complete, increase it. Most sites will work with a setting of 180 or 600 here. If you have a very big site which takes more than 5 minutes to back itself up, you might consider using Akeeba Backup Professional and the native CRON script (akeeba-backup.php) instead, as it's much more cost-effective.
- **Url you want to execute**: `http://www.example.com/index.php?option=com_akeeba&view=backup&key=ak33b4s3cRet`
- **Login** and **Password**: Leave them blank
- **Execution time** (the grid below the other settings): Select when you want your CRON job to run
- **Alerts**: If you have already set up alert methods in webcron.org's interface, we recommend choosing an alert method here and not checking the "Only on error" so that you always get a notification when the backup CRON job runs.

Now click on Submit and you're all set up!

A PHP alternative to wget

As user DrChalta pointed out in a forum post, there is an alternative to **wget**, as long as your PHP installation has the cURL extension installed and enabled. For starters, you need to save the following PHP script as backup.php somewhere your host's **cron** feature can find it. Please note that this is a command-line script and needn't be located in your site's root; it should be preferably located above your site's root, in a non web-accessible directory.

The script below is a modification over DrChalta's original script, taking into account changes made in later versions of our software. In order to configure it for your server, you only have to change the first three lines.

```
<?php
define('SITEURL', 'http://www.example.com'); // Base URL of your site
define('SECRETKEY', 'MySecretKey'); // Your secret key
define('PROFILE',1); // The profile's ID
```

```
// ====================== DO NOT MODIFY BELOW THIS LINE ======================
$curl_handle=curl_init();
curl_setopt($curl_handle,CURLOPT_URL,
SITEURL.'/index.php?option=com_akeeba&view=backup&key='.
SECRETKEY.'&profile='.PROFILE);
curl_setopt($curl_handle,CURLOPT_FOLLOWLOCATION,TRUE);
curl_setopt($curl_handle,CURLOPT_MAXREDIRS,10000); # Fix by Nicholas
curl_setopt($curl_handle,CURLOPT_RETURNTRANSFER,1);
$buffer = curl_exec($curl_handle);
curl_close($curl_handle);
if (empty($buffer))
    echo "Sorry, the backup didn't work.";
else
    echo $buffer;
?>
```

Where *www.yoursite.com* and *YourSecretKey* should be set up as discussed in the previous section.

The ampersands above should be written as a single ampersand, not as an HTML entity (&).

In order to call this script with a schedule, you need to put something like this to your crontab (or use your host's CRON feature to set it up):

```
0 3 * * 6 /usr/local/bin/php /home/USER/backups/backup.php
```

Where */usr/local/bin/php* is the absolute path to your PHP command-line executable and */home/USER/backups/backup.php* is the absolute path to the script above.

If you set up your **cron** schedule with a visual tool (for example, a web interface), the command to execute part is "*/usr/local/bin/php /home/USER/backups/backup.php*".

Thank you DrChalta for this wonderful tip!

Using the front-end backup in SiteGround and other hosts using cURL instead of wget

As one of our users pointed out in the support forum, finding the correct command to issue for the CRON job is tricky. What he writes applies not only to his host, SiteGround, but many other commercial hosts as well. We'll simply quote our user, bzcoder.

In the CPanel for SiteGround there is a cronjob option, you create a cronjob using that and use:

curl -b /tmp/cookies.txt -c /tmp/cookies.txt -L --max-redirs 1000 -v "*<url>*"

as your command.

Replace *<url>* with your backup URL. Make sure to use the initial url displayed on the backend NOT the final URL when you run the backup manually (been there, done that) - when you do that you end up with a url that doesn't work because of the extra parameter used in continuing the backup process.

6.1.2. Native CRON script

Tip

This option is only available in Akeeba Backup Professional.

If you have access to the command-line version of PHP, Akeeba Backup Professional includes an even better - and faster - way of scheduling your backups. The file `cli/akeeba-backup.php` can be executed from the

command-line PHP interface (PHP CLI). It doesn't require the front-end backup in order to work; it is a self-contained, native backup for your Joomla!™ site, even if your web server is down.

In order to schedule a backup, you will have to use the following command line to your host's CRON interface:

```
/usr/local/bin/php /home/USER/webroot/cli/akeeba-backup.php
```

where */usr/local/bin/php* is the path to your PHP CLI executable and */home/USER/webroot* is the absolute path to your web site's root. You can get this information from your host.

The backup script accepts three optional parameters:

- **--profile *profile_id*** allows you to select which backup profile you would like to use for the scheduled backup. The *profile_id* is the numeric profile ID you can see in your Control Panel page.
- **--description "*Your description*"** allows you set a backup description different than the default. Do not forget to enclose your description in double quotes, or this parameter will not work! Since Akeeba Backup 3.1 the description supports Akeeba Backup's file naming "variables", e.g. `[SITE]`, `[DATE]` and `[TIME]`. These variables are documented in the Output Directory configuration option's description. This allows you to use them in conjunction with this parameter to provide flexible backup descriptions.
- **--override "keyname=value"** allows you to override profile configuration variables. This parameter can appear an unlimited number of times in the command line. It can be used, for example, to provide the username and password to your cloud storage service in the command line, without having to store it in the backup profile's configuration, therefore never storing it in database and hiding it from other administrators. Please take a look at the "Overriding configuration variables" subsection for more information.
- **--quiet** will suppress all output except warnings and error messages. If the backup runs successfully you get no output at all. Note: this option was added in Akeeba Backup Professional 3.3.4.

The `akeeba-backup.php` script will return a different exit code, depending on the backup status. When the backup is successful and without warnings, the exit code will be 0. When the backup completed but with warnings, the exit code will be 1. Finally, if the backup fails, the exit code will be 2. This allows you to check the backup status, for example inside a shell script, for automation purposes.

In order to give some examples, I will assume that your PHP CLI binary is located in `/usr/local/bin/php` - a common setting among hosts - and that your web site's root is located at `/home/johndoe/httpdocs`.

1. Backup with the default profile (ID = 1) and default description:

   ```
   usr/local/bin/php /home/johndoe/httpdocs/cli/akeeba-backup.php
   ```

2. Backup with profile number 2 and default description:

   ```
   usr/local/bin/php /home/johndoe/httpdocs/cli/akeeba-backup.php --profile=2
   ```

3. Backup with the default profile (ID = 1) and a description reading "My automated backup":

   ```
   usr/local/bin/php /home/johndoe/httpdocs/cli/akeeba-backup.php --description="My auto
   ```

4. Backup with profile number 2 and a description reading "My automated backup":

   ```
   usr/local/bin/php /home/johndoe/httpdocs/cli/akeeba-backup.php --profile=2
     --description="My automated backup"
   ```

It goes without saying that the line breaks are for readability only. You should not include line breaks in your command line.

Special considerations:

- All parameters must start with a double dash. If you use a single dash, they will be ignored. This is a limitation of Joomla's CLI application interface –used by our script– which follows the UNIX conventions of command line parameters.

- Most hosts do not impose a time limit on scripts running from the command-line. If your host does and the limit is less than the required time to backup your site, the backup will fail.
- This script is not meant to run from a web interface. If your host only provides access to the CGI or FastCGI PHP binaries or pseudo-CRON (accessing a URL on a schedule), `akeeba-backup.php` will not work with them. The reason is actually the same as the time constraint above.
- Some servers do not fully support this backup method. The usual symptoms will be a backup which starts but is intermittently or consistently aborted in mid-process without any further error messages and no indication of something going wrong. In such a case, trying running the backup from the back-end of your site will work properly. If you witness similar symptoms please use the Alternative CRON Script, outlined in the next section.

Setting up a CRON job on cPanel

Note

This section depends on your host's control panel software. It is included it for informational purposes only and we cannot guarantee its accuracy. If you have questions about this please ask your host.

Go to your cPanel main page and choose the CRON Jobs icon from the Advanced pane. In the Add New CRON Job box on the page which loads, enter the following information:

Common Settings — Choose the frequency of your backup, for example once per day.

Command — Enter your backup command. Usually, you have to use something like:

```
/usr/bin/php5-cli /home/myusername/public_html/cli/akeeba-backup.php --pro
```

where *myusername* is your account's user name (most probably the same you use to login to cPanel) and *YourProfileID* is the numeric profile number you want to use for your backup job. Do note the path for the PHP command line executable: `/usr/bin/php5-cli`. This is the default location of the correct executable file for cPanel 11 and later. Your host may use a different path to the executable. If the command never runs, ask them. We can't help you with that; only those who have set up the server know the changes they have made to the default setup.

Finally, click the Add New Cron Job button to activate the CRON job.

Special considerations for HostGator

Note

This section depends on your host's control panel software. It is included it for informational purposes only and we cannot guarantee its accuracy. If you have questions about this please ask your host.

The location of the PHP CLI binary is `/usr/bin/php-cli`. This means that your CRON command line should look like:

```
/opt/php53/bin/php /home/myusername/public_html/cli/akeeba-backup.php --profile=YourPro
```

Overriding configuration variables

You can override or supply missing configuration variables in the command line. This is especially useful for security reasons. One security issue with the cloud storage service integration is that other Super Users can peek at Akeeba Backup's configuration and read the username, password or API keys used to access the cloud storage service. You can, however, leave these fields blank in the configuration and supply their values in the command line.

Overriding a configuration variable requires knowing its key name. The key names are represented in dot-format, i.e. `engine.postproc.s3.accesskey` for Amazon S3's access key. Determining the key name is quite

easy, as they are stored in INI files throughout the component's back-end. The first location you should look at is `administrator/components/com_akeeba/engine/core`, where you will find four INI files with general settings. Inside the `administrator/components/com_akeeba/engine` subdirectories you will find one INI file per engine.

In order to save you from trouble, here are the most useful key names. The names are designed to be self-explanatory.

JPS archive password	• engine.archiver.jps.key
ANGIE password	• engine.installer.angie.key
Amazon S3	• engine.postproc.s3.accesskey • engine.postproc.s3.secretkey
Microsoft Windows Azure BLOB Storage	• engine.postproc.azure.account • engine.postproc.azure.key
RackSpace CloudFiles	• engine.postproc.cloudfiles.username • engine.postproc.cloudfiles.apikey
CloudMe	• engine.postproc.cloudme.username • engine.postproc.cloudme.password
DreamObjects	• engine.postproc.dreamobjects.accesskey • engine.postproc.dreamobjects.secretkey
Dropbox (v1 API, old)	• engine.postproc.dropbox.token • engine.postproc.dropbox.token_secret
Dropbox (v2 API, new)	• engine.postproc.dropbox2.access_token
Remote FTP server	• engine.postproc.ftp.user • engine.postproc.ftp.pass
Google Drive	• engine.postproc.googledrive.refresh_token
Google Storage	• engine.postproc.googlestorage.accesskey • engine.postproc.googlestorage.secretkey
iDriveSynce	• engine.postproc.idrivesync.username • engine.postproc.idrivesync.password • engine.postproc.idrivesync.pvtkey
OneDrive	• engine.postproc.onedrive.access_token • engine.postproc.onedrive.refresh_token
Remote SFTP server	• engine.postproc.sftp.user

- engine.postproc.sftp.pass — Either the password for the username specified above, or the password to the private key file
- engine.postproc.sftp.privkey — Absolute path to the private key file (optional, for certificate authentication)
- engine.postproc.sftp.pubkey — Absolute path to the public key file (optional, for certificate authentication)

SugarSync
- engine.postproc.sugarsync.email
- engine.postproc.sugarsync.password

WebDAV
- engine.postproc.webdav.username
- engine.postproc.webdav.password

For your information, the configuration keys for cloud storage services can be found in the .ini files under `administrator/components/com_akeeba/BackupEngine/Postproc`.

Applying them on the command line is easy. Take this command line as an example:

```
usr/local/bin/php /home/johndoe/httpdocs/cli/akeeba-backup.php
--profile=2 --description="My automated backup"
--override="engine.postproc.s3.accesskey=ABCDEF"
--override="engine.postproc.s3.secretkey=1234567890abcdefgh"
```

In this case, we are telling the backup script to use the backup Profile with ID=2, give the backup description of "My automated backup" and then supply the S3 access and secret keys. The values of the override parameters must be enclosed in double or single quotes (depends on your Operating System), otherwise the operating system will not pass them back to the backup.php script. Do note that your command line MUST NOT include the line breaks in the previous example. The line breaks are there only for typesetting purposes.

Finally, it should be noted that you can use the command-line override feature to do more tricky configuration overrides, for example turning off the archive splitting or using a different backup output directory to enhance your security. If it's something you can do in the Configuration page of the component, you can also do it using command line overrides.

6.1.3. Alternative CRON script

Tip

This option is only available in Akeeba Backup Professional.

This script uses the front-end backup feature of Akeeba Backup to run a backup. This may work on some hosts where the regular CRON script doesn't. The alternative CRON script is located in `cli/akeeba-altbackup.php`, and must be run from the command-line PHP interface (PHP CLI).

As already stated in the Front-End Backup feature, we **VERY STRONGLY** recommend using the Front-End Backup feature -including the case where it's used by the alternative CRON script- only with sites configured to use HTTPS with a properly signed SSL certificate *for security reasons*: plain HTTP sites and self-signed HTTPS certificates can, under certain circumstances, lead to your Secret Word leaking *even if you are only using it with the alternative CRON script*. If a malicious user obtains the Secret Word they can launch a Denial of Service attack on your site and / or abuse Akeeba Backup's feature to obtain a copy of your site, including all privileged information. Getting a properly signed SSL certificate no longer costs any money. The Let's Encrypt certificate authority [https://letsencrypt.org/] offers free of charge SSL certificates. Most likely your hosting control panel already supports automatically acquiring and installing SSL certificates from Let's Encrypt. For example two of our favorite hosts, SiteGround and Rochen, have supported this since late 2015. If you are not sure, ask your host. Using HTTPS not only makes your site safer, it will even make it more popular with search engines. It's a win-win proposition!

You will have to a command line similar to this with your host's CRON interface:

```
/usr/local/bin/php /home/USER/webroot/cli/akeeba-altbackup.php
```

where */usr/local/bin/php* is the path to your PHP CLI executable and */home/USER/webroot* is the absolute path to your web site's root. You can get this information from your host.

The backup script accepts the following optional parameters:

- **--profile *profile_id*** allows you to select which backup profile you would like to use for the scheduled backup. The *profile_id* is the numeric profile ID you can see in your Control Panel page.
- **--no-verify** Only applies to HTTPS connections. If you have a plain HTTP site you can ignore this setting. Since Akeeba Backup 5.5.2 this script will check that the SSL certificate presented by the server is issued by a known, trusted Certification Authority and that the domain name included in the certificate matches the domain name of your site. If you want to disable this verification, e.g. because you're using a self-signed certificate or a certification authority internal to your organization you need to pass the --no-verify option. This will disable the verification, emulating the way this script worked in Akeeba Backup 5.5.1 and earlier.

In order to give some examples, we will assume that your PHP CLI binary is located in `/usr/local/bin/php` - a common setting among hosts - and that your web site's root is located at `/home/johndoe/httpdocs`.

1. Backup with the default profile (ID = 1)

   ```
   usr/local/bin/php /home/johndoe/httpdocs/cli/akeeba-altbackup.php
   ```

2. Backup with profile number 2

   ```
   usr/local/bin/php /home/johndoe/httpdocs/cli/akeeba-altbackup.php --profile=2
   ```

It goes without saying that the line breaks are for readability only. You should not include line breaks in your command line.

Special considerations:

- Most hosts do not impose a time limit on scripts running from the command-line. If your host does and the limit is less than the required time to backup your site, the backup will fail.
- This script is not meant to run from a web interface. If your host only provides access to the CGI or FastCGI PHP binaries, `backup.php` will not work with them. The solution to this issue is tied to the time constraint above. The workaround we're planning will solve both issues.
- You must enable the front-end backup feature of your Akeeba Backup Professional instalaltion and assign a Secret Key for it. This is possible by going to the Akeeba Backup Professional's Control Panel page and clicking on the Options button on the top right corner of the toolbar.
- Before using the alternative CRON script for the first time, or after moving your site to a new domain name and / or enabling HTTPS, you must visit the Akeeba Backup's Control Panel page at least once. This will cache the URL to your site for use by the alternative CRON script.
- Your host must support one of the three methods used by the helper script to access your front-end backup URL:

  1. The PHP cURL module.
  2. The fsockopen() method
  3. The fopen() URL wrappers

  If none of these methods is available, the backup will fail.
- Taking a backup with the front-end backup feature must be possible on your site. See the Discussion in the chapter about the front-end backup, especially the issues with SEO / SEF URL extensions and host firewalls.

- Your host may have a firewall setup which doesn't allow the CRON script to access the front-end backup URL if it's launched from the same server. In this case the backup will consistently fail without a new log file being produced and without a backup entry being written to the database. You will have to contact your host so that they can allow the script to access the front-end backup URL. Do note that despite the alternative CRON script and your site running on the same server, the firewall restriction might still be in place. This is counter-intuitive, but we've seen this happening on a few hosts.

If you are seeking assistance regarding a failed CRON job please indicate if and which of these steps you have already tried. We don't want to ask you to do something you've already tried.

Setting up a CRON job on cPanel

Note

This section depends on your host's control panel software. It is included it for informational purposes only and we cannot guarantee its accuracy. If you have questions about this please ask your host.

Go to your cPanel main page and choose the CRON Jobs icon from the Advanced pane. In the Add New CRON Job box on the page which loads, enter the following information:

Common Settings — Choose the frequency of your backup, for example once per day.

Command — Enter your backup command. Usually, you have to use something like:

```
/usr/bin/php5-cli /home/myusername/public_html/cli/akeeba-altbackup.php --
```

where *myusername* is your account's user name (most probably the same you use to login to cPanel) and *YourProfileID* is the numeric profile number you want to use for your backup job. Do note the path for the PHP command line executable: `/usr/bin/php5-cli`. This is the default location of the correct executable file for cPanel 11 and later. Your host may use a different path to the executable. If the command never runs, ask them. We can't help you with that; only those who have set up the server know the changes they have made to the default setup.

Finally, click the Add New Cron Job button to activate the CRON job.

Special notes for GoDaddy

Note

This section depends on your host's control panel software. It is included it for informational purposes only and we cannot guarantee its accuracy. If you have questions about this please ask your host.

According to our users who have tried this, this laternative script does work with GoDaddy. The command line you have to use is:

```
/usr/local/php5/bin/php "$HOME/html/cli/akeeba-altbackup.php" --profile=YourProfileID
```

where *YourProfileID* is the numeric profile number you want to use for your backup job.

The PHP executable we are using is the CLI rather than the default CGI. This is important; if you use the CGI executable then the script will not run. Don't forget to enable frontend backup and insert your secret word. To enable frontend backup go to Akeeba Backup under components, select configuration, select options from the navigation, then select the front-end backup tab to enable the settings.

If the backup completes successfully but the backup appears as "Failed" in the Manage Backups (formerly "Administer Backup Files") page, you'll have to apply a workaround. Go to Akeeba Backup and select your backup profile from the drop-down list. Then click on the Configuration button. In the configuration page check the Use database storage for temporary data option.

6.2. Checking for failed backups automatically

Tip

This option is only available in Akeeba Backup Professional.

While you can automate backups with any of the methods explained above, there is a small drawback. It is impossible to catch a failed backup if the backup failure was caused by a PHP error or the server killing the backup script for any reason (usually: time, file size and memory limits). This has the unwanted side effect of not knowing when your backup has failed unless you keep track of the backup records on your sites or the emails sent out by your CRON jobs (if any are sent at all – it depends on the server / service you are using).

You can automate the check for failed backups and have it email you when it detects that the latest backup has failed.

Warning

This is an optional, advanced and potentially dangerous feature: if you check for failed backups while a backup is still running you will cause the backup to fail. We recommend scheduling backup checks a substantial amount of time (e.g. 1 hour) after the expected end time of your backups.

6.2.1. Front-end backup failure check, for use with CRON

The front-end backup failure check feature lets you perform an unattended, scheduled failed backup check.

Before beginning to use this feature, you must set up Akeeba Backup to support the front-end backup option as explained in the relevant section of this documentation.

The URL to use to trigger the front-end backup failure check feature is

http://*www.yoursite.com*/index.php?option=com_akeeba&view=check&key=*YourSecretKey*

where *YourSecretKey* is the Secret Word for front-end backups, as configured in the component's Options page.

Please note that *YourSecretKey* must be URL-encoded. You can use an online tool like http://www.url-encode-decode.com or simply consult the Schedule Automatic Backups page.

If you want to automate the check you can schedule this URL just like you would a front-end backup URL. Furthermore, the front-end backup failure check URL does NOT use redirections. Therefore it can even be used with hosts and services which do not follow redirections.

6.2.2. CRON script for backup failure check

Tip

This option is only available in Akeeba Backup Professional.

The CRON script is located in `cli/akeeba-check-failed.php`, and must be run from the command-line PHP interface (PHP CLI).

In order to schedule a backup failure check, you will have to use the following command line to your host's CRON interface:

/usr/local/bin/php /home/USER/webroot`/cli/akeeba-check-failed.php`

where */usr/local/bin/php* is the path to your PHP CLI executable and */home/USER/webroot* is the absolute path to your web site's root. You can get this information from your host.

The same considerations and scheduling information as the backup CRON script apply.

6.2.3. Alternative CRON script for backup failure check

Tip

This option is only available in Akeeba Backup Professional.

This script uses the front-end backup failure check feature outlined above. The alternative CRON script is located in `cli/akeeba-altcheck-failed.php`, and must be run from the command-line PHP interface (PHP CLI).

In order to schedule a backup failure check, you will have to use the following command line to your host's CRON interface:

`/usr/local/bin/php /home/USER/webroot``/cli/akeeba-altcheck-failed.php`

where *`/usr/local/bin/php`* is the path to your PHP CLI executable and *`/home/USER/webroot`* is the absolute path to your web site's root. You can get this information from your host.

The same considerations and scheduling information as the alternative backup CRON script apply.

7. Site Transfer Wizard

What is the Site Transfer Wizard?

One of the most common uses of Akeeba Backup is transferring a site between different locations (folders, subdomains, domains and servers). Typically this involves taking a backup, downloading it to your computer, uploading it to the new location alongside Kickstart and launching Kickstart to extract the backup archive and proceed with the restoration. The download and upload part of this process takes a lot of time, especially when you have a slower connection. The Site Transfer Wizard will save you some precious time by eliminating the need to transfer the backup archive through your computer, instead performing a server to server transfer.

We recommend that you try using the Site Transfer Wizard *without* reading this documentation section. You only need to refer to this documentation in case a server issue or a mistake in the information you entered prevents you from using it. That's why this documentation section is brutally long; it's *troubleshooting*, not regular usage documentation. The Site Transfer Wizard is intuitive enough to use without reading its documentation.

Important

The Site Transfer Wizard IS NOT the only way to transfer your site with Akeeba Backup and IS NOT guaranteed to work on all servers. If your site is very big, your server too slow or simply doesn't support the requirements of the Site Transfer Wizard then the wizard will fail to transfer the backup archive for you. **We cannot do anything against your host's technical limitations. However, you can still transfer your site with the Manual method available in the Site Transfer Wizard**. In a nutshell: you can take a backup; download the backup archive files to your site; upload the backup archive files and Kickstart where you want to restore the site to; run Kickstart. The Wizard will display a video tutorial about this when you select the Manual method.

Prerequisites

Before you begin you must have create a new database for the destination site. This is something that Akeeba Backup and its restoration script is not allowed to do due to the configuration of most servers. This has to do with your server's database security settings and cannot be "worked around" in any way. If you are not sure how to do it please contact your host - this is a server-specific task and they are the only people who can help you with it.

You also need to know how to connect to the target location. This requires knowing the FTP, FTPS or SFTP connection information to the target location. This is required even if you are transferring to a subdirectory,

subdomain or domain on the same server your site is currently on. If you are not sure how to obtain this information please contact your host; they are the only people who can help you accurately figure out this information.

If you will be using FTP or FTPS to transfer your site your current server must either have the PHP cURL extension installed with FTP support or the PHP FTP functions enabled.It must not block outbound connection to the remote server's FTP port (typically port 21). The remote FTP server must allow connections from your site's current server and allow at least 7 connection attempts to be made within 1 second.

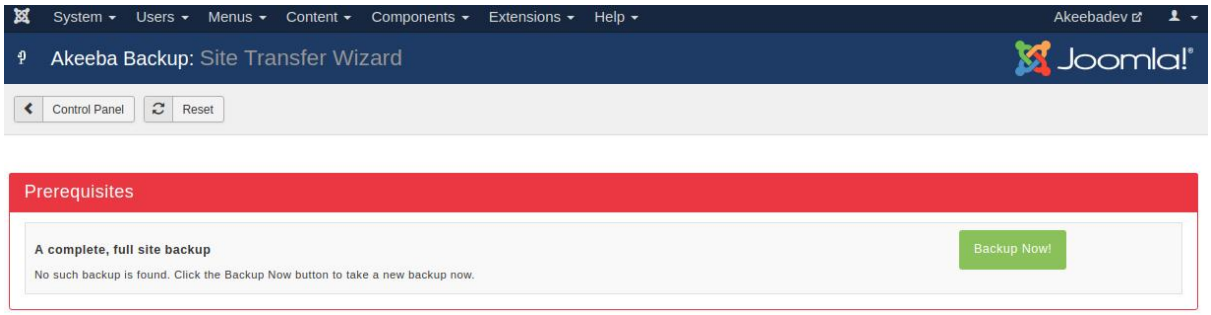
If you will be using SFTP to transfer your site your current server must either have the PHP cURL extension installed with SFTP support of the PHP SSH2 extension installed. It must not block outbound connection to the remote server's FTP port (typically port 22). The remote FTP server must allow connections from your site's current server and allow at least 7 connection attempts to be made within 1 second.

In every case the remote location MUST be accessible through HTTP/HTTPS over the Internet from your site's server and your computer. Akeeba Backup will be checking that and won't let you proceed with the transfer if it can't connect.

Backup age check

The Site Transfer Wizard requires a recent backup, taken within the last 24 hours using *the currently active backup profile*. If one is not detected you will be notified. If you want to use a backup taken with a different profile please remember to activate that profile from Akeeba Backup's main page before clicking on Site Transfer Wizard.

Backup age check

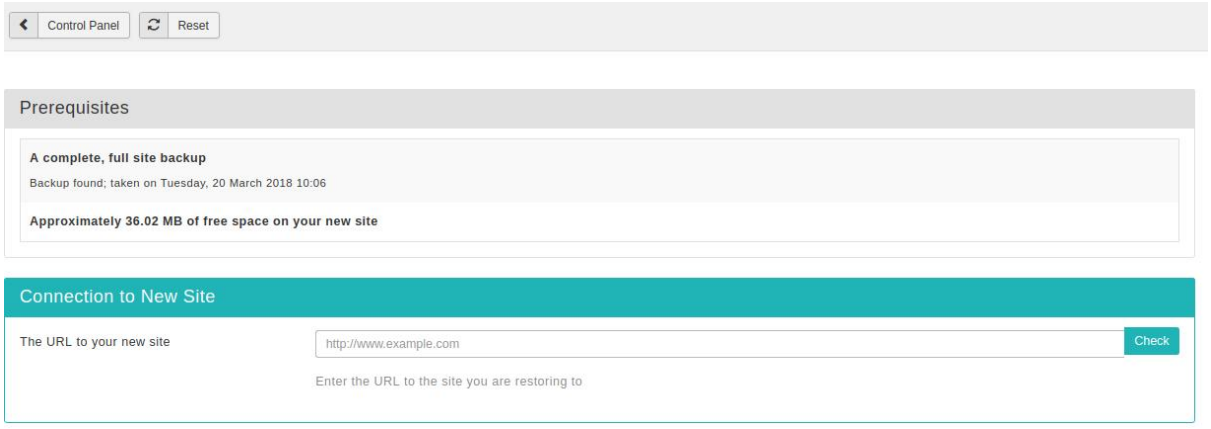


Click the Backup Now! button to take a new backup with the current backup profile.After the backup is complete you will need to go back to the main page and then click on Site Transfer Wizard again.

Setting up the transfer target URL

When a recent backup is detected the Site Transfer Wizard will let you know how much free space you will need (approximately!) on the target server. Please make sure that you have enough disk space before proceeding.

Setting up the transfer target URL



Afterwards please enter the URL of the target site and click the Check button. You must enter the full URL to the target site including the `http://` or `https://` prefix and any path to the site but without the index.php part. For example you need to enter something like `https://www.example.com`, `http://subdomain.example.net` or `http://localhost/mysite`.

The Site Transfer Wizard will check that the URL is accessible from your server.Please note that if the URL returns an error, including but not limited to 403 Forbidden and 500 Internal Server Error, you will receive a message telling you that the URL is inaccessible. In some *very rare* circumstances you may be receiving this message in error. In those cases you can click on the I want to ignore this warning and proceed at my own risk button and proceed anyway. Please note that you are doing so *at your own risk*. We will not be able to help you if something doesn't work or breaks!

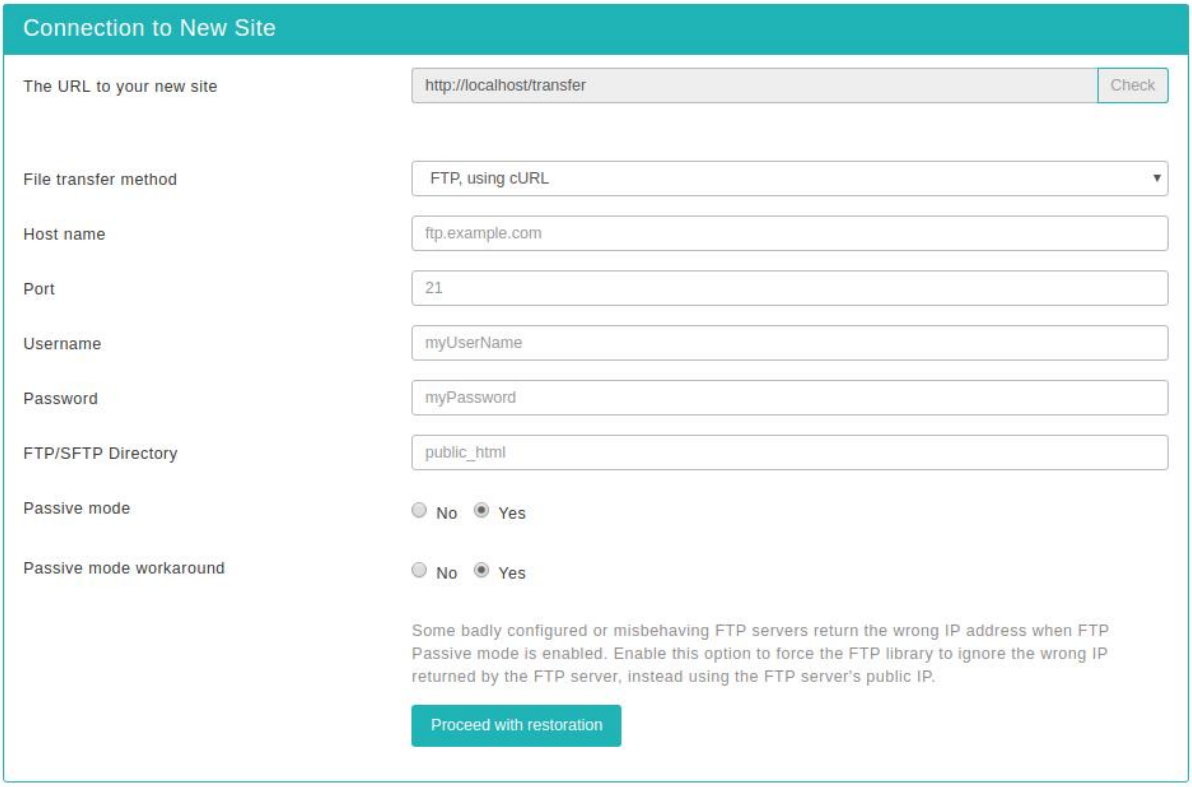
Tip

If at any point you realise you have entered the wrong URL you can click on the Reset button in the toolbar to clear all Site Transfer Wizard settings and start over.

Setting up the connection

The next step lets you tell the Site Transfer Wizard how to connect to your target site to transfer files.

Setting up the connection



Select one of the available transfer methods (not all of them may be available on your server):

FTP, using cURL	You will connect to your site using plain (insecure) FTP. This is the simplest file transfer protocol, supported by most hosts - however it's also the least secure. This method uses the PHP cURL extension which is compatible with most hosts.
FTPS, using cURL	You will connect to your site using plain FTP over a secure SSL/TLS connection. This is a simple file transfer protocol, supported by many hosts and it is quite secure. This method uses the PHP cURL extension which is compatible with most hosts.

SFTP, using cURL	You will connect to your site using file transfer over SSH (a.k.a. Secure File Transfer Protocol, or SFTP for short). This is an advanced file transfer protocol, supported by some hosts and it is the most secure. This method uses the PHP cURL extension which is compatible with most hosts.
FTP, native PHP functions	You will connect to your site using plain (insecure) FTP. This is the simplest file transfer protocol, supported by most hosts - however it's also the least secure. This method uses the PHP native FTP functions. You may experience some compatibility issues with badly configured FTP servers.
FTPS, native PHP functions	You will connect to your site using plain FTP over a secure SSL/TLS connection. This is a simple file transfer protocol, supported by many hosts and it is quite secure. This method uses the PHP native FTP functions. You may experience some compatibility issues with badly configured FTP servers.
SFTP, native PHP SSH2 extension	You will connect to your site using file transfer over SSH (a.k.a. Secure File Transfer Protocol, or SFTP for short). This is an advanced file transfer protocol, supported by some hosts and it is the most secure. This method uses the PHP SSH2 extension. Since this extension is currently marked as experimental it may not be available on your server or not work properly.
Manually	**If all else fails (your servers just can't talk to each other) choose this option, do not file a "bug" report** (as noted above, we can't override your hosts' technical limitations since *we are not your host, therefore cannot reconfigure your servers with more sane limits*). The manual method will give you instructions for performing a manual backup archive transfer, including a tutorial for restoring it after it's transferred. This is your failsafe method, one which has been used by hundreds of thousands of site integrators and site owners since 2006 to transfer their sites between different locations.

If your target site supports more than one transfer methods please try using the most secure ones first. The order of preference, from MOST to least secure is: SFTP, FTPS, FTP. Moreover, if you are given the choice between a method that uses cURL and one which doesn't please try using the cURL one first. If none of them works for you please check your connection information and retry. If nothing works despite the connection information being correct you have a case where the two servers cannot talk to each other due to networking, firewall or setup issues. The easiest thing you can do is use the Manually option to transfer your site by manually uploading your backup archive.

Enter the connection information below and click on Proceed with restoration to get to the next step. Please note that if you chose Manually above the next step simply gives you instructions for performing the transfer and the rest of this documentation section does not apply.

Files transfer and restoration

At this point the Site Transfer Wizard is going to make some sanity checks and upload some files on your server.

If the connection fails for any reason you will be told so. Please double check the connection information and the FTP/SFTP directory. The latter must exist and be both readable and writeable. If you still get an error despite all the connection information being correct please try a different connection method. If all available methods fail please do contact both hosts (the one your site is currently on and the one you're trying to transfer to). One or both of the servers have a server protection which prevents the two servers from talking to each other. If you cannot get your hosts to resolve that issue your only choice will be using the Manually option above. Unfortunately there is nothing we can do about it since it's the server which doesn't allow the server-to-server transfer in any way.

If the target server and location is the same as the one where your current site exists the process will be aborted. You MUST NOT use the Site Transfer Wizard to restore a backup archive on your own site. Either use the Restore feature in the Manage Backups Page (Professional version only) or use Akeeba Kickstart to extract the backup archive and start the restoration.

If a .htaccess file is detected on the target location the process will be aborted. The .htaccess files can interfere in the way PHP script execute, corrupting the upload of the backup archive or simply blocking the upload, extraction

or restoration process. As a precaution the Site Transfer Wizard will not proceed in this case unless you delete the .htaccess file.

After these basic checks the Site Transfer Wizard will try to upload the two Kickstart files (`kickstart.php` and `kickstart.transfer.php`) to your target location and create a new world-writable (0777 permissions) directory called `kicktemp`. Yes, we are aware that the world writable permissions are REALLY BAD for security - but only if you let them persist. We only create this directory temporarily and only use it for temporary data. After the process is done this directory is removed, therefore eliminating any possible security concern. If any of these operations fails you will receive an error message. If this happens please make sure that the target directory is writeable. If you are not sure please ask your host for assistance.

If the FTP/SFTP Directory you've entered does not correspond to the URL to the new site you have entered you will be told so. You CANNOT receive this message in error. If you get this message you MUST check that the directory corresponds to the URL you've entered. If you are not sure, or if you think that Akeeba Backup is wrong (it's not), do check with your host. **This is the most common mistake people make**. Trust us. This is exactly why we added this check.

Afterwards the Site Transfer Wizard will attempt to upload the backup archive file(s). This is done in small, 1Mb chunks. The file is NOT uploaded using FTP, FTPS or SFTP. Why? Because, as we explained previously in this documentation, transferring a big file can take too long which will cause PHP or your web server to halt with a timeout error. The Site Transfer Wizard is instead sending 1Mb of data at a time to Kickstart (which it uploaded in the previous step). Kickstart on the target location "assembles" the archive file(s) from these 1Mb chunks behind the scenes. This lets us transfer really big backup archives without timing out. The progress of the upload is displayed on the page.

However, this *may* lead to problems on some servers. Since the Site Transfer Wizard is making a lot of repeated requests to the kickstart.php URL on the target location some servers may mistakenly assume that this is an attack on the server. Other servers may not like that a lot of the CPU is being used by that site hosting account all of a sudden. If this kind of server protection is triggered you will receive an error message. Depending on the server and host they might also temporarily block the IP address of your site's current server, making it impossible to run the Site Transfer Wizard again for a period of a few minutes to a full day. If you get in this kind of situation you will have to use the Manually option and transfer the backup archives yourselves. Unfortunately there is nothing we can do about it since it's the server which doesn't allow the server-to-server transfer of large files.

When the backup archive files are fully transferred you will see a button called Run Kickstart. Click on it to launch Kickstart on the target URL. Kickstart allows you to extract the backup archive on the target server. This is required since the actual restoration script is stored inside the backup archive. If you are unsure how to proceed after this point please consult our video tutorials on transferring your site to a new server. Ignore the part where you upload Kickstart and the backup archive; this is already done for you by the Site Transfer Wizard.

Chapter 4. Miscellaneous Extensions (Modules, Plugins)

1. Akeeba Backup Notification plugin

This plugin is obsolete as of Akeeba Backup 3.11.0. Please use the Joomla! Extensions Manager to get informed for and install the updates of our components.

2. The CLI update notification and automatic update script

Note

This feature was removed in May 2017.

The automatic update script was removed in May 2017 due to massive bugs in Joomla! 3.7. These bugs broke all CLI scripts which make direct or indirect use of the JSession package, including simply checking if a user is logged in. This includes the update script. It cannot be fixed because the update script uses the core JUpdater and JInstaller APIs to fetch and install updates. Both of them use JSession which means they cannot be used from a CLI script.

The only thing we can do is remove a feature which can no longer work because Joomla! broke backwards compatibility with itself.

3. Backup on Update

Note

Displayed on the Plugin Manager as System - Backup on update

Joomla! 2.5 and later versions include the Joomla! Update component (originally developed as part of Admin Tools by our company, later donated to Joomla! and now maintained by the Joomla! team) which allows you to update Joomla! to its latest version. When you are updating between minor versions of Joomla! e.g. 3.2 to 3.3 or between major versions of Joomla! e.g. 2.5 to 3.x, some extensions on your site might experience problems or make your site inaccessible. It's always a good idea to take a backup of your site before upgrading Joomla!, but how many times did you forget to do it only to end up with an inaccessible site and a furious client? Our plugin is here to automate this process for you.

When this plugin is enabled it will "see" your attempt to update Joomla! and automatically launch Akeeba Backup to take a backup of your site. Once the backup is successfully complete it will take you back to Joomla! Update, allowing it to install the new Joomla! version. All this happens automatically. You and your clients can no longer forget to take a backup before updating Joomla!: the backup will be taken automatically.

Editing the plugin you will find the sole option, Backup Profile, which lets you define which Akeeba Backup profile to use for these automated backups. If you don't specify anything the default backup profile (the one with ID=1) will be used.

Tip

We recommend using a backup profile which stores a copy of the backup archive in external storage (e.g. Amazon S3, Dropbox or Box.com) on top of leaving a copy of the backup archive on your server. This way you have maximum protection against any kind of accidents caused by a failed or problematic Joomla! update.

When the plugin is enabled you will see an icon and a BoU label at the bottom of your site's administrator page, in the status area. You can temporarily disable (and re-enable) the backup on update feature by clicking on that icon.

Chapter 5. Restoring backups and general guidelines

We kindly request our users to read the general guidelines before filing a support ticket, a bug report or giving up. Most frustrating and "inexplicable" problems end up being a simple case of misunderstanding how things work, making the wrong configuration choice or a server issue which can typically be resolved by asking your host nicely. The guidelines below aim to prevent most of the common issues or, if prevention is unlikely, at least help you identify the root cause and tell you how to fix it.

1. General guidelines for backing up and restoring your site

Restoring sites with Akeeba Backup is easy. Sometimes it may even be *too easy* which makes you prone to making obvious mistakes due to the implied complacency of using third party software to restore your site. In the following paragraphs we explain how to avoid the most common mistakes. This is a long read but we recommend that you read and understand it. We will not accept any "bug" reports about these issues which are, ultimately, user errors. If you need clarifications about any of these issues, however, do feel free to ask us (and be specific about what you didn't understand). We will be happy to help you better understand the issue.

Make sure you have backups before you need to restore your site. Over the years we've come across many people who were furious that having installed, but not having used, our backup software didn't help them when their site got destroyed. Don't be that person. Take frequent backups of your site, at the very least before updating Joomla! and its extensions and after making any substantial change to your site. It's dull, it's boring, it's a chore, *it will save your life one day*. Always keep at least one copy (ideally: three copies) of your backups *outside of your site* and ideally outside of your computer too. What is the point of having your backups stored on your site's server when the server's disk crashes or the hosting company goes bankrupt?

Test your backups periodically. Maybe you excluded a database table you didn't mean to. Maybe a folder was unreadable but you ignored the warning. Maybe the FTP program you are using to download your backups is broken. Maybe you accidentally set the temp-directory or logs file directory of your site to your site's root or another important system folder, ending up excluding it automatically in the process. Maybe there was a bug in the version of Akeeba Backup you were using, it created a corrupt archive and you didn't update to the next version that fixed it (it happens once every 3-4 years, we usually fix the issue within 24 hours). No matter what happened, a corrupt backup can ruin your day. Test your backups periodically to make sure that they actually work.

Akeeba Backup archives are self-contained. The backup archive contains a copy of your site's files, an export of its database data *and the restoration script to put everything together*. This is very different to most other backup software in that the restoration script is inside the backup itself, not a separate thing you need to install. The only thing you need to do before restoring a backup is extract the backup archive. This can be done with Akeeba Kickstart (single file web application), Akeeba UNiTE (automatic site restoration running under the CLI) etc.

Restoring a backup overwrites the entire site (Joomla! installation). It cannot be used to transfer content between different Joomla! installations. An Akeeba Backup archive contains your entire site, files and database contents. Restoring a backup will overwrite the files and database tables that have the same name as those included in the backup. As a result it cannot be used to transfer content (e.g. articles) between two different Joomla! installations. It will transfer the entire Joomla! installation instead.

You DO NOT need to install Joomla! before restoring a backup. As explained above, an Akeeba Backup archive contains your entire site and the restoration script required to restore it. You do not need to have a functional site to restore a backup. The whole point of Akeeba Backup is restoring a backup when the site is no longer functional. Moreover, we recommend NOT installing Joomla! before restoring a backup because of the point described below about mixing different versions of Joomla!.

Restoring a backup does NOT delete files on your site which don't exist in your backup. If a file exists on the site you are restoring to but not inside the backup archive itself it will not be overwritten. This is important

in two cases. First, when you are restoring a backup after your site is hacked (*unhacking a site*). The hacker may have left behind files which can be used to re-hack your site. These files will not be deleted automatically. Use a component, such as Admin Tools Professionals with its PHP File Change Scanner, or a third party service such as myJoomla.com to detect and remove such files. Furthermore, if you are trying to *replace a site* with a new one. If the old site is based on an older version of Joomla!, a different CMS (e.g. WordPress), is a static site or had different extensions / templates installed these files will be left behind. In both of these uses cases you should take a copy of your site's files, then delete all files and folders, create a new database and *finally* restore the backup.

Make sure you are backing up only the files that belong to the site you want to back up. This is very important when you are backing up a site in the root of your domain but you have additional sites in subdirectories (or subdomains whose root directory is a subdirectory of your main site). Akeeba Backup does NOT assign meaning to your directory names! It does not know that the directory `old_site` has a copy of your site from two years ago or that the folder `gju4rl` has the files for a subdomain you use to share files with your cousin who lives in another country. Akeeba Backup will backup all files and folders under the root of your site unless you tell it otherwise with the Files and Directories Exclusion feature. In any other case restoring your main site would overwrite the files of all the sites in subdirectories under your main site's root!

Make sure you are backing up only the database tables that belong to the site you want to back up. This is very important when you are backing up a site that shares a database with another site. Akeeba Backup does NOT assign meaning to the prefix used by your database tables. All tables in the database used by your site will be backed up *regardless* of their database prefix. If you use the same database for the tables of other sites, e.g. sites installed in subdorectories or subdomains on the same hosting account, you MUST exclude them manually with the Database Tables Exclusion feature. Otherwise restoring your site would overwrite the database of all of the other sites sharing the same database.

Keep copies of your backup archives outside of your site. The reason is simple: human error. If you mess up the restoration and, at the same time, somehow delete your only backup archive you are in deep trouble. Always keep several copies of your backup archives before doing a restoration. We recommend having at least THREE copies: on your computer, on a cloud storage provider (e.g. Dropbox, OneDrive, Google Drive, ...) and on a USB stick you keep in a sealed envelope in your desk's drawer. You can never have too many copies of your backups - but you *can* have too few!

Always practice the restoration on a test server / subdomain before doing it on your live site. Do you know why the military has so many drills? By repeating the same task over and over they get to perform it near perfectly, every single time, without thinking - even when bullets are flying around them. You don't want to start learning how to restore backups when your site is down. At that point you want your site restored, pronto. That's why you should practice restoring backups before you *need* to restore them. Practice on a test server, e.g. a MAMP, WAMPServer or XAMPP installation on your computer, or a subdomain on a server under your control. Once you have done this a few times you can restore any site, anywhere, without much thinking and without mistakes.

Make sure you have a database. Even though Akeeba Backup's restoration script can try to create a database for you this WILL NOT work on most database servers for security reasons. Creating a database requires database server administrator (root) privileges which you typically do not have and, even if you do, *should not* use when restoring a site to a live server. Instead, create a database for your site in advance. You will also need to create a database user and give it the correct privileges as described in our troubleshooter [https://www.akeebabackup.com/documentation/troubleshooter/abidatabase.html].

Do not restore in a subdirectory of your main site. For example, if your site's root is in `public_html` do not restore to `public_html/dev`. The reason is that the `.htaccess` files, which tell Apache (your web server) how to server your site, *cascade*. That is, Apache will read all `.htaccess` files in all folders leading to the one hosting your site's `index.php` file. This *will* cause problems with the restored site which you will experience as 404, 403 and 500 error messages or blank pages. These have nothing to do with our software and / or the restoration. It's how your web server works. Use a subdomain instead.

If you are restoring on a subdomain, make sure that the subdomain's root directory is NOT a subdirectory of your main site. This is the same as the previous paragraph, really. Most hosting control panel software default to using a subdirectory of your site's root when creating a subdomain. For example, if your site is `www.example.com` and its root is `public_html` if you create the subdomain `dev.example.com` your hosting control panel will put its root in `public_html/dev`. Therefore you will have the problem we described

above. In this case ask your host what is the best way to create a root folder for the subdomain next to `public_html`, not inside it.

Try restoring to as close a PHP version as possible. Not all third party extensions support all PHP versions. If your site was running on PHP 5.6 and you try to restore it on a server running PHP 5.3 or PHP 7.1 your site *may* break. This has, again, nothing to do with Akeeba Backup. Upholding the minimum / maximum PHP version requirements of the software running on your site is your responsibility. We have no way of knowing that information. All we can do is print out the PHP version of the site you backed up and the site you are restoring to during restoration. Everything else is up to you.

Don't try to restore to a different database technology. If your site runs on MySQL don't try to restore it on a server that only supports PostgreSQL or Microsoft SQL Server. Even though Joomla! 3 supports all of these database technologies they are incompatible with each other and you *cannot* transfer data between them. You can only restore a site on the same database technology it was backed up on. Clarification: MySQL, Percona and MariaDB are all using the same database technology, collectively called "MySQL". While you can a site between these different MySQL-type database servers we recommend against it. Subtle differences between them may cause restoration errors in some cases. In the few cases we can prevent that, we have added the necessary workarounds. There are some cases we can do nothing about. If you get a database restoration issue please check if you're trying to restore to a different MySQL-like database server than the one you backed up from.

Do not try to overwrite one Joomla! version family with a different one. Overwriting a major version with another (e.g. restoring a backup taken on Joomla! 3.7 on top of a site running Joomla! 2.5 or vice versa) or between different minor versions (e.g. restoring a backup taken on Joomla! 3.7 on top of a site running Joomla! 3.6 or vice versa) will NOT work. Joomla! moves files around between minor and major versions. Since the backup does not delete files not present in the backup archive this will end up with Joomla! being "confused" and malfunctioning. In these cases you should delete the existing files and folders (except, perhaps, user generated content) before restoring the backup. You can safely restore a sub-minor (path-level) version on top of another. For example, you can safely restore a Joomla! 3.7.5 site on top of a Joomla! 3.7.3 site or vice versa.

Pay attention when restoring to a different domain, subdomain or folder: you will need to enter the domain name, directory and database connection information where you are restoring to. If you don't pay attention you may overwrite a site you didn't intend to touch!

The restored site is a fully functional clone of your original site. There is no functional difference between a restored site and one you built from scratch. This means that you can always backup the restored site and then restore that new backup on top of the original site. This makes Akeeba Backup ideal for live-to-development and development-to-live site transfers. If you are an advanced user such as a busy web agency do note that the process can be fully automated using Akeeba UNiTE: it can take a backup remotely, download it and restore it.

2. Guidelines for storing your backups remotely / "cloud backup"

Note

This only applies to Akeeba Backup Professional.

Uploading backups to a remote location requires setting up a Post-Processing Engine in the Configuration. By default, your backups are only stored locally, on the server where the site being backed up lives. If you want the backup to be uploaded to remote storage you have to go to the Configuration page and set up a Post-Processing Engine. As with all Configuration settings, this is set up *per backup profile*. Each profile can have a different post-processing setup - or no post-processing setup. Remember this if you're trying to figure out why your backups don't upload.

Uploading your backup archives can happen during or after taking the actual backup. All post-processing engines have an option to "Upload parts immediately". When this is enabled (checked), Akeeba Backup will upload each backup archive part file as it's finished being created. The first part of the backup is exempt from this rule: it is always uploaded after Akeeba Backup is done backing up your site. The first part contain special

information about the number of part files and / or the number and size of files in the backup, information which is only known after the backup is complete. When the "Upload parts immediately" option is disabled (unchecked, the default state) Akeeba Backup will finish taking a backup of your site and only then will it upload the backup to remote storage. Therefore, when you are perceiving an issue with Akeeba Backup "not uploading your backups" first check if you have an issue preventing the backup to be taken at all!

A failed upload to remote storage does not cause the backup record to be reported as failed. As far as Akeeba Backup is concerned, backing up and uploading are two distinct operations. If the backup completes but the upload fails the backup record will appear as "OK" (green), NOT as Failed (red). If *both* the backup and upload are successful the backup record will appear as either OK (if the backup archive is kept on your site's server) or Remote (if the backup archive is deleted from your site's server, the default option). This distinction makes sense: if the backup is complete you can still restore your site from the generated backup archive.

Most failed uploads are caused by timeouts. PHP and your web server have time limits, i.e. the maximum time a PHP script may process data before the web server aborts it. Uploading the backup archives to cloud storage takes time, the exact amount of which depends on the size of the file and the network speed. If that time is over either time limit your backup will fail. The time limit and the bandwidth are beyond our control. The only thing you can control is the size. Many post-processing engines support chunked uploading (breaking up the uploads in smaller bits and having the remote server piece together the file) and you can change their chunk size. A chunk size of 5 or 10 MB works best in most cases. For those post-processing engines which don't have an option for chunked uploads you will have to change the Part size for split archives in the Archiver Engine options. Again, a value of 5 or 10 MB works best in most cases. This setting will split our backup archive into multiple files (same base filename, the extensions are .j01, .j02, ..., .jpa; or .j01, .j02, ..., .jps; or .z01, .z02, ..., .zip;), the maximum size of each one being the value of this setting. To restore these backups just place ALL of these files in the same directory and choose the main .jpa, .jps or .zip file: the other parts are discovered and extracted automatically.

If you get no uploads / zero sized uploads but not error message, contact your host. We have seen many hosts putting a (broken) caching proxy in front of their web servers. Instead of letting Akeeba Backup communicate with the remote storage server they immediately return an HTTP 200 OK response *without contacting the remote storage server*. Unfortunately, for many remote storage services such as Dropbox, OneDrive and Google Drive *this would be the expected response when the upload succeeds*. How you can tell this happened? Check the Akeeba Backup log file. If the upload takes less than 2 seconds we GUARANTEE that your host is doing what we just described. Their first level support may deny it; ask to escalate to a server technician. They will add a proxy server exception and your remote backups will work perfectly.

You can't upload to multiple locations. You can only set up a single post-processing engine. Multiple upload locations would increase the load on your server and the likelihood that something fails during backup. Moreover, this does not offer the kind of redundancy you might hope to achieve. Instead, use Dropbox, OneDrive or Google Drive to automatically download the backup archive to your computer. Use a regular desktop backup software to back up the local copies of your site's backups to a NAS.

If some part files of your backups failed to upload use the Manage remote backups in the Manage Backups page to retry uploading them. Sometimes a temporary network issue may prevent the upload from going through. Using the Manage remote backups to retry the upload usually works just fine!

If your uploads fail with long, cryptic errors about the signatures being wrong please check the time and the timezone of your server. Most remote storage engines require your server's time to be set within a reasonable accuracy to the true time. This can be automated on your server by setting and running the ntpd service. If your host hasn't done so the time will drift until it's so far off the true time that uploads will fail. If you get these cryptic error messages about signatures first *triple check* that your credentials are set up correctly in the post-processing engine options in the Configure page for the backup profile that fails. If you have triple checked them and found them to be working, contact your host and ask them to check the time and timezone on the server. It's silly, but this is the second most common cause of upload failures (after the part size discussed above) that we keep seeing.

3. Restoring your backups

Please watch our Video Tutorials [https://www.akeebabackup.com/documentation/video-tutorials.html] for a quick (less than 10 minutes) overview of the whole process, from installing Akeeba Backup to restoring your backup archives.

If you are looking for the detailed user reference for ANGIE, the restoration script included in our backup archives, please look inside Akeeba Solo's documentation [https://www.akeebabackup.com/documentation/akeeba-solo/restoring-backups.html].

4. Troubleshooting restored sites

Please refer to our Troubleshooting Wizard's section on solving post-restoration issues [https://www.akeebabackup.com/documentation/troubleshooter/post-restoration.html]. Please note that all of them have nothing to do with Akeeba Backup, but can be attributed either to some server configuration mismatch or a pesky setting in some component, plugin, module or template.

5. Unorthodox: the emergency restoration procedure

Warning

THIS IS NOT THE REGULAR RESTORATION PROCEDURE.

The following instructions are meant to be used in absolute emergencies, when the regular restoration procedure does not work.

If you are not sure what the regular restoration process is STOP NOW and go to our site to watch the Video Tutorials.

Note

These instructions are meant to be first read before disaster strikes. They are not meant as a checklist you follow when you're stressed out by your site being down. Please keep that in mind while reading them. Thank you!

Inevitably, some people will end up with a backup file, a ruined site and a problem in the restoration procedure they can't work out. Almost always, the recipe includes a pressing deadline which requires that the site is on-line... yesterday. If you are in a situation like the one we just described, breathe. Do not panic. We've got you covered, with this concise manual site restoration guide. So, here it goes... it's manual Joomla! Site restoration In 7 steps or even less.

Step 1. Making sure it won't get worse.

Assuming such a situation, it's only human to be in panic and despair. Panic is a bad counsellor. It will give you wrong advice. Despair will only make you careless. So, people, get it together! Make a backup of the only thing separating you from complete disaster: the backup file. Burn it on a CD. Write it on your USB key. Put it on a couple of locations on your file server. Just make sure you'll have an extra copy in case you screw up.

This exercise has been proven to lower the probability of anything going wrong. Furthermore, it's good for your psychology. It gives you a sense of security you didn't have five minutes ago.

Step 2. Extracting the archive.

Now, we have to extract the archive somewhere on your local hard drive.

You'll have to use Akeeba Kickstart, available without charge from our website.

If you have a ZIP package you might be able to extract it using third party software. Typically, PKZIP for Windows, WinZIP and 7-Zip work best.

Step 3. Editing your database backup.

Take a look at the directory where you extracted your backup archive. Inside it there is a directory named `installation`. Inside this, there is a subdirectory named `sql`. Inside this there is a file, `site.sql` (older versions: `joomla.sql`), containing your database data. *COPY THIS TO ANOTHER LOCATION NOW!* We'll have to edit it, so please, don't tamper with the original, will you?

Open the copy of `site.sql` (older versions: `joomla.sql`). Use a text editor (we recommend gedit and Kate on Linux™, Notepad++ on Windows™; do not use Wordpad or Word!). If you were ever familiar with SQL, you'll recognize that each line consists of a single SQL command. But they have a problem: table names are mangled. You'll see that tables are in a form similar to #__banner instead of jos_banner. Ah, nice! We'll have to fix that.

Using your text editors Replace command, do the following changes:

- search for **CREATE TABLE `#__** replace with **CREATE TABLE `jos_**
- search for **DROP TABLE IF EXISTS `#__** replace with **DROP TABLE IF EXISTS `jos_**
- search for **INSERT INTO `#__** replace with **INSERT INTO `jos_**
- search for **CREATE VIEW `#__** replace with **CREATE VIEW `jos_**
- search for **CREATE PROCEDURE `#__** replace with **CREATE PROCEDURE `jos_**
- search for **CREATE FUNCTION `#__** replace with **CREATE FUNCTION `jos_**
- search for **CREATE TRIGGER `#__** replace with **CREATE TRIGGER `jos_**

The idea is to replace all instances of #__ (note that there are two underscores after the hash sign) with jos_ in the SQL command part (not the data part). DO NOT PERFORM A BLIND SEARCH AND REPLACE OF #__ WITH jos_ AS IT WILL CAUSE SEVERE PROBLEMS WITH SOME COMPONENTS. Easy, wasn't it? *NOW SAVE THAT FILE*!

Step 4. Restoring the database.

In order to restore the database on the server you'll have to use some appropriate tool. For small to moderately sized database dumps (up to 2Mb), we find that phpMyAdmin [http://www.phpmyadmin.net] does the trick pretty well, plus it's installed on virtually all PHP enabled commercial hosts. For larger dumps, we found that bigdump.php from Alexey Ozerov [http://www.ozerov.de/bigdump.php] works wonders. Another useful and very easy (or, should I say, easier) to use tool is Adminer [http://www.adminer.org/]. Use either of those tools - or any other of your liking - to restore your database.

Step 5. Upload your site's files.

First of all, delete the installation subdirectory from the directory you extracted the backup archive to. We won't be needing this any more. Then, using FTP - or any method you please - upload all of the files to the target server.

If you want to be thorough remember to set the directory and file permissions accordingly. If you just want to get the damn thing on-line ASAP, just skip this permissions thing; it will remind you of itself as soon as you try to do some website administration (like uploading a picture) after the site's back on-line.

Step 6. Edit configuration.php, if necessary.

If you were restoring to the same server location you took the backup on, nothing else is necessary. Your site should be back on-line now. If not, you'll have to edit the `configuration.php`.

You have Joomla! 1.5.x. Good news! Joomla! 1.5.x doesn't require you to specify some of the hard-to-obtain parameters. Your `configuration.php` consists of several lines. Each one is in the following form:

```
var $key = "value";
```

The key is the name of the configuration variable and value (inside double quotes!) is the value of the variable. Below we provide a list of the configuration variables which have to be modified to get up on-line.

dbtype — is the database driver Joomla! will use. It can be mysql, mysqli (notice the extra i in the end) or pdomysql. This depends on the kind of database you are using. If unsure, your best bet is mysqli.

host — is the database host name, usually localhost

user — is the database user name, assigned from your host company

password — is - obviously - the database password, assigned from your host company

db — is the database's name, assigned from your host company

dbprefix — is the database prefix; if you followed our instructions, it is jos_

live_site — Normally this is an empty string. If, however, your Joomla! site's front page looks as if all images and CSS files are not loading, you have to modify it and enter your site's base URL. For example, if the new site is located in `http://www.example.com/mysite/`, you have to locate the line starting with `var $live_site` and change it to become:

```
var $live_site = "http://www.example.com/mysite";
```

That's all! You're good to go.

Step 7. Enjoy success.

Your mission is accomplished. You are exhausted. Go drink whatever is your favourite drink and enjoy sweet success!

Chapter 6. Information for removed or canceled features

Sometimes, despite our best intentions, some features prove to either be far less than ideal or impossible to create in a way that makes practical sense. The reasons behind the decisions to remove or cancel features may not be obvious. These articles try to shed some light.

1. Microsoft OneDrive for Business

This article was written June 26th, 2018.

Executive summary

Executive summary: Microsoft's convoluted and self-contradicting documentation promised that this feature would be possible and it was (mostly) working when we released it. Soon after they removed this functionality and required the use of a different API to access OneDrive for Business. However, the new API is rate-limited, meaning that you can only make a handful of API calls every ten minutes. This makes it impossible to upload backup archives in small chunks to prevent your server from timing out. Therefore the new API is unsuitable for the purpose of storing backups. We had no option but to cancel this feature.

We might reconsider this decision if Microsoft Graph lifts the API rate limiting or increases it to practical levels.

Technical information

Despite the deceptively similar name, Microsoft OneDrive for Business (a feature of Office 365 for Business, let's call it 1DB for short) has nothing to do with Microsoft OneDrive (the consumer storage solution, let's call it 1D for short). The differences are in both authentication and API.

1D works with any individual Microsoft account. These accounts can be opened free of charge with just an email address. If you had a Hotmail, Outlook.com or Skype account it's already a Microsoft account. As part of your Microsoft account you get free 1D storage, no matter if you want it or not. If you buy a personal Office 365 subscription (NOT the Business one!) you get even more space on your 1D.

An application can talk to 1D by authenticating the user against the Microsoft Account using OAuth2 and using the returned token to access the OneDrive API (formerly: Windows Live API). This is straightforward and we have supported this since several years ago.

1DB is an entirely different beast. You cannot get it for free. Your organization or school (the Organization) needs to purchase an Office 365 For Business subscription. These come in different tiers. All tiers but the cheapest one allow the Organization's users to have 1DB storage.

The authentication for 1DB storage does not happen against a regular Microsoft account. Since 1DB is part of the Organization's Office 365 For Business plan the authentication has to happen against the Organization's accounts. These are special Microsoft account which only work within the Organization. The Organization is referred to as a "tenant" in Microsoft's documentation.

Regular OneDrive applications are not aware of "tenants". Therefore they can only authenticate regular Microsoft accounts through OAuth2. Now starts the real confusion. Microsoft's documentation indicates that there are two and a half ways to overcome this.

The first way is to create a Microsoft Azure Active Directory application for the tenant. This means that for every organization which would like to use Akeeba Backup or Akeeba Solo we would need to create a new authentication endpoint on our server and a different upload engine, maintaining them forever. This is impractical. If we had, say, 100 organizations using our software we'd need to create 100 different packages for each version of Akeeba

Backup / Akeeba Solo and create 100 update documents on our servers, maintain 100 different authentication endpoints and so on. This would require us to charge more 4,000 Euros per year for these clients, not 40 Euros per year, and have an army of code monkeys to maintain these integrations. This makes no economic sense, primarily for our clients: using Amazon S3 or BackBlaze or *anything* would still be cheaper for them!

The one and a half way is to create a multi-tenant Microsoft Azure Active Directory application. Following Microsoft's documentation it's absolutely clear that this no longer works. In fact, Synology seems to be using the same solution because when I (Nicholas, the lead developer) tried to use their OneDrive backup solution on my Synology NAS I got the same error I was getting when I was trying to do authentication with this method. I guess if a multi-million-dollar company can't figure this out either I shouldn't feel bad about my developer skills... But I digress. I was wondering why this happens. Digging around I figured it's because...

...Microsoft now offers a second way, creating a "converged app" on the same platform that the old, regular-Microsoft-Account-only apps were created. Well, OK. That's what we did.

The other problem we have to solve is which API we can use to talk to 1DB. Microsoft claims we have two options.

First, they offer the OneDrive API (formerly Windows Live API) which is the traditional way to talk to OneDrive. This is what we use for our consumer OneDrive integration. According to Microsoft we can use this to access 1DB as long as we have the special endpoint for the tenant our user belongs to.

The second option is Microsoft Graph which tries to integrate all Microsoft services' APIs under a unified umbrella. It can be used not only to upload stuff to 1DB but also give us the all important information we need for the OneDrive API: the special endpoint for the tenant of the user we just authentication.

So, following Microsoft's convoluted and self-contradicting documentation we have now a token we can use with Microsoft Graph API to get the OneDrive API endpoint for the tenant of the user we logged in. That works and an endpoint is indeed returned. This endpoint is then used with the OneDrive API to talk to the user's 1DB storage.

The latter part worked when we launched the integration. A month later we started receiving reports that the integration was not working. Indeed, creating a fresh authentication token no longer worked but the old authentication token we had created a month earlier worked!

That's of course because Microsoft changed something in their APIs *AGAIN*. Each "converged" application has two sets of permissions, "delegated" and "application". Delegated permissions are given to the application when a user logs in and cannot exceed the permissions the user has. These are permissions like "create files", "read the contents of files" and "let me store the token so I don't have to ask the user before every action". Application permissions must be given by an administrator of the tenant and are applied regardless of who actually logs in. Microsoft's documentation explains that we should be using delegated permissions with the OneDrive API and indeed that *used* to work. What happened recently is that the "legacy" OneDrive API was applying Application permissions, not Delegated permissions, at least for new tokens. This meant that we had to make a change in our "converged application" but these changes only applied if an administrator re-authenticated our backup software.

In short: if you were given an 1DB account from your organization or school and you were not an administrator there you could NOT use it. This is not an integration that makes sense; we couldn't release that crap and claim that we support OneDrive for Business! So we tried to find a solution.

Enter day 2 of debugging (that's on top of 3 days of developing the feature in the first place).

Remember Microsoft Graph API? Microsoft is trying to shove it down developers' throats so we said what the heck, let's try it. The first hurdle is that Microsoft's documentation for the Graph API is *wrong*. The information they claim is returned is not. The parameters to call it are not what they document. So, implementation has to be the same as with all other Microsoft's APIs and their shoddy documentation: trial and error. Here comes the second hurdle. Microsoft Graph API is rate limited. You can only do about 5 requests every ten minutes. At this point it was very clear that this meant two things. First, developing anything would take weeks just because the documentation is wrong and trying to figure things out by educated guesses would require looooooooooooooooooooooooooooooooooong breaks between actions. Long enough to make a developer lose his train of thought and have to start over, adding an overhead of 20' every time. Second, uploading a backup archive in small chunks to prevent a server timeout requires much more than 5 requests in 10 minutes. So this method would fail if we were to use it for storing backups.

So, it ends up that Microsoft has made it so that OneDrive for Business is impossible to use efficiently for storing backups UNLESS you concede that only organization administrators should be able to set up OneDrive for Business as backup storage. This is absurd. Also, Microsoft's behaviour the last 10 years shows that their APIs change drastically every year, making this a very expensive proposition both for us and the businesses using our software. At some point we will drop old Joomla and PHP versions. If your site still runs on them, using an older version of our software, and Microsoft changes their APIs -which they definitely will!- you are out of luck.

Part II. Security information

Table of Contents

Chapter 7. Introduction

1. Foreword

Since you have chosen Akeeba Backup for backing your site up, it is obvious that you are using Joomla!™ as your web-based Content Management System. By using Joomla!™ you have embarked to the joyful adventure of managing a PHP powered website. Usually, this last part is gone unnoticed. The fact that you are using a PHP application is often taken for granted, but when it comes down to security and problem solving, this is the key concept of which you should have a strong grasp.

This part of the documentation deals with the basic concepts of PHP website management and their implications upon using Akeeba Backup. In this part, we will see the intricacies of access permissions, web site users and the impact of various PHP settings on your site's operability and security. This is not meant to be a concise manual on website administration. There are plenty of web and off-line resources with more in-depth information on the subject, but this introduction will quickly get you up to speed.

This document is no light reading; it is purposely sprinkled with a lot of tech-talk, albeit explained in layman's terms. Our objective was not to write a document which can be read and understood in a single reading. Some things you will understand by the first time you'll have read it. Most of it you will only get it after reading it again. A few shady areas will only become clear reading over again and referring to it every time you get stuck managing your site.

2. Why you need to care about ownership and permissions?

Most probably your server is running on Linux™, or another UNIX™-derivative operating system. You might have read, or heard, how these operating systems are safer and more secure than others. This is just half the story. The real security power of such operating systems stems from the way they manage files and directories, allowing or disabling access to them depending on who asks for it and what he's trying to do.

This management is pretty much like electricity in the Western world. It never gets in your way and you don't think about it, but you must have some basic understanding of it so as not to run the risk of getting toasted by it. That's how it goes with ownership and permissions. You might not think about them a lot, but potentials crackers do. If you don't manage permissions wisely, you might be creating a security hole on your server which can be exploited by a malicious cracker. Nobody wants his site cracked, right?

The following chapter will analyze how your web server works under the hood, so that you can grasp the third chapter, which analyzes all the ways you can secure your backup files so as not to fall prey on a cracker.

Chapter 8. How your web server works

1. Users and groups

The concept of users is the fundamental block of ownership separation on multiuser operating systems. All Windows™ versions based on the NT kernel are such; Windows™ NT, 2000, XP, Vista are all multiuser operating systems. Other UNIX variants are also inherently multiuser, including Linux™ , BSD™ flavours, MacOSX™ , etc. Since most web servers capable of running Joomla!™ are based on Linux™ , we will talk about the Linux™ user system, which is in fact the same as the UNIX user system; after all, GNU/Linux is nothing but an open-source UNIX variant which became very popular among geeks and recently among other people, too.

1.1. Users

As we mentioned, the fundamental block of ownership separation is a *user* . Each user has an entry in the system's password database and consists of a *user name* and a numeric *user ID* . A user is not necessarily linked to a physical person; in fact, most utilities and services create and operate under a user of their own.

The numeric user ID is an unsigned integer, therefore it can take a value between 0 and 65534. The user name and the numeric user ID are usually linked with an one to one relationship, meaning that if you know either one you can find the other one. The exception to this is most ISPs. In this case, because there are more users than the available number of user IDs, some numeric IDs will be reused, breaking the one to one relationship. However, on most - if not all - hosts, the one to one relationship exists.

Some user IDs are special. By convention, user IDs below 500 are reserved for system users. These are special users which are not assigned to some physical person. One of them, zero (0), has a very special meaning; it is assigned to the *super user* , commonly called *root* . This user is the God of the system. He has unlimited powers. He can override all access restrictions and make any kind of modification. For this reason, no sane system administrator logs in under that user. They will always log in under a normal user and only temporarily log in as root whenever they need to change system-wide settings.

1.2. Groups

Defining permissions per user is tiresome on systems which have more than a few users. In order to combat this inconvenience, all UNIX systems have the notion of *groups* . A group is nothing but a collection of users. The relationship to users is a many-to-many relationship, meaning that one user can belong to many groups and one group can contain many users. To keep things dead simple, groups have the same format as users. Each group has a *group name* and a numeric *group ID* . Again, not all groups are linked to a physical person; in fact there are a number of de facto group names used to control access to crucial system resources.

The numeric group ID is an unsigned integer, therefore it can take a value between 0 and 65534. The group name and group ID are linked with an one to one relationship, meaning that if you know either one you can find the other one. I am not aware of exceptions to this rule and I can't think a reason, either.

There are some special group ID's. By convention, zero (0), is assigned to the root's group. Its sole member should be root, or other users with a user ID of 0. It empowers its members to do anything they please on the system, almost like the user ID 0 does. Noticed the "almost" part? Belonging to the root group alone, without having a user ID of 0, does not give you infinite powers but it *does* grant you very broad access indeed!

Every user can belong to many different groups. To simplify things a little bit, every user has a so-called default group. This means that one of the groups he is a member of will be his effective group, unless otherwise specified, in all operations.

1.3. How users and groups are understood by UNIX-derived systems

This section is a bit ahead of the rest of this chapter, I know that. The information contained here, though, clarify a lot of what will follow, so it seemed only appropriate to include it here.

Every time the system has to store the owning user and group of a system item, it does so by storing the numeric user and group IDs, not the names! The names are only used as a convenience; you can't remember that John's user ID is 637, but it's easy to remember that his user name is john. Likewise, remembering that group ID 22 controls access to the CD-ROM drive is improbable, while remembering that the group named cdrom does that is self-understood.

Important

User IDs for a user with the same user name on different systems can be different. A user named example on system A and system B might have one user ID on system A and a completely different on system B. However, all UNIX-derived systems really know about are IDs, not names!

This is very (read: extremely) important when you transfer files from one system to another. All archive types which store owner information (for example GNU `tar`) store nothing but the numeric ID's. Moving these to another system and extracting them will screw up ownership and permissions. Just because you have the user ID 567 on Host A doesn't mean that you won't end up with user ID 678 on Host B; extracting such an archive would make all your files owned by someone else, effectively screwing up your site.

2. Ownership

The term *ownership* implies that system items belong to someone. In the context of web site management the items we are interested in are files and *processes* . Everybody understands what files are, but the term *processes* is rarely understood amongst webmasters. So, let's explain it.

2.1. Process ownership

Every time you run a program, be it interactive or a system service, you create a process. A process is a piece of code being executed by the operating system. A process can *spawn* child processes which can spawn new *threads* . In layman's terms, a program can start other instances of itself or another program and they, in turn, can start small pieces of executable code which can run in parallel with the main program.

Programs do not start spontaneously. Someone has either got to start them, or instruct the system to start them when some criteria are met. This sentence is the acknowledgement of the simplicity behind a computer system; it can't think on its own, humans have to tell it what to do one way or the other. Based on how a program starts, it process will be owned by some user.

In the first and simplest case, when you start a program, the ownership is almost self-understood. You are logged in as some user, so the process of the program you have executed is owned by your user. It's simple as that. This also implies that the process has the same permissions as the owning user, that's why we say that the process runs *under* this user; its access level is at most as much as the owning user, so the process is *under* the user.

The other case, instructing the system to start a process, is somewhat different. Usually, the utilities which are used to start programs automatically are the system initialisation scripts, time-based execution programs (for example, `cron` and `at`), etc. All of these programs are in most cases owned by root and are executed under root privileges. On top of that, most programs started this way are system services, running as long as the system is up and running. But do you remember what we said before? Root is the God of the system. Normally, these programs would get root's privileges, posing a huge security hole. If there is a bug in the program and some malicious user exploits it, he could wreck havoc on the system; root is above all restrictions.

In order to combat this possibility, UNIX systems employ a feature which allows processes to *drop privileges* and run under a different user than the one which started them. In fact, they change their ownership! To prevent

abuse of this feature, a process must run under root privileges to be able to switch to another user. This feature is extensively used by system services, including MySQL and Apache.

In the context of web site management, Apache is of special interest. Apache is the de facto web server for Linux systems and is being used by over 50% of Internet sites, according to NetCraft's August 2008 survey. Chances are you are using it on your site, too. Apache, like most UNIX services (affectionately called *daemons*) uses the feature to drop privileges. The user and group under which it runs are defined in its configuration files. These configuration files are usually out of the reach of regular users (like you!) on commercial hosts, for security reasons.

There is a **special case** which acts as the exception to the Apache rule. Many commercial hosts run **suPHP**-enabled Apache installations. This is an extension to the normal PHP's mode of operation which allows each PHP page to run in a process owned by the file's owner (more on file ownership in the next sub-section). This means that each of the PHP files under your account on such a host run as the user which has been assigned to your account. And, if this still isn't apparent to you, such hosts nullify the burden of ownership and permissions (more on permissions in the next section). To put it clearly: with suPHP the file owner, your own user and the Apache user are one and the same. If you are looking for a decent host, find one which is using suPHP. It's better for security and removes a lot of administrative burden from you. A win-win situation.

2.2. File ownership

Everybody knows what a file is, right? Well, we all know intuitively what a file *might* be, but we seldom know what *exactly* it is. A file is actually consisted of at least two parts. The first part is the file data, what we intuitively understand as the file contents. The second part is the file system entry, which makes the file data an identifiable entity. This is where the operating system stores all kinds of information, such as how the file is named, where it is located in the file system hierarchy, when it was modified, etc. It also contains information about who owns the files and what are the file's permissions. You might be surprised reading this, but only this latter, informative, part is required for a file. Really!

It seems absurd to have a file without file data, but it is anything but that. There are some special "files" (more correctly: file system entries) in the UNIX world. You have devices, whose "files" actually point to a serial input/output provided by this device, for example the serial port of your computer. There are directories, which obviously don't have any data contained; they are used for organising files only. There are soft links, which are pointers to other files in the file system, used to have standardised names and locations on files which might be moved around or have varying names. There are also these wired beasts called "hard links", some peculiar file system entries which point to the file data of another file, making virtually impossible to know which is the "original" file and which is its clone. Their usefulness is only apparent to the UNIX gurus, therefore out of the scope of this document. For the purpose of website management we are only concerned about regular files (hereby called "files"), directories and soft links (hereby called "links").

All files, directories and links are owned by a user and a group, be they files or links. In fact, they are owned by a user ID and a group ID. Normally, the ownership is inherited by the creating process's ownership. When you create a file directly from an interactive editor application the editor's process is owned by your user ID and your default group ID, therefore the file will be owned by your user ID and your default group ID.

Links are a special case on their own. They are not files, they are pointer to files. The ownership (and permissions) of links is irrelevant. Whenever a process tries to access a link, the underlying operating system "follows" the link, until it finds a regular file. Therefore, the ownership that matters is that of the file linked to, not the link itself. This feature of the operating system prevents unauthorised access to arbitrary files, normally accessible to specific users only, from users who just happen to know the path to those files.

What is especially interesting is the correlation between FTP, web server and file ownership. Whenever you access FTP, you log in as some user. This user is linked to a system user (often the same user assigned to you by host), so logging in FTP actually has the same effect as logging into the system as this user. Common sense implies that all file operations are performed under this user and all files created (read: uploaded) through FTP will be owned by this user.

Conversely, whenever you are using a web interface to perform file operations, you are using a web application - or any PHP script/application for that matter - running on the web server whose process is owned by a different

user. Therefore, whenever you create files from a web application, they will be owned by the user the web server runs under.

The distinction of file ownership in these two cases is of paramount importance when you get stuck with files which are accessible to FTP but inaccessible to the web server, or vice versa. This minute distinction is the cause of a lot of grief to many webmasters, so beware!

3. Permissions

So far you have learned about users, groups and ownerships. But how do they all stick together? Why these are necessary to have in the first place? The reason is simple: security. In multiuser operating systems you normally don't like users snooping around other people's files, especially when those files contain sensitive information, such as passwords. The most common method for overcoming this problem is to assign *permissions* on each system item, controlling who can do what. This simple concept works wonderfully; it's like putting doors on a building and giving people only the keys for the doors to areas they should have access to.

3.1. The three types of permissions

We already learned that each system item is owned by a user ID and a group ID. Whenever a process tries to access a system item, the operating system checks the permissions and decides if it will proceed with the operation or deny access. It seems reasonable to have control over what a process with the same owning user ID can do with it, what the a process with the same owning group ID can do with it and, finally, what the rest of the world can do with it. Indeed, this is the rationale behind the three types of permissions we can define on UNIX systems. In order of precedence they are:

User permissions	They are the access rights granted to the owning user of the item. Every process with the same owning user ID as the item's owning user ID has these access rights. These access rights have precedence over all other permissions.
Group permissions	These are the access rights granted to the owning group of the item. Every process with the same owning group ID as the item's owning group ID has these access rights. These access rights are applied only if the owning user ID's of the process and the item do not match, but their owning group ID's match.
Other permissions	These are the access rights granted to the rest of the world. If the owning user ID's of the process and the item do not match and the same happens for the owning group ID's as well, these access rights will be applied.

3.2. What permissions can control

We will be focused on permissions on files and directories, the building blocks of a web site. Permissions can control only three different actions:

Read	The ability to read a file, or get a directory listing.
Write	The ability to write to a file, or the ability to create, rename and delete files and subdirectories on a directory.
Execute (or Browse, for directories)	For files, it controls the ability to be directly executable from the command line. It is only meaningful for binary programs and executable scripts. For directories, it controls the ability to change to that directory. Note that if this is disabled you can't usually obtain a directory listing and file read operations might fail.

These three actions, combined with the three access request groups (owning user, owning group and the rest of the world) give us a total of nine distinct operations which can be controlled. Each action is an on/off switch. If a permission is set, it is turned on and the right to perform the action is granted. If the permission is not set, the switch is off and the right to perform the action is not granted.

3.3. Permissions notation

The two most common notations for permissions is the *textual notation* and the *octal notation*. Each one has its own virtues.

3.3.1. The textual notation

The textual notation is traditionally used in UNIX long directory listing format and in most FTP clients listings as well. It consists of ten characters. The first one displays the file type. It can be one of dash (regular file), "d" (a directory) or "l" (a link). The following nine characters display the permissions, consisting of three groups of three letters each. The groups are in order of appearance: owning user, owning group and others. The permissions on each group are in order of appearance: read (denoted with r), write (denoted with w) and execute/browse (denoted with x). If a permission is not set, a dash appears instead of the letter.

For example, the string `-rwxr-xr-x` means that it is a regular file, the owning user has read/write/execute permissions, the owning group has read and execute permissions and so does the rest of the world. On the other hand, the string `dr-x------` indicates that we have a directory whose owning user has read and browse permissions and everybody else (owning group and the rest of the world) have no right to access it.

3.3.2. The octal notation

This is the de facto standard geeks use to communicate permissions. The benefit of this approach is that you only need four characters to fully define them and they're easier to read (to the trained eye, at least).

Permissions are in fact a bit field. Each permission is a bit which can be turned on or off. If you put bits together they form bytes (by grouping eight bits together). Many bytes one next to the other form a computer-readable representation of a whole number (an integer). If you write this down in base 8, you've got the octal representation. If you didn't understand this, it's OK. We'll explain it the easy way.

The octal notation consists of four numbers. In the context of web site management you can consider the first to be always zero and sometimes omitted. The next three numbers describe each one the permissions. The second number describes owning user permissions. The third number describes owning group's permissions. The fourth number describes the permissions for the rest of the world. Each number is 0 to 7. The meaning of each number is simple:

0 No access

1 Execute/browse access only

2 Write access only

3 Write and execute/browse access

4 Read access only

5 Read and execute/browse access

6 Read and write access

7 Full access

It is almost apparent that "1" stands for execute only, "2" stands for write only and "4" stands for read only. Adding these values together gives you the rest of the combinations. You can't add together the same value (1+1 is forbidden as it is meaningless), so each of the composite values can be broken down to its components very easily. You don't even have to memorise the whole table!

A permission of 0777 means that the owning user, owning group and the rest of the world can read, write and execute the file (full permissions for everyone). A 0764 permission means that the owning user has full access, the owning group has read and write access and the rest of the world have read only access.

Chapter 9. Securing your Akeeba Backup installation

1. Access rights

As with every software which can access your site as a whole, Akeeba Backup needs to control who's got access to its backup functionality. Akeeba Backup fully supports Joomla's access control features, allowing you to set specific permissions for specific user groups. You can change this behavior from the component's Options button in the Control Panel page - just like with any other Joomla! component.

The front-end backup feature is a different story. Since it has to be available to unattended scripts which can't use cookies and interactive user authentication, a different approach was taken. Instead of requiring the user to have logged in with Joomla! it uses a simple "secret word" authentication model. Because this "secret word" is transmitted in clear text we strongly advise against using it over anything else than a local network (for example, an automated tool running on the same host as the web server). If you have to use it over the Internet we strongly advise using a secure protocol connection (HTTPS) with a valid commercially acquired certificate.

2. Securing the output directory

Securing the backup output directory

By default the component uses a non secure location to store its backup files and temporary files, within your site's file system hierarchy, namely `administrator/components/com_akeeba/backup`. This location is well known and can be - theoretically - accessed directly from a web browser. Since the backup output directory stores the results of your backup attempts, that is SQL files containing database backups and archive files containing all of your site, a malicious person with access to this location could steal sensitive information or compromise your site's integrity.

The first line of defense, is to use mangled, hard to guess, names for the SQL backup. However, it wouldn't take an attacker that long to figure out the filename. Remember: security through obscurity is no security at all!

As a second line of defense, we include a secure `.htaccess` on the default backup output directory to disable direct web access. However, this is only possible on Apache-powered web servers which allow the use of `.htaccess` files. You should check with your host to ensure that this kind of protection is possible on your site.

However, this is not enough. Using a well known location would allow an attacker exploiting a security issue in a third party component to gain access to the backup archives. The only way around that is using a different directory, ideally one above your site's root.

3. Securing file transfers

Whenever you download your backup files you can fall prey to a malicious user. Backup files are transferred unencrypted (unless you access your site's administrator section through the HTTPS protocol). It is possible for a resourceful hacker to launch a man-in-the-middle attack. In such a case, whatever you download from your site will be directed to the hacker's computer before reaching yours.

To avoid such insecure scenarios, we advise against using the Download button in the backup administration page. We suggest that you use Secure FTP (SFTP) instead. Avoid using the plain old FTP, because your password and data are transmitted in clear text (unencrypted) over the Internet. Also avoid FTPS and FTPES (FTP over SSL) as they have some security restrictions, like requiring your FTP server to have a commercially obtained SSL certificate in order to be really effective. Sometimes, your host will allow secure access to a web based control panel which has a file download feature. You could use this, it's as safe as it gets.

There is also another reason why not to use the Download button in the backup administration page. Your host neither discriminates the back end and front end pages of your Joomla! site, nor your IP from the rest of the world. As a result, every time you use the Joomla!™ back end, the data transferred counts towards your monthly bandwidth quota. Backup archives are large, sometimes in the hundreds of megabytes. Transferring them through the Download feature will incur a huge loss on your monthly bandwidth quota. Using Secure FTP or your host's control panel *usually* does not count through the bandwidth quota and should be used instead. It's better to ask your host, though; some include the FTP and SFTP traffic in your monthly bandwidth quota. Finally, the Download feature doesn't work with all possible configurations and has objective problems with the handling of very large archives; this is a technical limitation which can not be overcome in the PHP level the component operates. Most notably, many servers which use the FastCGI mode do not work at all with the Download button. They will simply throw an HTTP 500 error page, or a "file not found" message. We've tried all the tricks in the book and then some more, but there's really absolutely nothing we can do about it. Sorry.

Important

The preferred and suggested method for downloading your backup files - for several reasons - is using FTP in BINARY mode, preferably over an encrypted connection. Alternatively, you can use Remote CLI which allows you to use this approach when downloading backup archives.

Part III. Appendices

Table of Contents

Appendix A. The JPA archive format, v.1.2

Design goals

The JPA format strives to be a compressed archive format designed specifically for efficiency of creation by a PHP script. It is similar in design to the PKZIP format, with a few notable differences:

- CRC32 is not used; calculation of file checksums is time consuming and can lead to errors when attempted on large files from a script running under PHP4, or a script running on PHP5 without the hash extension.
- Only allowed compression methods are store and deflate.
- There is no Central Directory (simplifies management of the file).
- File permissions (UNIX style) are stored within the file.

Even though JPA is designed for use by PHP scripts, creating a command-line utility, a programming library or even a GUI program in any other language is still possible. JPA is not supposed to have high compression rations, or be secure and error-tolerant as other archive formats. It merely an attempt to provide the best compromise for creating archives of very large directory trees using nothing but PHP code to do it.

This is an open format. You may use it in any commercial or non-commercial application royalty-free. Even though the PHP implementation is GPL-licensed, we can provide it under commercial-friendly licenses, e.g. LGPL v3. Please ask us if you want to use it on your own software.

Structure of an archive

An archive consists of exactly one Standard Header and one or more Entity Blocks . Each Entity Block consists of exactly one Entity Description Block and at most one File Data Block . All values are stored in little-endian byte order, unless otherwise specified.

All textual data, e.g. file names and symlink targets, must be written as little-endian UTF-8, non null terminated strings, for the widest compatibility possible.

Standard Header

The function of the Standard Header is to allow identification of the archive format and supply the client with general information regarding the archive at hand. It is a binary block appearing at the beginning of the archive file and there alone. It consists of the following data (in order of appearance):

Signature, 3 bytes	The bytes 0x4A 0x50 0x41 (uppercase ASCII string “JPA”) used for identification purposes.
Header length, 2 bytes	Unsigned short integer represented as two bytes, holding the size of the header in bytes. This is now fixed to 19 bytes, but this variable is here to allow for forward compatibility. When extra header fields are present, this value will be 19 + the length of all extra fields.
Major version, 1 byte	Unsigned integer represented as single byte, holding the archive format major version, e.g. 0X01 for version 1.2.
Minor version, 1 byte	Unsigned integer represented as single byte, holding the archive format minor version, e.g. 0X02 for version 1.2.
File count, 4 bytes	Unsigned long integer represented as four bytes, holding the number of files present in the archive.

Uncompressed size, 4 bytes	Unsigned long integer represented as four bytes, holding the total size of the archive's files when uncompressed.
Compressed size, 4 bytes	Unsigned long integer represented as four bytes, holding the total size of the archive's files in their stored (compressed) form

Extra Header Field - Spanned Archive Marker

This is an optional field, written after the Standard Header but before the first Entity Block, denoting that the current archive spans multiple files. Its structure is:

Signature, 4 bytes	The bytes 0x4A, 0x50, 0x01, 0x01
Extra Field Length, 2 bytes	The length of the extra field, without counting the signature length. It's value is fixed and equals 4.
Number of parts, 2 bytes	The total number of parts this archive consists of.

When creating spanned archives, the first file (part) of the archive set has an extension of .j01, the next part has an extension of .j02 and so on. The last file of the archive set has the extension .jpa.

When creating spanned archives you must ensure that the Entity Description Block is within the limits of a single part, i.e. the contents of the Entity Description Block must not cross part boundaries. The File Data Block data can cross one or multiple part blocks.

Entity Block

An Entity Block is merely the aggregation of an Entity Description Block and at most one File Data Block. An Entity can be at present either a File or a Directory. If the entity is a File of zero length or if it is a Directory the File Data Block is omitted. In any other case, the File Data Block must exist.

Entity Description Block

The function of the Entity Description Block is to provide the client information about an Entity included in the archive. The client can then use this information in order to reconstruct a copy of the Entity on the client's file system. It is a binary block consisting of the following data (in order of appearance):

Signature, 3 bytes	The bytes 0x4A, 0x50, 0x46 (uppercase ASCII string “JPF”) used for identification purposes.
Block length, 2 bytes	Unsigned short integer, represented as 2 bytes, holding the total size of this Entity Description Block.
Length of entity path, 2 bytes.	Unsigned short integer, represented as 2 bytes, holding the size of the entity path data below.
Entity path data, variable length.	Holds the complete (relative) path of the Entity as a UTF16 encoded string, without trailing null. The path separator must be a forward slash (“/”), even on systems which use a different path separator, e.g. Windows.
Entity type, 1 byte.	• 0x00 for directories (instructs the client to recursively create the directory specified in Entity path data). • 0x01 for files (instructs the client to reconstruct the file specified in Entity path data) • 0x02 for symbolic links (instructs the client to create a symbolic link whose target is stored, uncompressed, as the entity's File Data Block). When the type is 0x02 the Compression Type MUST be 0x00 as well.
Compression type, 1 byte.	• 0x00 for no compression; the data contained in File Data Block should be written as-is to the file. Also used for directories, symbolic links and zero-sized files.

- 0x01 for deflate (Gzip) compression; the data contained in File Data Block must be deflated using Gzip before written to the file.
- 0x02 for Bzip2 compression; the data contained in File Data Block must be uncompressed using BZip2 before written to the file. This is generally discouraged, as both the archiving and unarchiving scripts must be ran in a PHP environment which supports the bzip2 library.

Compressed size, 4 bytes	An unsigned long integer representing the size of the File Data Block in bytes. For directories, symlinks and zero-sized files it is zero (0x00000000).
Uncompressed size, 4 bytes	An unsigned long integer representing the size of the resulting file in bytes. For directories, symlinks and zero-sized files it is zero (0x00000000).
Entity permissions, 4 bytes	UNIX-style permissions of the stored entity.
Extra fields data, variable length	The extra fields for each file are stored here. The total length of extra fields is included in the Block Length above

Each Extra Fields consists of:

Extra Field Identifier, 2 bytes	A signature denoting the data stored in the extra field
Extra Field Length, 2 bytes	The length (in bytes) of the Extra Field Data
Extra Field Data, variable length	The internal structure varies by the type of the Extra Field, as noted in the Extra Field Identifier

Timestamp Extra Field

Its purpose is to store the date and time the file was modified. This extra field should be ignored for directories and symlinks, or - if present - the Timestamp should be set to 0x00000000. Its format is:

Extra Field Identifier, 2 bytes	The bytes 0x00 0x01
Extra Field Length, 2 bytes	The value 0x08 stored in little-endian format
Timestamp, 4 bytes	A 4-byte UNIX timestamp of the file's modification time, as returned by filemtime().

File Date Block

The File Data Block is only present if the Entity is a file with a non-zero file size. It can consist of one and only one of the following, depending on the Compression Type:

- Binary dump of file contents or textual representation of the symlink's target, for CT=0x00
- Gzip compression output, without the trailing Adler32 checksum, for CT=0x01
- Bzip2 compression output, for CT=0x02

Change Log

Revision History

June 2009	NKD, Akeeba Developers	http://www.akeebabackup.com

Updated to format version 1.1, fixed incorrect descriptions of header signatures

Appendix B. The JPS archive format, v.2.0

Design goals

The JPS format strives to be a compressed archive format designed specifically for efficiency of creation by a PHP script, while providing secure AES-128 encryption of the file descriptor and file contents. It is similar in design to the JPA, with a few notable differences:

- Both the file descriptor and the file data are split to 64Kb blocks encrypted using Rijndael-128 in CBC mode (that's the same as AES-128)
- All files are compressed using Deflate (ZLib)

Even though JPS is designed for use by PHP scripts, creating a command-line utility, a programming library or even a GUI program in any other language is still possible. JPS is supposed to have low to medium compression rations, and be secure. However it is not as error-tolerant as other archive formats.

This is an open format. You may use it in any commercial or non-commercial application royalty-free. Even though the PHP implementation is GPL-licensed, we can provide it under commercial-friendly licenses, e.g. LGPL v3. Please ask us if you want to use it on your own software.

Important

When the password is blank, no encryption takes place. Archivers should take this into account when creating files. Unarchivers should also take this into account when the user passes an empty string as their password.

When a non-blank password is used, all files are encrypted using the same password. More specifically, all data blocks are encrypted using the same password.

Security

The security of the format largely hinges on the assumption that Rijndael-128 in CBC mode with randomized IVs in each encrypted stream is not susceptible to KPA (known plain-text attacks). Should a KPA be found against the encryption algorithm the obvious crib would be the encrypted file header of the first file in an Akeeba Backup / Akeeba Solo archive which is very predictable. Even if the order of files were randomized, there are well-known files (part of the installer) with known contents, making them relatively easy to identify by their relative size in the archive. However, as we said above, the encryption algorithm is not known to be susceptible to KPA, nullifying this threat.

Another defense you can use when creating the archive is the use of a non-static salt for PBKDF2 key expansion. This means that the cryptographic key which could theoretically be brute forced by means of a KPA would only apply to a specific encrypted block. It would then take another, more computationally expensive, brute force attack against the password to decrypt the entire archive. The downside is that this is a much slower encryption method since a key needs be derived for every encrypted block of data. Counter-intuitively this could lead to worse security since the practical considerations of the implementation lead to using a much smaller number of iterations with a weaker hashing algorithm which may end up being easier to brute-force, especially for the shorter passwords.

Our recommendation for v2.0 archives is using key expansion with a static salt, a high number of iterations (e.g. 64000) and a strong hashing algorithm (e.g. SHA512).

Key Expansion

JPS v.2.0 (PBKDF2)

JPS v.2.0 is using a different, more secure, key expansion scheme that JPS v.1.x. PBKDF2 is used on the user-supplied password to generate the encryption key. PBKDF2 was selected over memory-hard algorithms (like bcrypt, scrypt, Argon2 etc) for performance reasons, considering that encryption has to also take place on shared hosts with limited resources and old versions of PHP which don't even support these newer hashing algorithms. As processors get faster and old PHP versions become increasingly obsolete we might revise the key expansion algorithm in the future.

The supported PBKDF2 algorithms at this time are SHA-1 (used by default), SHA-256 and SHA-512. The algorithm used throughout the archive is specified in the archive header. Even though SHA-1 is not collision-resistant, the high number of iterations mitigates that risk.

The number of iterations used throughout the archive is also specified in the archive header. By default it's 100,000. This is a moderately high number of iterations while still being practical on resource-limited shared hosting.

There are two possibilities for the salt used for PBKDF2. One possibility is using a static salt, found in the archive's header. In this case you only perform key expansion once and use the expanded key for all encrypted blocks in the archive. The other possibility is having a different salt per encrypted block. In this case a key expansion is executed per encrypted block, therefore using a different encryption key for each block.

Important

We STRONGLY recommend using long (64 or more characters), completely random passwords which make use of lowercase and uppercase Latin letters, numbers and special characters (top row on US-format keyboards). Use a password manager to generate and store these passwords. Taking these precautions make password brute forcing with conventional technology highly impractical in the foreseeable future.

JPS v.1.x (Rijndael-128 CTR)

All JPS v.1.x format use a very naive key expansion, based on Rijndael-128 running in CTR (counter) mode. The implementation details can be found in the Encrypt class' expandKey method. The obvious downside is that only up to 16 bytes of the password (which may be as little as 5.3 characters in UTF-8 encoding) are taken into account. The other obvious downside is that the key is simply the password being encrypted with a version of itself in CTR mode which is not very cryptographically safe. The shortcomings of this approach were exacerbated in the first public version of the JPS format (1.9) which used the key as an IV for all encrypted blocks, weakening the security of the format.

This key expansion is not supported since JPS v.2.0.

Structure of an archive

An archive consists of exactly one Standard Header and one or more Entity Blocks . Each Entity Block consists of exactly one Entity Description Block and at most one File Data Block. Each FIle Data Block consist of one or several Data Chunk Blocks. All values are stored in little-endian byte order, unless otherwise specified.

All textual data, e.g. file names and symlink targets, must be written as little-endian UTF-8, non null terminated strings, for the widest compatibility possible.

Standard Header

The function of the Standard Header is to allow identification of the archive format and supply the client with general information regarding the archive at hand. It is a binary block appearing at the beginning of the archive file and there alone. It consists of the following data (in order of appearance):

Signature, 3 bytes	The bytes 0x4A 0x50 0x54 (uppercase ASCII string “JPS”) used for identification purposes.
Major version, 1 byte	Unsigned integer represented as single byte, holding the archive format major version, e.g. 0x02 for version 2.0.
Minor version, 1 byte	Unsigned integer represented as single byte, holding the archive format minor version, e.g. 0x00 for version 2.0.
Spanned archive, 1 byte	When set to 1, the archive spans multiple files
Extra header length, 2 bytes	The total length of extra headers. In version 2.0 of the format it is always 76.

The total size of this header is 8 bytes, plus the size of the extra headers (if any).

Key Expansion Extra Header

The function of the Key Expansion Extra Header is to let you know of the PBKDF2 key expansion algorithm's configuration parameters used throughout this backup archive. It consists of the following data:

Identification Header, 4 bytes	The bytes 0x4A 0x48 0x00 0x01 used for identification purposes
Extra Header Size, 2 bytes	Unsigned short integer, little endian, holding the total size of this extra header (including the 4 bytes of the identification header), i.e 76 for a version 2.0 header
Algorithm, 1 byte	Unsigned byte holding the ID of the hash algorithm used for PBKDF2. The valid algorithms are: • 0 = SHA-1 • 1 = SHA-256 • 2 = SHA-512 Values up to and including 127 are reserved for future use.
Iterations, 4 bytes	Unsigned long integer, little endian, with the number of iterations to use in PBKDF2
Use Static Salt, 1 byte	Unsigned byte. When it is 1 use the Static Salt below with PBKDF2 unless otherwise specified in the encryption block. This allows you to cache the expanded key for encryption / decryption purposes. This is only recommended if you are using SHA-256 or SHA-512 with a high number of iterations. If this is 0 we recommend setting the Static Salt to all null bytes.
Static Salt, 64 bytes	The rest of the extra header (64 bytes in v.2.0) is the Static Salt mentioned above.

Entity Block

An Entity Block is merely the aggregation of exactly one Entity Description Block, followed by the encrypted contents of exactly one Entity Description Block Data and zero or one instances of a File Data Block. An Entity can be at present a File, Symbolic Link or Directory. If the entity is a File of zero length or if it is a Directory the File Data Block is omitted. In any other case, the File Data Block must exist.

Entity Description Block Header

The function of the Entity Description Block Header is to allow a client to read the encrypted Entity Description Block Data. It is a binary block consisting of the following data (in order of appearance):

Signature, 3 bytes	The bytes 0x4A, 0x50, 0x46 (uppercase ASCII string “JPF”) used for identification purposes.

Encrypted size, 2 bytes	The encrypted size of the following Entity Description Block Data
Decrypted size, 2 bytes	The decrypted size of the following Entity Description Block Data

Entity Description Block Data

it purpose is to provide the client information about an Entity included in the archive. The client can then use this information in order to reconstruct a copy of the Entity on the client's file system. The data is written to the archive encrypted with Rijndael-128 in CBC mode. The Entity Description Block Data consists of the following information before it is encrypted:

Length of entity path, 2 bytes.	Unsigned short integer, represented as 2 bytes, holding the size of the entity path data below.
Entity path data, variable length.	Holds the complete (relative) path of the Entity as a UTF16 encoded string, without trailing null. The path separator must be a forward slash (“/”), even on systems which use a different path separator, e.g. Windows.
Entity type, 1 byte.	• 0x00 for directories (instructs the client to recursively create the directory specified in Entity path data). When the entity type is 0x00 the Compression Type MUST be 0x00 as well. • 0x01 for files (instructs the client to reconstruct the file specified in Entity path data) • 0x02 for symbolic links (instructs the client to create a symbolic link whose target is stored, uncompressed, as the entity's File Data Block). When the type is 0x00 the Compression Type MUST be 0x00 as well.
Compression type, 1 byte.	• 0x00 for no compression; the data contained in File Data Block should be written as-is to the file. Also used for directories, symbolic links and zero-sized files. • 0x01 for deflate (Gzip) compression; the data contained in File Data Block must be deflated using Gzip before written to the file. • 0x02 for Bzip2 compression; the data contained in File Data Block must be uncompressed using BZip2 before written to the file. This is generally discouraged, as both the archiving and unarchiving scripts must be ran in a PHP environment which supports the bzip2 library.
Uncompressed size, 4 bytes	An unsigned long integer representing the size of the resulting file in bytes. For directories, symlinks and zero-sized files it is zero (0x00000000).
Entity permissions, 4 bytes	UNIX-style permissions of the stored entity.
File Modification Time, 4 bytes	The UNIX timestamp of the file's last modification time. For directories and symlinks it must be ignored and set to 0x00000000.

File Data Block

The File Data Block is only present if the Entity is a file with a non-zero file size. It consists of one or more Data Chunk Blocks. Do note that the File Data Block has no header. The collection of one or several Data Chunk Blocks is called the "File Data Block".

Data Chunk Block

Each Data Chunk Block consists of the following information:

Encrypted size, 4 bytes	Unsigned long containing the size, in bytes, of the encrypted data.
Decrypted size, 4 bytes	Unsigned long containing the size, in bytes, of the decrypted data. If the decryption yields more bytes, the extraneous bytes must be trimmed off.
Encrypted data, variable length	The decrypted data is compressed, depending on the Compression Type, and then encrypted using AES-128 in CBC mode. The compression format used may be: • Binary dump of file contents or textual representation of the symlink's target, for CT=0x00 • Gzip compression output, without a trailing Adler32 checksum, for CT=0x01 • Bzip2 compression output, for CT=0x02

In split archives, the first 8 bytes must appear within the same part. They may or may not be in the same part as the Entity Description Block Data. The Encrypted Data can span multiple parts. Since the minimum part size is 64Kb and the maximum Decrypted Size can't be over 64Kb, the Encrypted Data will either be in the same part in its entirety, or span exactly two parts.

Encrypted data block format

The encrypted blocks have one of the following possible formats. You can detect the data format in two ways.

First, the legacy format is only used with JPS version 1.9 and below. If the file header claims that the archive is JPS 1.10 then the current format MUST be used.

If you do not or cannot trust the file header you can do a simple heuristics. Read the last 24 bytes of the encrypted block. If the first four bytes are JPIV you definitely have a current format block. Otherwise you most likely have a legacy format block (there's 1 in 4,228,250,625 chance of false detection).

JPS 2.0 (Current)

In this format the IV for each encryption block is always different, produced by a crypto safe PRNG (either OpenSSL or mcrypt). Therefore the encryption *is* safe against cryptanalysis (as far as an attack against Rijndael-128 itself is not discovered). Moreover, it allows for the inclusion of a per-block salt for PBKDF2 key expansion.

Encrypted data, variable length	This data is encrypted with Rijndael-128 using the IV described below.
Per-Block Salt, 68 bytes (OPTIONAL)	The literal string JPST followed by the 64 bytes of the per-block salt. Discard the JPST marker and use the rest as the salt for the PBKDF2 algorithm. This section MUST be present when the Use Static Salt flag in the archive header is 0. This section MAY be present when the Use Static Salt flag in the archive header is 1. This means that you shouldn't simply skip checking the existence of this section just because Use Static Salt is 1. If it's present, use it and derive a new, per-block encryption key.
Initialization Vector (IV) data block, 20 bytes	The literal string JPIV followed by the 16 bytes (128-bit) Initialization Vector data. Discard the JPIV marker and use the rest of the block as the IV for your Rijndael-128 decryption engine.
Decrypted data length, 4 bytes	The size of the decrypted data in bytes. Since Rijndael-128 in CBC mode encrypts data in 16-byte (128-bit) blocks it needs to pad data with length not exactly divisible by 16 with zero bytes. These zero bytes are not part of the input data and need to be discarded. For example, if your decrypted data length is 24 and the Rijndael-128 decryption result is 32 bytes you need to throw away the last 8 bytes which are just the zero (null) padding bytes.

JPS 1.10 (Previous)

Note

Only compatible with JPS 1.10 archive files. Not compatible with JPS 1.9 archive files. Obsolete since JPS 2.0.

In this format the IV for each encryption block is always different, produced by a crypto safe PRNG (either OpenSSL or mcrypt). Therefore the encryption *is* safe against cryptanalysis (as far as an attack against Rijndael-128 itself is not discovered).

Encrypted data, variable length	This data is encrypted with Rijndael-128 using the IV described below.
Initialization Vector (IV) data block, 20 bytes	The literal string JPIV followed by the 16 bytes (128-bit) Initialization Vector data. Discard the JPIV marker and use the rest of the block as the IV for your Rijndael-128 decryption engine.
Decrypted data length, 4 bytes	The size of the decrypted data in bytes. Since Rijndael-128 in CBC mode encrypts data in 16-byte (128-bit) blocks it needs to pad data with length not exactly divisible by 16 with zero bytes. These zero bytes are not part of the input data and need to be discarded. For example, if your decrypted data length is 24 and the Rijndael-128 decryption result is 32 bytes you need to throw away the last 8 bytes which are just the zero (null) padding bytes.

JPS 1.9 and below (Legacy)

Note

Only compatible with JPS 1.9 and 1.10 archive files. Obsolete since JPS 2.0.

In this format the IV is always the same and derived from the encryption key. For this reason the encryption is NOT safe against some methods of cryptanalysis which could compromise the encryption key.

Encrypted data, variable length	This data is encrypted with Rijndael-128.
Decrypted data length, 4 bytes	The size of the decrypted data in bytes. Since Rijndael-128 in CBC mode encrypts data in 16-byte (128-bit) blocks it needs to pad data with length not exactly divisible by 16 with zero bytes. These zero bytes are not part of the input data and need to be discarded. For example, if your decrypted data length is 24 and the Rijndael-128 decryption result is 32 bytes you need to throw away the last 8 bytes which are just the zero (null) padding bytes.

End-of-archive header

This header is written after the end of the archive data, at the end of the last part of the archive.

When creating spanned archives, the first file (part) of the archive set has an extension of .j01, the next part has an extension of .j02 and so on. The last file of the archive set has the extension .jps. You must also ensure that the Entity Description Block is within the limits of a single part, i.e. the contents of the Entity Description Block must not cross part boundaries. The File Data Block data can cross one or multiple part blocks, but the header of each Data Chunk Block must both be inside the same part.

This header is written after the end of the archive data, at the end of the last part of the archive. Its structure is:

Signature, 3 bytes	The bytes 0x4A, 0x50, 0x45 ("JPE")
Number of parts, 2 bytes	The total number of parts this archive consists of. Non-spanned archives should set this to 1.

File count, 4 bytes	Unsigned long integer represented as four bytes, holding the number of files present in the archive.
Uncompressed size, 4 bytes	Unsigned long integer represented as four bytes, holding the total size of the archive's files when uncompressed.
Compressed size, 4 bytes	Unsigned long integer represented as four bytes, holding the total size of the archive's files in their stored (compressed) form

The size of the EOA header is 17 bytes for version 1.9 of the format.

Change Log

Revision History		
	July 2010	NKD, Akeeba Ltdhttp://www.akeebabackup.com
Described version 1.9		
Revision History		
	January 2017	NKD, Akeeba Ltdhttps://www.akeebabackup.com
Described version 2.0		

Appendix C. GNU Free Documentation License

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety

of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to

ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections

as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket

the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/ [http://www.gnu.org/copyleft/] .

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (C) YEAR YOUR NAME.
>
> Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.