

Akeeba Subscriptions User's Guide

Nicholas K. Dionysopoulos

Akeeba Subscriptions User's Guide

Nicholas K. Dionysopoulos

Publication date April 2011

Abstract

This book covers the use of the Akeeba Subscriptions component and its bundled modules and plugins for selling and managing subscriptions on your Joomla!™-powered web sites.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 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 can be found on-line at <http://www.gnu.org/licenses/fdl.html>.

Table of Contents

| | |
|--|----|
| 1. Introduction and installation | 1 |
| 1. Introducing Akeeba Subscriptions | 1 |
| 2. Requirements and compatibility | 2 |
| 3. Installation | 2 |
| 3.1. Installation | 2 |
| 3.2. Installation troubleshooting | 2 |
| 3.3. Updating to the latest release | 5 |
| 4. Uninstallation | 6 |
| 2. Initial set-up and usage | 7 |
| 1. How subscriptions work | 7 |
| 2. Configuration options | 8 |
| 3. Subscription Levels | 9 |
| 4. Tax Rules | 12 |
| 5. Upgrade Rules | 13 |
| 6. Coupons | 14 |
| 7. Subscriptions management | 15 |
| 8. Affiliates management | 16 |
| 9. Front-end items | 16 |
| 10. Importing from other components | 17 |
| 3. Payment methods, integrations and plugins | 18 |
| 1. Payment methods | 18 |
| 1.1. Paypal | 18 |
| 1.2. None | 19 |
| 1.3. WorldPay | 19 |
| 1.4. Off-line | 20 |
| 1.5. 2Checkout Standard Purchase Routine | 20 |
| 1.6. ccAvenue | 21 |
| 1.7. eWay | 21 |
| 2. Integration with third party software | 22 |
| 2.1. Community Builder integration | 22 |
| 2.2. ccInvoices integration | 23 |
| 2.3. Community ACL integration | 23 |
| 2.4. DOCman Integration | 24 |
| 2.5. JCE Integration | 25 |
| 2.6. JomSocial integration | 26 |
| 2.7. Joomla! 1.6 User Groups Integration | 27 |
| 2.8. JUGA Integration | 28 |
| 2.9. JoomlaXI JomSocial User Profile Types integration | 29 |
| 2.10. K2 Integration | 30 |
| 2.11. NinjaBoard Integration | 31 |
| 2.12. Tienda Integration | 32 |
| 2.13. Delete users on subscription expiration | 33 |
| 2.14. VirtueMart Integration | 33 |
| 2.15. VirtueMart 2 Integration | 34 |
| 2.16. RedShop Integration | 35 |
| 2.17. Sample Fields | 36 |
| 2.18. Automatic Country and City fill | 36 |
| 2.19. Custom SQL scripts | 36 |
| 3. Other plugins | 36 |
| 3.1. Subscription expiration control | 37 |
| 3.2. Subscription emails | 37 |

| | |
|--|----|
| 3.3. Subscription expiration notification | 37 |
| 3.4. Content restriction | 38 |
| 3.5. The Akeeba Subscriptions Link (aslink) plugin | 39 |
| 4. Akeeba Subscriptions' modules | 41 |
| 1. List of active subscriptions | 41 |
| 2. List subscription levels | 41 |
| 5. Developers' information | 42 |
| 1. The "akeebasubs" plugin events | 42 |
| 1.1. onAKSubscriptionChange | 42 |
| 1.2. onAKUserRefresh | 42 |
| 1.3. onSubscriptionFormRender | 43 |
| 1.4. onValidate | 44 |
| 1.5. onAKUserGetData | 44 |
| 1.6. onAKUserSaveData | 45 |
| 2. The "akpayment" plugin events | 45 |
| 2.1. onAKPaymentGetIdentity | 45 |
| 2.2. onAKPaymentNew | 45 |
| 2.3. onAKPaymentCallback | 45 |

Chapter 1. Introduction and installation

1. Introducing Akeeba Subscriptions

At a glance

Akeeba Subscriptions is a subscriptions management component for Joomla!™ 1.5/1.6/1.7 and compatible distributions. It is built using our renowned Framework on Framework architecture which extends the standard Joomla! API, ensuring greater stability and compatibility across different Joomla! releases. It is licensed under the GNU General Public License (GPL) version 3 [<http://www.gnu.org/licenses/gpl.html>] or –at your option– any later version published by the Free Software Foundation. Its licensing scheme means that you are free (and, in fact, more than welcome) to install it on as many sites as you want, whenever you want and use it for as long as you want, no strings attached. There are no secret per-domain licensing fees and you can use it to sell one or several millions of subscriptions without any hidden costs. We love Freedom of choice as much as you do!

The killer features

Its feature list is nothing sort of amazing, either. Out of the box, Akeeba Subscriptions supports these features:

- Streamlined administrator interface which can even present you an interactive sales graph and sales report as soon as you launch the component
- Rich subscription levels (subscription packages) editor which allows you to choose different images for each of your subscriptions and even a different order confirmation and order cancellation text to show to your users.
- The easiest subscription management interface you've seen on a component. It will even show you your users' faces, powered by Gravatar.
- Users can upgrade or expand (renew) their subscriptions. Renewing a subscription will create a new subscription which becomes valid the very second their old one expires. Users do not lose any of their subscription time when renewing, unlike most other subscription systems out there.
- Full support for delayed payments, e.g. when using e-checks with PayPal.
- Discount coupon codes which allow you to set an absolute money value or percentage discount for all or a specific subscription level and user, have publish up/publish down dates or a usage limit (e.g. the coupon code is valid only for the first 100 people to use it) – or a combination of any and all of the above!
- Automatic discounts for upgrading or renewing subscriptions based on the subscription level and days of presence in the subscription package. This allows you to easily create rules like: 30% discount if you renew up to 30 days before the end of your subscription, 15% discount if you renew within the last 30 days, no discount otherwise.
- Full support for complex tax calculations based on country, state and ZIP code. It fully integrates with the European Union's VIES system so that you can charge no VAT tax for intra-EU B2B transactions.
- The subscription form can work with or without Javascript. With Javascript it becomes a fully fledged, auto-validating subscription form. Without Javascript it works as a standard web form, accessible to users who do not wish / can not use Javascript on their browser.
- Integration with Joomla! 1.6/1.7 and later user group mapping
- Integration with third party components: JUGA, K2, DOCman, JCE, NinjaBoard, VirtueMart, Tiendam JomSocial, Community Builder, ccInvoices and much more! Check out our documentation for more information.

- Content restriction: a content plugin to show parts of your content only to registered subscribers, without the need of any external tool.
- Payment methods: PayPal (for personal, verified and business accounts) is supported out of the box. Other payment methods (ccAvenue, WorldPay, 2Checkout and more) are being added continuously. Check out our documentation for more information.
- Send emails to subscribers upon subscription, when their subscription/payment status changes and when their subscription is about to expire

Support policy

Please note that the software is provided free of charge, but support is not. You need to have an AKEEBADELUXE, SUPPORT or FORUMSUPPORT subscription on AkeebaBackup.com to seek support regarding setting up, using and customizing Akeeba Subscriptions.

2. Requirements and compatibility

Akeeba Subscriptions 1.0.b4 and later will attempt to detect if your server meets the minimum requirements. If it does not, it will refuse to install. In any case, Akeeba Subscriptions requires the following server-side configuration:

- PHP 5.2.7 or later. Running it on earlier versions will either be problematic or it won't work at all.
- MySQL 5.0.41 or later. Earlier database server versions will not be supported. Do note that earlier releases of MySQL are obsolete and not supported any more by Oracle (the company who controls the development of MySQL).
- Joomla! 1.5.14 or later (fully supported) or Joomla! 1.6.0 or later.

Since version 2.0.a1, Akeeba Subscriptions no longer uses the Nooku Framework which extends its compatibility with more server environments and third-party software.

Important

If you installed Akeeba Subscriptions 1.0.b4 or earlier on your site and your site displays a blank page, please remove the `index.html` files from your site's root and your site's administrator directories.

3. Installation

3.1. Installation

Installing the package is the same as with any other Joomla! component. Go to your site's back-end, Extensions, Install/Uninstall (Joomla! 1.5) or Extensions, Manage (Joomla! 1.6) and click on Browse. Locate the ZIP package and click on Upload and Install. If the installation fails, please refer to the installation troubleshooting section of this guide.

3.2. Installation troubleshooting

Joomla! 1.6/1.7 is logging you out during installation

The first thing you might observe is that **Joomla! 1.6 is logging you out when trying to install** Akeeba Subscriptions. This is due to a bug in Joomla! 1.6. Please follow these instructions [http://docs.joomla.org/Why_does_the_administrator_logoff_all_of_the_sudden] to fix the database table that causes this issue. After doing that you will have to log in again to your site. You will see that everything is missing from the back-end interface.

Don't panic! That's part of the Joomla! bug. Just log out using the link on the top-right of your page, then log back in. Everything is back to normal and you can retry the installation.

Joomla! 1.6/1.7 says "can't build admin menu" and fails

Note

This problem should have gone away since Akeeba Subscriptions 1.0.0 (Stable) and later releases

On some other occasions, especially when trying to install or update the component after an installation error, Joomla! will complain that it **can not build admin menus**. This is due to another Joomla! 1.6 bug we have sent a patch for [http://joomla.org/gf/project/joomla/tracker/?action=TrackerItemEdit&tracker_item_id=25663] to the Joomla! development team. Meanwhile, you will have to run a few SQL statements against your database with a database editor such as phpMyAdmin:

```
DELETE FROM jos_assets WHERE `name` = 'com_akeebasubs';
DELETE FROM jos_menu WHERE `menutype` = 'main' AND `alias` = 'com_akeebasubs';
DELETE FROM jos_extensions WHERE `type` = 'component' AND `element` = 'com_akeebasubs';
```

Remember to change *jos_* with your database tables' name prefix if you changed it during installation! It's easy to find out what it is. Take a look at the database and you'll see that all of your tables begin with the same few letters. That's your prefix. After running those SQL statements you can proceed with the reinstallation of the component.

Version check

Check your Joomla! version. Joomla! 1.5.0 through 1.5.10 can not install Akeeba Subscriptions. This is a limitation of these extremely outdated releases. If you have such a site, upgrade it. Joomla! 1.5.11 through 1.5.14 may or may not be able to install Akeeba Subscriptions. Again, these are limitations of those outdated Joomla! releases on certain hosts. The only workaround is to upgrade your Joomla! installation.

If you have Joomla! 1.6 make sure that you are using Joomla! 1.6.0 (the "stable" release, not the RC or any of the betas). Mind you, we have only tested the installation with Joomla! 1.6.2 and later releases. Akeeba Subscriptions works perfectly on Joomla! 1.7 sites.

Checking your temporary directory

First, we will have to make sure that you are using a valid temporary directory. Many sites are configured to use the system-wide (/tmp) directory or an invalid directory, causing installation problems. In order to change your site's temporary directory setting you have to follow this procedure, depending on your Joomla! version.

Joomla! 1.5

1. Go to your site's administrator back-end and click Help, System Info from the top menu.
2. Click on the Directory permissions link
3. Scroll down the page and find the first Cache Directory line. It is the fourth from the bottom of the page.
4. Next to the Cache directory label you can see a path, e.g. /home/myuser/public_html/cache
5. Replace the cache word with tmp in that path, i.e. /home/myuser/public_html/tmp and note down this path. This is your new temp path.
6. Go to Site, Global Configuration menu item from the top menu.
7. Click on the Server link

8. Find the "Path to Temp-folder" and replace its contents with the new temp path from step #5.
9. Save your Global Configuration

Joomla! 1.6/1.7

1. Create a file named mynewtmp.php with the following contents:

```
<?php echo dirname(__FILE__).'/tmp';
```

2. Upload the file to your site and access it as <http://www.example.com/mynewtmp.php> where www.example.com is the domain name to your site.
3. Write down what you see on the screen. That's your new temporary directory path.
4. Remove the mynewtmp.php file from your site.
5. Go to your site's administrator back-end and click on Site, Global Configuration menu item from the top menu.
6. Click on the Server link
7. Find the "Path to Temp-folder" and replace its contents with the new temp path from step #3.
8. Save your Global Configuration

Enable FTP

On most shared hosts which do not run suPHP (if you didn't understand anything, most likely the following *is applicable to you too*) you have to enable Joomla!'s FTP layer. Otherwise Joomla! won't be able to write the files to its directories and installation will fail.

1. Go to Site, Global Configuration menu item from the top menu.
2. Click on the Server tab
3. Set Enable FTP to Yes
4. In the FTP Host try using `127.0.0.1` or `localhost` or the FTP hostname assigned by your host
5. In the FTP Username and FTP Password fields provide the FTP username and password assigned by your host
6. In the FTP Root you have to type in the FTP path to your site's root. Here is the easy way to find it using FileZilla [<http://filezilla-project.org/download.php>]:

Connect to your site using FileZilla. Navigate inside the folder Joomla! is installed in. Usually it's a directory named `public_html`, `htdocs`, `www` or something similar. If unsure don't ask us, ask your host. Now, on the right-hand pane you will find the FTP path. Most likely it will look something like `/public_html`. Copy this and paste it into the FTP Root text box in your Joomla!'s Global Configuration page.

7. Save your Global Configuration. If you got everything correctly, you should see a message that your configuration was saved. If you see an error message please seek assistance on the Joomla! Forum [<http://forum.joomla.org>].

Manual installation

Sometimes Joomla!™ is unable to properly extract ZIP archives due to technical limitations on your server. In this case, you can follow a manual installation procedure.

First, you have to extract the installation ZIP file in a subdirectory named `akeeba` on your local PC. Then, upload the entire subdirectory inside your site's temporary directory. At this point, there should be a subdirectory named `akeeba` inside your site's temporary directory which contains all of the ZIP package's files.

If you are unsure where your site's temporary directory is located, you can look it up by going to the Global Configuration, click on the Server tab and take a look at the Path to Temp-folder setting. The default setting is the `tmp` directory under your site's root. Rarely, especially on automated installations using Fantastico, this might have been assigned the system-wide `/tmp` directory. In this case, please consult your host for instructions on how to upload files inside this directory, or read the instructions above about changing your Joomla!™ temporary directory back to the default location and making it writable.

Assuming that you are past this uploading step, click on the Extensions, Install/Uninstall (Joomla! 1.5) or Extensions, Manage (Joomla! 1.6 users) link on the top menu. In this page, locate the Install Directory edit box in the Install from Directory area. It is already filled in with the absolute path to your temporary directory, for example `/var/www/joomla/tmp`. Please append `/akeeba` to it. As per our example, it should look something like `/var/www/joomla/tmp/akeeba`. Then, click on the Install button.

Still problems?

If you still can't install Akeeba Subscriptions and you are receiving messages regarding unwritable directories, inability to move files or other similar file system related error messages, please do not ask us for support. These errors stem from your site set up and can best be resolved by asking for help in the official Joomla!™ forums [<http://forum.joomla.org>]. We can only support software we develop ourselves. Joomla!'s extension installer is certainly not developed by us and –believe us– we have tried to improve it and submitted some still pending patches to the Joomla! project. Therefore we regret that we have to say that, but we can't help you. Thank you for your understanding.

3.3. Updating to the latest release

Akeeba Subscriptions can be updated with three different methods: installing the new version on top of the old one, using the integrated Live Update system or through Joomla! 1.6 and later's extensions update feature.

Updating directly

This is the failsafe approach, but the least convenient. Download the latest Akeeba Subscriptions release from <https://www.AkeebaBackup.com/latest> and save the ZIP file to your hard disk. Log in to your site's backend, click on Extensions, then Install/Uninstall (Joomla! 1.5) or Extensions Manager (Joomla! 1.6 and later). Use the Browse... button to locate the ZIP file you downloaded, then click on Upload and Install. All Joomla! versions since 1.5.5 are smart enough to understand that you're doing an upgrade instead of installation and adjust the process accordingly.

Important

Do NOT uninstall Akeeba Subscriptions before updating it! Uninstalling will remove all of your data, including all subscriber information!

Using Live Update

Since Akeeba Subscriptions 1.0.0 we have integrated our Live Update system inside Akeeba Subscriptions. Log in to your site's backend and go to Components, Akeeba Subscriptions. Look towards the bottom of the page. There should be an icon which reads "Update found" when there is a new version available. Click on it and then click on "Update now". The new version will be downloaded and installed automatically for you. In case this doesn't work, or if "Live Update not supported" is displayed below the icon, please make sure that your host's firewall allows TCP/IP communications over port 80 and 443 to `akeebabackup.com` and `s3.amazonaws.com`. If your host requests IP addresses instead of domain names, please ask them to trace them from the server as they are multicast hostnames, which means that they resolve to a different IP depending on where in the world you are.

Using Joomla! 1.6 and later's extension update

Since Joomla! 1.6, the Joomla! Extensions Manager allows directly updating your extensions. Just log in to the backend of your site and go to Extensions, Extension Manager. Click on the Update link below the toolbar. Then click on the Find Updates button. If there is a new Akeeba Subscriptions release it will appear in the list below. Tick the box on the left of the row and then click on the Update button. If your site is compatible with this Joomla! feature, you will see the new version being installed automatically for you.

Something not working right after the update?

Sometimes Joomla! "forgets" to copy all updated files. This is something that we have seen many times on Joomla! 1.5 sites. In this case, simply follow the instructions in the Updating Directly section above. This will force Joomla! to retry updating the component, copying the missing files and everything will work again.

4. Uninstallation

You can uninstall the component just like any other Joomla! component. In your site's back-end, just go to Extensions, Install/Uninstall (Joomla! 1.5) or Extensions Manager (Joomla! 1.6 or later), click on Uninstall, select Akeeba Subscriptions and click on Uninstall. This will completely remove Akeeba Subscriptions including all subscriber information.

Chapter 2. Initial set-up and usage

1. How subscriptions work

Before you begin setting up Akeeba Subscriptions, you should be aware of the way subscriptions work under the hood. The most basic concept is the subscription level (referred to simply as Level from now on). This is what you are selling to your customers. Your customers buy a subscription to a Level. The composite piece of information containing the Level, payment information and expiration time is called a Subscription.

When a user enters the subscription page, Akeeba Subscriptions first fetches the price associated with the Level your user wants to buy. It will then try to understand if there is an applicable discount. First, it will look into Upgrade definitions and produce a list of upgrade rules applicable for the user and the Level he wishes to buy. If there are multiple applicable upgrade rules, the one giving the maximum discount is elected. Next up, it takes a look at the coupon code, is supplied. If it is an active coupon and your user can use it to get a discount, the coupon's discount is calculated. In the case where there are two discounts, both an upgrade discount and a coupon discount, the highest one is elected. The level price minus the discount is called the Net Price.

Right up, Akeeba Subscriptions will try to determine the applicable tax by going through all the tax rules. It will go through the tax rules and keep the one which is the closest match to the country, zip code, city and VIES registration status. If no match is found, Akeeba Subscriptions will select the tax rate defined in the very first tax rule you have defined. This is called the Tax Amount. The sum of the Net Price and Tax Amount is the Gross Amount and that's what the user will have to pay.

When the user hits the Subscribe button, Akeeba Subscriptions will first check if this is a new or existing user. If it is a new user, it will create a new –but inactive– user account based on the information the user has supplied. A new, inactive subscription will be created for your user and he's redirected to the payment gateway. Exception: if you have an 100% or more discount (which means that the user shall pay nothing) he's not redirected to the payment gateway. Instead, he's redirected to the Subscription Confirmation page and his subscription becomes instantly active.

Once the user finishes the payment, he's redirected to the Subscription Confirmation page. If he changes his mind and cancels the payment, he will be takes to the Subscription Cancellation page. The text on each page is determined by the relevant fields in the Level configuration.

At the same time, your payment processor will send a post-back to your site, notifying Akeeba Subscriptions about the payment status. If the payment is marked as complete, the subscription is activated. Its start and expiration date are calculated based on the exact date and time that the payment post-back was received and the Level's configured length. If the associated user account was marked as blocked (inactive), it is activated. If any integrations are set up, they will run. For example, you may be adding users to JUGA groups upon a successful subscription.

Based on the above, you need to configure the following –in the order shown– for Akeeba Subscriptions to work:

- Configuration Options - well, that's just the currency you're selling in!
- Subscription Levels - the subscription products you are selling, i.e. your inventory
- Tax Rules - determine if the user has to be charged taxes over the subscription amount and how much that will be
- Upgrade rules - (optional) discounts based on what other products the user has bought and how long he's been a subscriber to them
- Coupons - (optional) discount codes used for promotions

Alternatively, you can import Subscription Levels and Coupons from other subscription systems instead of manually defining them.

The sections below will tell you how to go through each of these steps.

2. Configuration options

Click on Parameters (Joomla! 1.5) or Options (Joomla! 1.6, 1.7 and later) in the toolbar links area. The available options are:

Currency Options

Currency code The three letter ISO 4217 currency code [<http://www.xe.com/iso4217.php>]. Common values are EUR (Euro), USD (United States Dollar), GBP (Great Britain Pound), CAD (Canada Dollar), AUD (Australia Dollar), JPY (Japan Yen), CNY (China Yuan) and MXN (Mexico, Pesos). This is used on virtually all of the payment processors to notify them about the currency you're selling your goods in. Please remember that all of your payment processors must support the selected currency. Some payment processors only accept a subset of currencies, most usually only one of USD, GBP and EUR.

Default: EUR (Euro)

Currency Symbol The currency symbol to be shown to your web visitors, e.g. € for Euro, £ for British Pound, \$ for US/Australia/Canada Dollar or ¥ for Japan Yen. This is free form text used to prefix the prices. If your currency has no special symbol you can use text to describe it, or leave blank to completely omit the currency symbol display throughout the components.

Default: € (Euro symbol)

Back-end display

Show Gravatar images in Subscriptions page When enabled, Gravatar images are shown next to each user in the Subscriptions page. If disabled, there will be no avatar images displayed.

Default: Yes

Images directory The directory, relative to your site's root, where the subscription level images will be stored.

Default: images (Joomla! 1.6+) or images/stories (Joomla! 1.5)

Front-end display

Show steps bar Should we render the steps bar (1-2-3) on the top of the page?

Default: Yes

Allow collection of personal information and business registrations When enabled, Akeeba Subscriptions will ask for address information and business registration information. This information is required if you have to issue invoices or receipts to your customers. On some jurisdictions, the automatic receipt generated by, for example, PayPal is adequate. In this case, there is no point collection this information on your site. In this case, set this option to No and Akeeba Subscriptions will not ask your customers for their address, city, state, ZIP, country, business registration, business name, occupation or VAT number. This also means that AS will no longer allow different tax rates based on city, state, country or VIES registration status; the default (first) tax rate will be applied instead.

Default: Yes

Show login box When enabled, if the user is not logged in yet when he tries to subscribe, a login box will be displayed on top of the subscriptions page, allowing her to login to the site. If disabled, this login box

will not be disabled. Many people have reported that the login feature is confusing and redundant, as there is usually a login button or module for the same function, hence this option.

Permissions (Joomla! 1.6 or later only)

This is the standard Joomla! 1.6 or later ACL permissions widget. The meaning of the different ACL settings is as follows:

| | |
|------------------|---|
| Configure | Allows the user to access the Options page of the component (the very page you are on right now) |
| Access Component | Allows backend access to the component. Frontend access IS NOT affected by this setting! |
| Create | Allows the creation of new records on any backend page (e.g. Subscription Levels, Subscriptions, Users, Tax Rules, Coupons and so on) |
| Delete | Allows the deletion of records on any backend page |
| Edit | Allows the user to edit any record on any backend page |
| Edit State | Allows the user to publish/unpublish records on any backend page |

3. Subscription Levels

You can access this feature by clicking on Subscription Levels in the toolbar links area.

Creating or editing a Subscription Level, you have these options:

| | |
|---------------------------------------|---|
| Title | How the subscription will be presented to your users. We strongly suggest using all-capital letters and no spaces for easier administration, albeit you can really use anything you like. |
| Image | Select a picture to represent your subscription. The image must be located in your site's images/stories (Joomla! 1.5) or images (Joomla! 1.6) directory. |
| Subscription Length (days) | How many days a subscription in this level lasts. Common values are 30 (one month), 180 (half a year) and 365 (one year). |
| Price | The price of a subscription in this level. If you need to enter a decimal value, use a dot as the decimal separator. For example, 12.30 is valid, whereas 12,30 is INVALID and will be interpreted as 1230. Do NOT use a thousands separator. For example 1000 is valid, whereas 1,000 is INVALID. |
| Slug | The alias, used to construct the URL. Use only lowercase Latin characters without diacritics and accents, dashes and underscores. For example uber-sub is valid, whereas über SUB is INVALID. |
| Published | Should it be shown to your users? |
| First expiration notification (days) | Akeeba Subscriptions will send a notification email to users whose subscription is expiring soon. It can send up to two emails. This parameter tells Akeeba Subscriptions how many days before the subscriptions expiration the first email will be sent. Set to 0 to not send out a notification email. |
| Second expiration notification (days) | Akeeba Subscriptions will send a notification email to users whose subscription is expiring soon. It can send up to two emails. This parameter tells Akeeba Subscriptions how many days before the subscriptions expiration the second email will be sent. Set to 0 to not send out a notification email. |
| Short Description | A description of the subscription to show to your users. Keep it short (ca. 30 words or less) or it will overflow the boundaries of the subscription presentation areas in the front-end. |

Since Akeeba Subscription 1.0.b4 you can use content plugin codes (like our own aslink plugin) in the description.

Message to show after successful sign-up The text in this area will be presented to the users after they successfully complete their payment. Use it to thank them for giving you their money and show them important information about their subscription, e.g. links to your support method, download links etc.

Warning

Do not include any sensitive information (such as "secret" download links) as the content of this page can be directly accessed by a knowledgeable user by manipulating the URL. If you want to include such information, please use the Restricted Content plugin bundled with Akeeba Subscriptions to make sure that the information will only be shown to users holding a valid subscription.

Since Akeeba Subscription 1.0.b4 you can use content plugin codes (like our own aslink plugin) in the message.

Message to show after sign-up cancellation The text in this area will be presented to the users after they cancel their payment (if your payment gateway supports this feature). This is your last attempt to changing your user's mind or have them give you feedback on the reasons they decided to not move forward with their subscription. Use it wisely and do not insult your users, please :)

Since Akeeba Subscription 1.0.b4 you can use content plugin codes (like our own aslink plugin) in the message.

Multi-lingual support

Akeeba Subscriptions 2.0 and later are able support multi-lingual sites without the need of a third-party extension. This is done by using language code "merge tags" in the three last fields outlined above: short description, message to show after successful sign-up and message to show after sign-up cancellation. The merge tags in the text look like this:

```
[IFLANG en-GB]This is the English description[/IFLANG]
[IFLANG fr-FR]Ceci est la description française[/IFLANG]
[IFLANG de-DE]Diese ist die deutsche Beschreibung[/IFLANG]
[IFLANG el-GR]##### # #####[/IFLANG]
```

Essentially, you have to wrap your per-language text in [IFLANG *languageCode*] and [/IFLANG] merge tags. The languageCode you must use is the five letter language code used by Joomla!'s translation packages. For example, the English language is en-GB (British English), en-AU (Australian English) or en-US (Unites States English), German is de-DE (Germany) or de-AT (Austria) and so forth. If you're unsure, please go to your site's back-end, Language Manager and take a look at the language code in there.

Please note that you can not use the same language code twice.

Customising the messages to show on successful sign-up and cancellation

Since Akeeba Subscriptions 2.0.a3, you can personalise the messages which are shown after a successful sign-up or cancellation of the subscription process. This is done using "merge codes". The merge codes are explained below. Please note that you can use them in conjunction with Joomla! content plugins as well. In fact, using the merge codes you are open to a world of possibilities, even integrating an external affiliate system –like OSI Affiliates– on your site.

The available merge codes are:

- [SUB:ID] The numeric, unique Subscription ID
- [SUB:USER_ID] The numeric Joomla! user ID of the subscriber
- [SUB:AKEEBASUBSCRIPTIONLEVELID] The numeric ID of the subscription level
- [SUB:PUBLISH_UP] The exact date and time the subscription will be activated in YYYY-MM-DD hh:mm:ss format, e.g. 2011-12-31 13:10:50.
- [SUB:PUBLISH_DOWN] The exact date and time the subscription will be deactivated in YYYY-MM-DD hh:mm:ss format, e.g. 2012-12-31 13:10:49.
- [SUB:ENABLED] This returns 1 if the subscription is enabled (e.g. the payment processor already notified us that the transaction is valid and it's not a renewal for a future date) or 0 if it's not enabled yet.
- [SUB:PROCESSOR] The name of the payment processor plugin, e.g. "paypal" for the PayPal payment plugin
- [SUB:PROCESSOR_TRANSACTION_KEY] Unique transaction ID assigned by the payment processor. IMPORTANT! This may NOT be available if the payment processor has not contacted your site with the result of the transaction before redirecting the user back to your site.
- [SUB:STATE] The payment state. C means completed, P is pending, X is cancelled, N means it hasn't been processed yet. IMPORTANT! This may NOT be available if the payment processor has not contacted your site with the result of the transaction before redirecting the user back to your site.
- [SUB:NET_AMOUNT] The amount the user paid, before taxes.
- [SUB:TAX_AMOUNT] The amount of taxes that the user paid.
- [SUB:GROSS_AMOUNT] The total amount the user paid, including taxes.
- [SUB:CREATED_ON] The exact date and time the user pressed the Subscribe Now button in YYYY-MM-DD hh:mm:ss format.
- [SUB:AKEEBASUBSCRIPTIONCOUPONID] The coupon used during the subscription, or 0 if no coupon was used
- [SUB:AKEEBASUBSCRIPTIONUPGRADEID] The upgrade rule automatically applied to the subscription, or 0 if no upgrade rule was used
- [SUB:AKEEBASUBSCRIPTIONAFFILIATEID] The affiliate who referred this subscription, or 0 if no affiliate was used
- [SUB:PREDISCOUNT_AMOUNT] The amount of the subscription, before any coupon or upgrade rule discount was applied
- [SUB:DISCOUNT_AMOUNT] The discount amount (coupon, upgrade rule) applied to the subscription
- [USER:ISBUSINESS] If the user chose to perform a business registration, 0 otherwise
- [USER:BUSINESSNAME] Business name
- [USER:OCCUPATION] Business occupation specified
- [USER:VATNUMBER] VAT registration number
- [USER:VIESTRIBUNAT] If VAT number is VIES-registered
- [USER:ADDRESS1] The address field (part 1)

[USER:ADDRESS2] The address field (part 2)

[USER:CITY] City

[USER:STATE] State (two letter code); only exists for Canada and USA

[USER:ZIP] ZIP/Postal Code

[USER:COUNTRY] Two-letter ISO code of the selected country, e.g. DE for Germany, FR for France, US for USA, CA for Canada and so on

[CUSTOM:*yourFieldName*] *yourFieldName* is the name of a custom field in all uppercase letters. Custom fields can be defined in plugins. If you have created any custom field plugins, you know what this is. If you don't know what this is, you most likely don't need it!

4. Tax Rules

You can access this feature by clicking on Tax Rules in the toolbar links area.

Out of the box, Akeeba Subscriptions is set up to not apply any VAT or other tax at all. However, this is not what most users would like. Below we are providing the most common setups based on your business type. Our lawyers told us that we have to say this to you: Please note that we are not lawyers or tax technicians. As a result, our advice is based on our personal experience, may be flawed and in no way should be considered a legal or financial advice. Should you chose to use it you do so at your own risk. We strongly advise you to ask a qualified tax technician, lawyer or other relevant professional qualified by law to give you advice regarding tax laws. That said, here's the most common tax setups:

Extra-EU businesses, e.g. US based

Tax Rules determine if the user should be charged a tax amount on top of the regular price of the subscription level. If you do not wish to charge any taxes (e.g. you have already included any applicable taxes in your subscription price), you will still have to create a default rule with 0 tax charges. In order to do so, click on the New button, select the "- Select -" option in the Country drop-down and set the tax rate to 0, then click on Save. Now you have created a catch-all rule with 0% tax. Likewise, if you want to charge all users a flat tax rate, say 18%, do the same thing, but set the tax rate to your desired percentage, in our case 18 (note: without the % sign!) and click on Save.

Intra-EU businesses with a VIES registered VAT number, or extra-EU businesses with an EU VAT number

If you are a business based in the European Union with a VIES-registered VAT number you have to create a total of 30 rules to conform to the overcomplicated EU tax regulations. First, create a default rule with 0% tax as noted above. Then do the same thing, but set the VIES Reg option to Yes and the tax rate to 0%. Now create new tax rules for each one of these countries, VIES Reg set to No and tax rate set to your local VAT tax rate (e.g. 23% for Greece): United Kingdom, Austria, Belgium, Bulgaria, Cyprus, Czech Republic, Germany, Greece, Denmark, Estonia, Spain, Finland, France, Hungary, Ireland, Italy, Lithuania, Luxembourg, Latvia, Malta, Netherlands, Poland, Portugal, Romania, Sweden, Slovenia, Slovakia. For the country of your business (e.g. Greece for a Greece-based business) create on more rule with VIES Reg set to Yes and the tax rate set to your local VAT tax rate (e.g. 23% for Greece). This way, the following happens:

- If the customer is extra-EU he will be charged no VAT
- If the customer is an intra-EU business, he will be charged no VAT (he's liable to VAT on his local tax authorities, not your country's tax authorities)

- If the customer is intra-EU, but not a business with a VIES-registered VAT number, he's paying VAT.
- If the customer is coming from your own country, you are charging him with VAT no matter what.

Do note that the same setup is required if you are an extra-EU business –e.g. US-based– with an EU VAT ID (it's in the format EU012345678). In this case, you have no EU country of residence, so you can skip the last tax rule mentioned above.

Intra-EU businesses without a VIES-registered VAT number

Please ask your lawyer whether it's legal for you to sell subscriptions to residents of a country other than yours. There is a gap in the EU directives regarding that case, therefore we can't provide you with even semi-accurate tax setup information within any stretch of imagination.

Tax Rule options

Each tax rule has the following options:

| | |
|----------|--|
| Country | Which country this tax rule is applicable in. If empty (the "-" Select "-" pseudo-option of the country list) it applies to all countries. |
| State | Which state this tax rule is applicable in. Note: only US and Canada states and provinces can be selected. If left empty (the N/A option) it applies to all states. |
| City | Which city this tax rule is applicable in. Do note that the spelling of the city must match, but it's case insensitive. This means that new york, NEW YORK and New York are equivalent. Leave blank to match all cities. |
| VIES Reg | If set to Yes, the tax rule matches only those clients with a valid, VIES registered EU VAT number. If you don't know what VIES registration is, read this [http://ec.europa.eu/taxation_customs/taxation/vat/traders/vat_number/index_en.htm]. You can verify if a VAT number of an EU resident or EU registered business is also VIES registered through EU's official web application on Europa.eu [http://ec.europa.eu/taxation_customs/vies/]. |
| Tax Rate | The tax rate percentage. If you need to enter decimal digits, use a dot not a comma. For example, 17.5 is valid, 17,5 is INVALID. Do NOT type the percentage sign. For example, 23 is valid, 23% is INVALID. The percentage is always implied, that's why it's printed after the field's contents. |
| Enabled | If set to Yes, the VAT Rule will be applied, otherwise it will be ignored (as if it weren't set at all) |

5. Upgrade Rules

You can access this feature by clicking on Upgrades in the toolbar links area.

This feature allows you to create automatically applied discount rules to reward your loyal customers by giving them a discount. If you are not sure this is what you want to do, I strongly suggest reading Seth Godin's "How should you treat your best customers?" [http://sethgodin.typepad.com/seths_blog/2011/02/how-should-you-treat-your-best-customers.html] and view this video of his [<http://www.attorneymarketing.com/2010/04/25/seth-godin-explains-how-to-build-a-tribe-of-loyal-clients/>] where he explains his ideas on small businesses even more clearly.

Each Upgrade rule has the following options:

| | |
|---------------|--|
| Title | Just a title to remind you of what this upgrade rule does |
| Published | If set to No, it won't be taken into account during a new subscription's price calculation |
| From level | The subscription level for which the user must already have a valid subscription in order for the upgrade rule to take effect |
| To Level | The subscription level which the user is currently trying to subscribe to and onto which the discount will be applied |
| Min. Presence | The minimum number of days the user has to have a valid subscription to From Level for this rule to be applied |
| Max. Presence | The maximum number of days the user has to have a valid subscription to From Level for this rule to be applied |
| Discount Type | If it's Value, the Value field contains currency, e.g. if you're using Euros and type a value of 20 then 20 Euros will be subtracted from the price. If it's Percent, then the Value field contains a percentage, e.g. if you type a value of 20 then a 20% discount will be applied to the price. |
| Value | See above. |

Practical examples of upgrade rules:

You want to give a 30% discount to users with a SUB1 annual subscription if they renew up to 30 days before their subscription expiration. Both levels set to SUB1, min presence is set to 0, max presence is set to 335 (one year is 365 days, minus 30 days, equals 335 days), discount type Percent, value 30.

You want to give a 15% discount to users with a SUB1 annual subscription if they renew during the last 30 days of their subscription. Both levels set to SUB1, min presence is set to 335 (one year is 365 days, minus 30 days, equals 335 days), max presence set to 365 (one full year) discount type Percent, value 15.

You want to give a 10 Euro discount to users with a SUB1 annual subscription who are now buying a SUB2 subscription. From Level set to SUB1, To Level set to SUB2, min presence set to 0, max presence set to 365 (one year), discount set to Value, value set to 10.

You want users buying both SUB1 and SUB2 to have a 10 Euro discount. First, create a rule like in the previous example. Then create another rule with From Level set to SUB2, To Level set to SUB1, min presence set to 0, max presence set to 365, discount set to Value, value set to 10. This way, whichever subscription they buy first, they'll get the discount when they buy the other subscription as well.

You want to give a complimentary (free) SUB3 subscription to users who have already bought SUB1 and SUB2: you can't do that with Upgrade Rules, as you can not combine multiple subscriptions in a single rule. Instead, use a coupon code with 100% discount and show this coupon code only to subscribers who have already bought a SUB1 and SUB2 subscription using the Restricted Content plugin bundled with Akeeba Subscriptions.

6. Coupons

You can access this feature by clicking on Coupons in the toolbar links area.

This feature allows you to create discount coupon codes. When a user enters a valid coupon code during registration, it's subtracted from the price of his subscription. You can let coupons work on specific subscriptions, specific users, for a predefined time period or even for a predefined number of times before becoming inactive. You can even create 100% discount coupons for giveaways or other similar promotional reasons.

Each coupon supports the following fields:

| | |
|--------------------------|---|
| Title | A descriptive title for your coupon. It is only displayed to you, it is not displayed to your users. |
| Coupon code | The coupon code which the user has to enter in order to receive the discount. Very important: coupon codes are case insensitive, therefore it's best to type them in all capital letters. Moreover, we suggest using only capital Latin letters without accents or diacretics and numbers, so that all users can type them in using their keyboard. |
| Discount type | You can either specify Percent (give an X% discount over the price) or Value (deducts X from the price). It works in conjunction with the Value field below. For example, if you are using Euros as your currency, you set Value here and type in a value of 10, the coupon gives 10 Euro discount. |
| Discount value | See above |
| Published | Set to Yes for the coupon to become active |
| Hits | This is the number of times this coupon has been used. |
| Valid from | When the coupon will become active (note: the date is expressed in the GMT/UTC timezone). Leave blank or set to 0000-00-00 00:00:00 to make the coupon active right now. Note: a published coupon will not be taken into account before its Valid From date. |
| Valid to | When the coupon will become inactive (note: the date is expressed in the GMT/UTC timezone). Leave blank or set to 0000-00-00 00:00:00 to let Akeeba Subscriptions ignore this field and let the coupon active until you manually unpublish it. Note: a published coupon will not be taken into account after its Valid To date. |
| Limit to a specific user | Click on Select to pick a user for which the coupon will be active. For all other users the coupon will have no effect. Leave blank to let all users –even new ones– to use the coupon. |
| Subscription Levels | Choose the subscription levels which the coupon code can be applied to. Leave it blank (or make sure that only the - Select - pseudo-option is selected) to let Akeeba Subscriptions apply the coupon to all current and future subscription levels you are offering on your site. |
| Hits limit | If it is greater than zero, the coupon can be used up to this number of times before it becomes inactive. In other words, when the Hits counter becomes equal to Hits Limit, the coupon can no longer be used by anyone. Leave it blank to not apply any such limit. |

7. Subscriptions management

You can access this feature by clicking on Subscriptions in the toolbar links area.

This is a standard Joomla! administrator page which needs no explanation.

The icons on the left hand side of the user information are Gravatars. If the user has linked a Gravatar to their email address, it is displayed next to their name, otherwise a default avatar (head-shaped cut out) is displayed.

In order to edit a subscription, click on the leftmost column (the numeric ID) of the subscription record. Clicking on the other links will edit the respective item, i.e. clicking on the subscription level will open the subscription level editor.

You can use the Run Integrations button any time in order to have Akeeba Subscriptions re-process all integrations with third party components. This is useful when you suspect an integration has gone wrong, you changed the options in an integration plugin or you just imported a large number of subscriptions.

Using the CSV button allows you to export the current view into a CSV file for further processing in a spreadsheet application such as Microsoft Office Excel, Apple Numbers or OpenOffice.org Calc. Do note that the full set of raw data is dumped to the CSV file.

Important

Clicking on the Published column apparently has no effect. This is not a bug; it's the expected behaviour. The publish status of each subscriptions is defined by the payment state and the valid from/ to dates. If the payment state is marked as "Completed" then the subscription is forced to be active within the time period defined by the valid from / to columns. If you want to disable a paid subscription, set its payment state to cancelled and unpublish it.

8. Affiliates management

Since Akeeba Subscriptions 2.0.b1 there is a very basic affiliate management system. It is extremely (even stupidly!) simple by design. If you want something more complex, you can easily integrate Akeeba Subscriptions with a web-based affiliate management system, like OSI Affiliates.

The concept of affiliates in Akeeba Subscriptions is based on some simple rules:

- You manually create affiliates in the back-end of the component
- Each affiliate is linked to a Joomla! user account and is assigned a commission percentage
- When you pay your affiliate, you manually create an affiliate payment record to let Akeeba Subscriptions know how much you paid him and when
- Each subscription with a payment status of C (Completed) counts towards the amount owed to the affiliate
- Each affiliate payment deducts from the amount owed to the affiliate
- There is no "affiliate report" page of any kind

It's an extremely simple system, suitable for small sites with a handful of trusted affiliates. If you're interested in integrating a much more powerful third party affiliate system, please contact our support.

9. Front-end items

There are four types of front-end menu items you can create:

| | |
|--|---|
| Subscribe • Specific Level | Allows you to create a menu link to the subscription page of a specific Subscription Level |
| Subscription Levels • Awesome Layout | Presents a list of all available subscription levels in a comparison table, using CSS3 styling. It's the most eye-catching presentation layout and it's best to use only if you have a small number of subscription levels (4 to 5) |
| Subscription Levels • All Levels | Presents a list of all available subscription levels as a two column list of boxes. It is a relatively dull presentation mode, but is suitable for a very large number of subscription levels (over 6) |
| My subscriptions • All of my subscriptions | Think of it as a kind of "My Account" page. It shows the user all of his subscriptions and allows him to review the information on each one of them, as well as renew them. |

The subscription page defines two module positions where you can publish modules to further expand the information presented on the page. Publishing modules in the `akeebasubscriptionsheader` position will make them appear at the very top of the page, right above the Steps bar. That's a good place to publish information about SSL security or banners about your latest special offers. Publishing modules in the `akeebasubscriptionsfooter` position will

make them appear right after the end of the subscription form. This is a good place to put a custom HTML module with links to your terms of service, business contact information, etc.

Additionally, the page which lists the subscriptions defines another two module positions where you can publish modules to further expand the information presented on the page. Publishing modules in the `akeebasubscription-slistheader` position will make them appear at the very top of the page, right above the list of subscription levels. Publishing modules in the `akeebasubscriptionslistfooter` position will make them appear right after the end of the subscription levels list.

10. Importing from other components

Note

This feature is temporarily removed from Akeeba Subscriptions 2.0.a1 and 2.0.a2. We plan on including it again in 2.0.b1 and later.

Most likely you are already using a subscriptions system and wouldn't like to have to manually import all of your subscriptions to Akeeba Subscriptions. That's why we are offering an automated import feature. You can access it from Akeeba Subscription's dashboard page, by clicking on the Operations tab, and then on Import.

The options are:

AMBRA.Subscriptions imports subscription levels and subscriptions from Dioscouri's AMBRA Subscriptions (a.k.a. `AMBRA.subs`)

AMBRA.Subscriptions if you are using `AMBRA.subs` with my modified "PayPal with VAT" payment plugin and the with PayPal with customizations for integrating coupons into `AMBRA.subs`, you can click on this option. It will VAT plugin import subscription levels, subscriptions and any coupon codes defined. Moreover, it will also import the extra data stored per user, e.g. their address and VAT number.

Warning

Importing from other components will **DELETE AND REPLACE ANY AND ALL EXISTING DATA IN AKEEBA SUBSCRIPTIONS**. There is **NO TURNING BACK**. Do **NOT** use this option after you have been using Akeeba Subscriptions and have new subscribers on your site.

Please remember to review your Subscription Levels after import. Most notably, the text of the subscription completion and subscription cancellation pages will be missing from all subscription levels.

Chapter 3. Payment methods, integrations and plugins

Akeeba Subscriptions is designed to be a modular system, powered by standard Joomla! plugins. Using plugins you can add payment methods as well as integration with third party software. Moreover, core functionality of Akeeba Subscriptions (like sending out emails upon subscription or email reminders for subscription expiration) is take care of by plugins. This documentation chapter will guide you through the details of each plugin.

1. Payment methods

All payment methods are standard Joomla! plugins, belonging to the `akpayment` group. If you want to enable a payment method, just publish the relevant plugin.

1.1. Paypal

Displayed in the Plugins Manager as: Akeeba Subscriptions Payment - Paypal

This payment method integrates the standard PayPal Website Payments service, available without any setup fee to all verified PayPal clients. In other words, that's the standard way to use PayPal as your payment processor. The options you have in this plugin are:

| | |
|-------------------------|---|
| Payment option title | Leave blank to use "PayPal" or type in a custom name to show to your customer's, e.g. "PayPal, bank account or Credit Card" |
| Merchant ID | Your PayPal email address or your secure merchant ID (if PayPal has assigned you with one) |
| Secure POST | If enabled, the verification postback to PayPal will be over an SSL connection. In order for this to work, your host needs to support SSL connections using cURL to PayPal's IP address block. |
| Sandbox | When enabled, transactions are processed by PayPal Sandbox, i.e. PayPal's testing mode. This is designed for integration tests, BEFORE going live with your site. NEVER, EVER, NO MATTER WHAT YOU DO, SHOULD YOU NOT ENABLE THIS ON PRODUCTION SITES! IF YOU DO, YOU WILL LOSE MONEY AND/OR CUSTOMERS. |
| Sandbox merchant | The dummy merchant email used with PayPal's Sandbox. |
| PayPal Callback Text | (optional) Normally, when the user completes the payment, PayPal shows him a button with the default label "Return to Merchant". If you type in something in here, it will be shown on that button instead of the default label. For example, you can type in something like "Return to Example.com" |
| Custom Header Image | (optional) The URL to an image to be shown on PayPal's checkout page header. It'd better be hosted on an HTTPS URL, otherwise your customers will get a warning about insecure content on their browser. |
| Header Background Color | The hex code of PayPal checkout page's header background color. For example, white is FFFFFFFF (note: there is no # sign in front of the hex color code!) |
| Header Border Color | The hex code of PayPal checkout page's header border color. For example, light gray is CCCCCC (note: there is no # sign in front of the hex color code!) |

1.2. None

Displayed in the Plugins Manager as: Akeeba Subscriptions Payment - None

This is a dummy payment plugin which immediately activates a subscription without payment.

Warning

DO NOT USE THIS ON PRODUCTION SITES. IT IS MEANT FOR TESTING PURPOSES ONLY. ENABLING IT ON A PRODUCTION SITE ALLOWS ANYONE, INCLUDING UNREGISTERED USERS, TO SUBSCRIBE TO ANY SUBSCRIPTION THEY WANT **WITHOUT PAYING ANYTHING AT ALL**. You have been strongly warned.

1.3. WorldPay

Displayed in the Plugins Manager as: Akeeba Subscriptions Payment - WorldPay

This payment method integrates the WorldPay Hosted Payment Page payments service, available with all entry-level offerings of WorldPay.

Important

WorldPay's Hosted Payment Page mode does not allow your visitors to come back to your site. Therefore, they will not see your custom payment success and cancellation messages. You will instead have to use custom HTML pages uploaded to WorldPay's servers.

Warning

You need to perform some setup before using this method (the following steps are taken verbatim from WorldPay's Test and Go Live documentation):

- Log in to the WorldPay Merchant Interface
- Select Installations from the left hand navigation
- Choose an installation and select the Integration Setup button for either the TEST or PRODUCTION environment
- Check the Enable Payment Response checkbox
- Enter the WPDISPLAY ITEM tag which includes the dynamic url variable into the Payment Response URL input field:

```
<wpdisplay item=MC_callback>
```
- Enter a password into the Payment Response Password input field; this is your Callback Password and you will have to copy it to the plugin's setup parameters
- Select the Save Changes button

Please make sure that you have completed all of the steps above and have configured all the parameters BEFORE enabling this plugin. Do not ask for support if you haven't done that yet.

The options you have in this plugin are:

| | |
|----------------------|---|
| Payment option title | Leave blank to use "WorldPay" or type in a custom name to show to your customer's, e.g. "Visa, MasterCard or American Express cards" |
| Installation ID | Your WorldPay installation ID, as communicated to you by WorldPay. |
| Callback Password | The callback password you have specified in your WorldPay account, used to verify that the payment notification come, indeed, from WorldPay and avoid fraudulent requests |
| Test mode | When enabled, transactions are processed by WorldPay's test server, i.e. WorldPay's testing mode. This is designed for integration tests, BEFORE going live with your site. NEVER, EVER, NO MATTER WHAT YOU DO, SHOULD YOU NOT ENABLE THIS ON PRODUCTION SITES! IF YOU DO, YOU WILL LOSE MONEY AND/OR CUSTOMERS. This option should be used only in the integration testing phase during your initial setup. During this phase, visitors can use one of the dummy card numbers to buy subscriptions without money changing hands. When you're ready to go live with your site, disable this feature. |

1.4. Off-line

Displayed in the Plugins Manager as: Akeeba Subscriptions Payment - Off-line

This is a payment plugin which lets your users complete their payment off-line, e.g. through bank (wire) transfer, money check, Western Union or any other means of payment that you allow. It works very simply: it just presents your user with a customizable instructions message. In that message you can use the variables `{SUBSCRIPTION}` to display the subscription ID and `{AMOUNT}` to display the amount due. Please use all caps for these variables and include the curly braces.

Once the user effects the payment, simply go to your Subscriptions view, enable his subscriptions and set the payment status to "Completed". That's all!

1.5. 2Checkout Standard Purchase Routine

Displayed in the Plugins Manager as: Akeeba Subscriptions Payment - 2Checkout Standard Purchase Routine

This payment method integrates the 2Checkout Standard Purchase Routine payments service (commonly simply referred to as 2checkout or 2CO).

Important

Akeeba Subscriptions can not pass the currency to 2Checkout. You **MUST** make sure that the currency configured in 2Checkout and the currency displayed in your site match to avoid nasty surprises.

The options you have in this plugin are:

| | |
|----------------------|---|
| Payment option title | Leave blank to use "2Checkout" or type in a custom name to show to your customer's, e.g. "Visa, MasterCard or American Express cards" |
| Seller ID | Your 2Checkout seller numeric ID, as communicated to you by 2Checkout. |
| Secret word | The secret word you have specified in your 2Checkout account, used to verify that the payment notification come, indeed, from 2Checkout and avoid fraudulent requests. It is case sensitive, i.e. ABC, abc and Abc are three different secret words. |
| Demo mode | When enabled, transactions are processed by 2Checkout's as test payments. This is designed for integration tests, BEFORE going live with your site. NEVER, EVER, NO MATTER WHAT YOU DO, SHOULD YOU NOT ENABLE THIS ON PRODUCTION SITES! IF YOU DO, |

YOU WILL LOSE MONEY AND/OR CUSTOMERS. This option should be used only in the integration testing phase during your initial setup. When you're ready to go live with your site, disable this feature.

Warning

NONE of the Demo payments will be accepted by the plugin! Demo payments deliberately carry an invalid signature and will be marked as "Cancelled" in Akeeba Subscriptions. The log file will mention them as fraud attempts. This is deliberate and should not be considered a bug.

1.6. ccAvenue

Displayed in the Plugins Manager as: Akeeba Subscriptions Payment - ccAvenue

This payment method integrates the ccAvenue payments service, the leading payment processor in India. It can be used for transactions in Indian Rupees or US Dollars.

Important

Akeeba Subscriptions can not pass the currency to ccAvenue. You **MUST** make sure that the currency configured in ccAvenue and the currency displayed in your site match to avoid nasty surprises.

The options you have in this plugin are:

| | |
|----------------------|---|
| Payment option title | Leave blank to use "ccAvenue" or type in a custom name to show to your customer's, e.g. "Visa, MasterCard or American Express cards" |
| Merchant ID | Your ccAvenue Merchant ID, as communicated to you by ccAvenue. |
| Working Key | The password you have specified in your ccAvenue account, used to verify that the payment notification come, indeed, from ccAvenue and avoid fraudulent requests. It is case sensitive, i.e. ABC, abc and Abc are three different secret words. |

1.7. eWay

Displayed in the Plugins Manager as: Akeeba Subscriptions Payment - eWay

This payment method integrates the eWay payments service. It can work with all three local payment services offered by eWay (Australia, New Zealand and UK/Europe).

Important

Each local eWay payment service supports different currencies. The Australian service only supports AUD, the New Zealand service only supports NZD and the UK service only supports GBP and EUR - as far as I can see. You have to make sure you have signed up to the correct payment service and set up the currency in Akeeba Subscriptions accordingly.

The options you have with this plugin are:

| | |
|----------------------|---|
| Payment option title | Leave blank to use "eWay" or type in a custom name to show to your customer's, e.g. "Credit Card" |
| eWay Site | Select which local eWay site you are signed up with. IMPORTANT! If this selection is incorrect, you will get an error when you try to subscribe. |

| | |
|------------------|--|
| eWay Customer ID | Enter your numeric eWay Customer ID. Use 87654321 for testing. |
| eWay Username | Enter your eWay username. Use TestAccount for testing. |
| Company logo URL | Optional. The URL of your logo image. It can be hosted on your website but MUST use https. This is the top image block of the payment web page and it is restricted to 960x65 pixels. A default secure image will be used if you leave this field empty. |
| Page banner URL | Optional. The URL of the payment page's banner. It can be hosted on your website but MUST use https. This is the second image block of the payment web page and it is restricted to 960x65 pixels. A default secure image will be used if you leave this field empty. |
| Language | This automatically translates headings and button text to the desired language. The default is English. |
| Company name | Optional. This will be displayed as the company is purchasing from. Including this is highly recommended. |
| Page title | Optional. This value is used to populate the browser's title bar at the top of the screen. |
| Page description | Optional. Used as a greeting message to the customer and is displayed above the customer's order details. |
| Page footer | Optional. The page footer text can be customised and populated below the customer's order details. Useful for contact information. |

2. Integration with third party software

One of the most important aspects about subscription systems is their ability to integrate with third party software. A subscription all by itself means pretty much nothing. It's just a "flag". The added value is when this subscription allows the user holding it to actually do something on the site. At this point in time, there are only a handful of integrations. If you are interested in more integrations let us know and we'll add them to our implementation to-do list.

All integrations with third party applications are standard Joomla! plugins, published in the akeebasubs group.

2.1. Community Builder integration

Displayed in the Plugins Manager as: Akeeba Subscriptions - Community Builder Integration

This integration plugin allows you to add and remove users from Community Builder groups based on their subscription status. Moreover, it allows you to automatically authorize subscribers and deauthorize them when their subscriptions expires, essentially controlling their access to your Community Builder community based on their subscription status.

Please note that this is a smart integration, exactly as described in the JUGA plugin in this documentation; instead of adding users when a subscription goes active and removing them when a subscription goes inactive, Akeeba Subscriptions plugins take a look at all of the user's active and inactive subscriptions and decide if the user should be added to or removed from Community Builder groups or authorized/deauthorized.

The plugin has these options:

| | |
|-------------------------|--|
| Add to CB groups | When enabled, new subscribers are automatically added as Community Builder users (a default profile is created for them that they can later fill in) |
| Automatic authorization | Select the subscription levels will be automatically authorized. When a user has an active subscription to one of these levels, he will be marked as an authorized user who has read the terms |

of use in Community Builder, essentially allowing her to have access to the Community Builder community on your site.

Automatic deauthorization Select the subscription levels will be automatically deauthorized. When a user no longer has an active subscription to any of these levels, he will be marked as an unauthorized user who has not read the terms of use in Community Builder, essentially not allowing her to have access to the Community Builder community on your site any more.

2.2. ccInvoices integration

Displayed in the Plugins Manager as: Akeeba Subscriptions - ccInvoices Integration

This integration allows you to automatically issue and send (paid) invoices to your customers once they complete a successful registration. When this plugin is enabled, the following will take place when a user signs up and his payment is successful:

- If he doesn't exist as a ccInvoice contact, a new contact will be created for the user.
- A new paid invoice is generated with the item description reading "Subscription to *level title*" where "level title" is the title of the subscription level the user subscribed to.
- A PDF of the invoice, based on the default template, is generated and emailed to the user

Please note that the default email text in ccInvoices prompts the receiver to pay the invoice by the due date. You may want to change the wording, as the invoices generated from subscriptions are already paid for. You don't want to confuse your customers!

The plugin has no parameters. If you would like to customise its behaviour, please use our forum to post a feature request, describing what you have in mind.

2.3. Community ACL integration

Displayed in the Plugins Manager as: Akeeba Subscriptions - Community ACL Integration

This integration plugin allows you to add and remove users from Community ACL Groups, Roles and Functions based on their subscription status.

Please note that this is a smart integration, exactly as described in the JUGA plugin in this documentation; instead of adding users when a subscription goes active and removing them when a subscription goes inactive, Akeeba Subscriptions plugins take a look at all of the user's active and inactive subscriptions and decide which groups the user should be added to or removed from.

Before we begin taking a look at the plugin options, please note that Community ACL defines Groups, Roles and Functions. To make things more complicated, a Role can belong to a Group or not and a Function can belong to a Rule or not. Are you confused yet? Tell me about it... Unfortunately, we have to follow the same inconvenient structure for the plugin parameters. Each entry which can be assigned or removed from the user's record can therefore consist of a Group, a Group and a Role or Group, Role and Function. For example:

My Group

defines an entry which only mentions a Group. The Role assigned to the user will be the first Role which belongs to My Group or, if no roles are specifically defined for My Group, the first Role which does not belong to any group (usually the default "Subscriber" role created by Community ACL during installation). The function assigned to the user will be the first function which belongs to the selected Role or, if none is defined, the first function which does not belong to any role.

My Group | My Role

defines an entry which only mentions a Group and a Role. If there is a role named My Role belonging to My Group it will be used. Otherwise, we'll try to find a role named My Role which doesn't belong to any group whatsoever. If this fails, the Role assigned to the user will be the first Role which belongs to My Group or, if no roles are specifically defined for My Group, the first Role which does not belong to any group (usually the default "Subscriber" role created by Community ACL during installation). The function assigned to the user will be the first function which belongs to the selected Role or, if none is defined, the first function which does not belong to any role.

My Group|My Role|My Function

defines an entry which mentions a Group, a Role and a Function. If there is a role named My Role belonging to My Group it will be used. Otherwise, we'll try to find a role named My Role which doesn't belong to any group whatsoever. If this fails, the Role assigned to the user will be the first Role which belongs to My Group or, if no roles are specifically defined for My Group, the first Role which does not belong to any group (usually the default "Subscriber" role created by Community ACL during installation). The function assigned to the user will be the one named My Function, belonging to the selected Role. If it does not exist, we'll try to find a function named My Function which does not belong to a role. If there is none, we'll use the first function which belongs to the selected Role or, if none is defined, the first function which does not belong to any role.

Tip

It's impossible to understand any of the above. It's impossible to setup ACLs with the complexity introduced by Groups, Roles and Functions. Take my advice and only use Groups in both Community ACL and this plugin and just forget about Roles and Functions if you value your sanity.

The plugin has these options:

Add to Community ACL groups This is a list for assigning subscription levels to Community ACL groups. When a user has an active subscription to the specified level, he'll be added to the Community ACL group/role/function mentioned on the right hand of the equal sign. Each assignment is done like this:

```
LEVEL1=entry1 , entry2 , entry3
```

Where entry1, entry2 and entry3 is a Group/Role/Function entry as described above.

If you want to assign multiple subscription levels to multiple entries, you have to do something like that (separate multiple lines with a single press of the ENTER key on your keyboard):

```
LEVEL1=entry1 , entry2 , entry3  
LEVEL2=entry2  
LEVEL3=entry4 , entry1
```

Remove from Community ACL groups Likewise, this is the list of the Community ACL groups to remove a user from when his subscription to the specified level is no longer active. Do note that you can specify a different set of groups to remove the user from than the ones you are adding him to. For example:

```
LEVEL1=entry1 , entry2  
LEVEL2=entry2  
LEVEL3=entry1
```

2.4. DOCman Integration

Displayed in the Plugins Manager as: Akeeba Subscriptions - DOCman Integration

This integration plugin allows you to add and remove users from DOCman groups based on their subscription status. This allows you to control downloads/uploads to your DOCman downloads based on a user's subscription status and can form the basis of a commercial file distribution system.

Please note that this is a smart integration, exactly as described in the JUGA plugin in this documentation; instead of adding users when a subscription goes active and removing them when a subscription goes inactive, Akeeba Subscriptions plugins take a look at all of the user's active and inactive subscriptions and decide which groups the user should be added to or removed from.

The plugin has these options:

Add to DOCman groups This is a list for assigning subscription levels to DOCman groups. When a user has an active subscription to the specified level, he'll be added to the DOCman group mentioned on the right hand of the equal sign. Each assignment is done like this:

```
LEVEL1=Group 1,Group 2,Group 3
```

Where LEVEL1 is the Title of the subscription level and Group 1, Group 2 and Group 3 are the names of the DOCman groups you want to add a user to when he subscribes. If you want to assign multiple subscription levels to multiple groups, you have to do something like that (separate multiple lines with a single press of the ENTER key on your keyboard):

```
LEVEL1=Group 1,Group 2,Group 3
LEVEL2=Group 2
LEVEL3=Group 4,Group 1
```

You can also use the numeric IDs of subscription levels or DOCman groups instead of their titles. However, we consider working with numeric IDs very counter-intuitive.

Remove from DOCman groups Likewise, this is the list of the DOCman groups to remove a user from when his subscription to the specified level is no longer active. Do note that you can specify a different set of groups to remove the user from than the ones you are adding him to. For example:

```
LEVEL1=Group 1,Group 2
LEVEL2=Group 2
LEVEL3=Group 1
```

In this example, together with the rules displayed in "Add to DOCman groups", when a subscription to the LEVEL1 level expires, the user will still belong to DOCman Group 3, as it's not removed from his account; only Group 1 and Group 2 are removed. Likewise, if the user has an expired LEVEL1 subscription and an active LEVEL2 subscription he'll now belong to Group 2 and Group 3: Group 3 because it's not removed when his subscription expires and Group 2 because his active LEVEL2 subscription suggests that he should be granted Group 2 access despite his expired LEVEL1 subscription.

If you find this hard to understand, join the club. ACLs are not meant to be easy to setup or understand. It's a power feature and, as such, poses a great difficulty to even the most seasoned web site integrators. Take your time and experiment on a dev copy of your site before going live.

2.5. JCE Integration

Displayed in the Plugins Manager as: Akeeba Subscriptions - JCE Integration

This integration plugin allows you to add and remove users from JCE (Joomla! Content Editor) groups based on their subscription status. This allows you to control the editing permissions of individual users based on their subscription status.

Please note that this is a smart integration, exactly as described in the JUGA plugin earlier in this documentation; instead of adding users when a subscription goes active and removing them when a subscription goes inactive, Akeeba Subscriptions plugins take a look at all of the user's active and inactive subscriptions and decide which groups the user should be added to or removed from.

The plugin has these options:

Add to JCE groups This is a list for assigning subscription levels to JCE groups. When a user has an active subscription to the specified level, he'll be added to the JCE group mentioned on the right hand of the equal sign. Each assignment is done like this:

```
LEVEL1=Group 1,Group 2,Group 3
```

Where LEVEL1 is the Title of the subscription level and Group 1, Group 2 and Group 3 are the names of the JCE groups you want to add a user to when he subscribes. If you want to assign multiple subscription levels to multiple groups, you have to do something like that (separate multiple lines with a single press of the ENTER key on your keyboard):

```
LEVEL1=Group 1,Group 2,Group 3
LEVEL2=Group 2
LEVEL3=Group 4,Group 1
```

You can also use the numeric IDs of subscription levels or JCE groups instead of their titles. However, we consider working with numeric IDs very counter-intuitive.

Remove from JCE groups Likewise, this is the list of the JCE groups to remove a user from when his subscription to the specified level is no longer active. Do note that you can specify a different set of groups to remove the user from than the ones you are adding him to. For example:

```
LEVEL1=Group 1,Group 2
LEVEL2=Group 2
LEVEL3=Group 1
```

In this example, together with the rules displayed in "Add to JCE groups", when a subscription to the LEVEL1 level expires, the user will still belong to JCE Group 3, as it's not removed from his account; only Group 1 and Group 2 are removed. Likewise, if the user has an expired LEVEL1 subscription and an active LEVEL2 subscription he'll now belong to Group 2 and Group 3: Group 3 because it's not removed when his subscription expires and Group 2 because his active LEVEL2 subscription suggests that he should be granted Group 2 access despite his expired LEVEL1 subscription.

If you find this hard to understand, join the club. ACLs are not meant to be easy to setup or understand. It's a power feature and, as such, poses a great difficulty to even the most seasoned web site integrators. Take your time and experiment on a dev copy of your site before going live.

2.6. JomSocial integration

Displayed in the Plugins Manager as: Akeeba Subscriptions - JomSocial Integration

This integration plugin allows you to add and remove users from JomSocial groups based on their subscription status. This allows you to control access to your JomSocial groups based on a user's subscription status.

Please note that this is a smart integration, exactly as described in the JUGA plugin in this documentation; instead of adding users when a subscription goes active and removing them when a subscription goes inactive, Akeeba Subscriptions plugins take a look at all of the user's active and inactive subscriptions and decide which groups the user should be added to or removed from.

The plugin has these options:

Add to JomSocial groups This is a list for assigning subscription levels to JomSocial groups. When a user has an active subscription to the specified level, he'll be added to the JomSocial group mentioned on the right hand of the equal sign. Each assignment is done like this:

```
LEVEL1=Group 1,Group 2,Group 3
```

Where LEVEL1 is the Title of the subscription level and Group 1, Group 2 and Group 3 are the names of the JomSocial groups you want to add a user to when he subscribes. If you want to assign multiple subscription levels to multiple groups, you have to do something like that (separate multiple lines with a single press of the ENTER key on your keyboard):

```
LEVEL1=Group 1,Group 2,Group 3
LEVEL2=Group 2
LEVEL3=Group 4,Group 1
```

You can also use the numeric IDs of subscription levels or JomSocial groups instead of their titles. However, we consider working with numeric IDs very counter-intuitive.

Remove from JomSocial groups Likewise, this is the list of the JomSocial groups to remove a user from when his subscription to the specified level is no longer active. Do note that you can specify a different set of groups to remove the user from than the ones you are adding him to. For example:

```
LEVEL1=Group 1,Group 2
LEVEL2=Group 2
LEVEL3=Group 1
```

In this example, together with the rules displayed in "Add to JomSocial groups", when a subscription to the LEVEL1 level expires, the user will still belong to JomSocial Group 3, as it's not removed from his account; only Group 1 and Group 2 are removed. Likewise, if the user has an expired LEVEL1 subscription and an active LEVEL2 subscription he'll now belong to Group 2 and Group 3: Group 3 because it's not removed when his subscription expires and Group 2 because his active LEVEL2 subscription suggests that he should be granted Group 2 access despite his expired LEVEL1 subscription.

If you find this hard to understand, join the club. ACLs are not meant to be easy to setup or understand. It's a power feature and, as such, poses a great difficulty to even the most seasoned web site integrators. Take your time and experiment on a dev copy of your site before going live.

2.7. Joomla! 1.6 User Groups Integration

Displayed in the Plugins Manager as: Akeeba Subscriptions - Joomla! 1.6 Usergroups Integration

This integration plugin allows you to add and remove users from Joomla! 1.6 groups based on their subscription status. This allows you to fully utilize Joomla! 1.6's built-in ACL capabilities on compatible extensions.

Please note that this is a smart integration, exactly as described in the JUGA plugin earlier in this documentation; instead of adding users when a subscription goes active and removing them when a subscription goes inactive, Akeeba Subscriptions plugins take a look at all of the user's active and inactive subscriptions and decide which groups the user should be added to or removed from.

The plugin has these options:

Add to Joomla! 1.6 groups This is a list for assigning subscription levels to Joomla! groups. When a user has an active subscription to the specified level, he'll be added to the Joomla! group mentioned on the right hand of the equal sign. Each assignment is done like this:

```
LEVEL1=Group 1,Group 2,Group 3
```

Where LEVEL1 is the Title of the subscription level and Group 1, Group 2 and Group 3 are the names of the Joomla! groups you want to add a user to when he subscribes. If you want to

assign multiple subscription levels to multiple groups, you have to do something like that (separate multiple lines with a single press of the ENTER key on your keyboard):

```
LEVEL1=Group 1,Group 2,Group 3
LEVEL2=Group 2
LEVEL3=Group 4,Group 1
```

You can also use the numeric IDs of subscription levels or Joomla! groups instead of their titles. However, we consider working with numeric IDs very counter-intuitive.

Remove from Joomla! 1.6 groups

Likewise, this is the list of the Joomla! groups to remove a user from when his subscription to the specified level is no longer active. Do note that you can specify a different set of groups to remove the user from than the ones you are adding him to. For example:

```
LEVEL1=Group 1,Group 2
LEVEL2=Group 2
LEVEL3=Group 1
```

In this example, together with the rules displayed in "Add to Joomla! groups", when a subscription to the LEVEL1 level expires, the user will still belong to Joomla!'s Group 3, as it's not removed from his account; only Group 1 and Group 2 are removed. Likewise, if the user has an expired LEVEL1 subscription and an active LEVEL2 subscription he'll now belong to Group 2 and Group 3: Group 3 because it's not removed when his subscription expires and Group 2 because his active LEVEL2 subscription suggests that he should be granted Group 2 access despite his expired LEVEL1 subscription.

If you find this hard to understand, join the club. ACLs are not meant to be easy to setup or understand. It's a power feature and, as such, poses a great difficulty to even the most seasoned web site integrators. Take your time and experiment on a dev copy of your site before going live.

2.8. JUGA Integration

Displayed in the Plugins Manager as: Akeeba Subscriptions - JUGA Integration

This integration plugin allows you to add and remove users from JUGA groups based on their subscription status. Please note that this is a smart integration; instead of adding users when a subscription goes active and removing them when a subscription goes inactive, Akeeba Subscriptions plugins take a look at all of the user's active and inactive subscriptions and decide which groups the user should be added to or removed from. The idea is this:

1. All groups to be added to, based on active subscriptions of the user, are assembled to the ADD list.
2. All groups to be removed from, based on inactive subscriptions of the user, are assembled to the REMOVE list.
3. If a JUGA group appears in both the ADD and REMOVE list, it is deleted from the REMOVE list. In other words "add" trumps "delete". In order to understand this further, here's the deal. If one subscription with access to a JUGA group expires, but another subscription granting access to the very same group is still active, the user's access to this group must remain active.

The plugin has these options:

Add to JUGA groups

This is a list for assigning subscription levels to JUGA groups. When a user has an active subscription to the specified level, he'll be added to the JUGA group mentioned on the right hand of the equal sign. Each assignment is done like this:

```
LEVEL1=Group 1,Group 2,Group 3
```

Where LEVEL1 is the Title of the subscription level and Group 1, Group 2 and Group 3 are the names of the JUGA groups you want to add a user to when he subscribes. If you want to

assign multiple subscription levels to multiple groups, you have to do something like that (separate multiple lines with a single press of the ENTER key on your keyboard):

```
LEVEL1=Group 1,Group 2,Group 3
LEVEL2=Group 2
LEVEL3=Group 4,Group 1
```

You can also use the numeric IDs of subscription levels or JUGA groups instead of their titles. However, we consider working with numeric IDs very counter-intuitive.

Remove from JUGA groups

Likewise, this is the list of the JUGA groups to remove a user from when his subscription to the specified level is no longer active. Do note that you can specify a different set of groups to remove the user from than the ones you are adding him to. For example:

```
LEVEL1=Group 1,Group 2
LEVEL2=Group 2
LEVEL3=Group 1
```

In this example, together with the rules displayed in "Add to JUGA groups", when a subscription to the LEVEL1 level expires, the user will still belong to JUGA's Group 3, as it's not removed from his account; only Group 1 and Group 2 are removed. Likewise, if the user has an expired LEVEL1 subscription and an active LEVEL2 subscription he'll now belong to Group 2 and Group 3: Group 3 because it's not removed when his subscription expires and Group 2 because his active LEVEL2 subscription suggests that he should be granted Group 2 access despite his expired LEVEL1 subscription.

If you find this hard to understand, join the club. ACLs are not meant to be easy to setup or understand. It's a power feature and, as such, poses a great difficulty to even the most seasoned web site integrators. Take your time and experiment on a dev copy of your site before going live.

2.9. JoomlaXI JomSocial User Profile Types integration

Displayed in the Plugins Manager as: Akeeba Subscriptions - JoomlaXI JomSocial User Profile Types Integration

This integration plugin allows you to add and remove users from JomSocial user types based on their subscription status. This allows you to control access to your JomSocial groups based on a user's subscription status. This integration requires that both JomSocial by Azrul and JomSocial User Profile Types by JoomlaXI be installed on your site.

Please note that this is a smart integration, exactly as described in the JUGA plugin in this documentation; instead of adding users when a subscription goes active and removing them when a subscription goes inactive, Akeeba Subscriptions plugins take a look at all of the user's active and inactive subscriptions and decide which groups the user should be added to or removed from.

The plugin has these options:

Add to JoomlaXI JomSocial User Profile Types

This is a list for assigning subscription levels to JoomlaXI JomSocial User Profile Types. When a user has an active subscription to the specified level, he'll be added to the JoomlaXI JomSocial User Profile Type mentioned on the right hand of the equal sign. Each assignment is done like this:

```
LEVEL1=Group 1,Group 2,Group 3
```

Where LEVEL1 is the Title of the subscription level and Group 1, Group 2 and Group 3 are the names of the JoomlaXI JomSocial User Profile Types you want to add a user to when he subscribes. If you want to assign multiple subscription levels to multiple groups, you have to do something like that (separate multiple lines with a single press of the ENTER key on your keyboard):

```
LEVEL1=Group 1,Group 2,Group 3  
LEVEL2=Group 2  
LEVEL3=Group 4,Group 1
```

You can also use the numeric IDs of subscription levels or JoomlaXI JomSocial User Profile Types instead of their titles. However, we consider working with numeric IDs very counter-intuitive.

Remove from JoomlaXI JomSocial User Profile Types

Likewise, this is the list of the JoomlaXI JomSocial User Profile Types to remove a user from when his subscription to the specified level is no longer active. Do note that you can specify a different set of groups to remove the user from than the ones you are adding him to. For example:

```
LEVEL1=Group 1,Group 2  
LEVEL2=Group 2  
LEVEL3=Group 1
```

In this example, together with the rules displayed in "Add to JoomlaXI JomSocial User Profile Types", when a subscription to the LEVEL1 level expires, the user will still belong to JoomlaXI JomSocial User Profile Types Group 3, as it's not removed from his account; only Group 1 and Group 2 are removed. Likewise, if the user has an expired LEVEL1 subscription and an active LEVEL2 subscription he'll now belong to Group 2 and Group 3: Group 3 because it's not removed when his subscription expires and Group 2 because his active LEVEL2 subscription suggests that he should be granted Group 2 access despite his expired LEVEL1 subscription.

If you find this hard to understand, join the club. ACLs are not meant to be easy to setup or understand. It's a power feature and, as such, poses a great difficulty to even the most seasoned web site integrators. Take your time and experiment on a dev copy of your site before going live.

2.10. K2 Integration

Displayed in the Plugins Manager as: Akeeba Subscriptions - K2 Integration

This integration plugin allows you to add and remove users from K2 1.6 groups based on their subscription status. This allows you to fully utilize K2's limited ACL capabilities (basically, just control front-end editing).

Warning

K2 only allows a user to belong to one and only one group. This means that only the latest subscription's K2 group will be applied to your user if he holds several subscriptions.

The plugin has these options:

Add to K2 groups This is a list for assigning subscription levels to NinjaBoard groups. When a user has an active subscription to the specified level, he'll be added to the NinjaBoard group mentioned on the right hand of the equal sign. Each assignment is done like this:

```
LEVEL1=Group 1
```

Where LEVEL1 is the Title of the subscription level and Group 1 is the names of the K2 group you want to add a user to when he subscribes. If you want to assign multiple subscription levels to multiple groups, you have to do something like that (separate multiple lines with a single press of the ENTER key on your keyboard):

```
LEVEL1=Group 1  
LEVEL2=Group 2  
LEVEL3=Group 3
```

You can also use the numeric IDs of subscription levels or K2 groups instead of their titles. However, we consider working with numeric IDs very counter-intuitive.

Remove from K2 groups Likewise, this is the list of the K2 groups to remove a user from when his subscription to the specified level is no longer active. Do note that you can specify a different set of groups to remove the user from than the ones you are adding him to. For example:

```
LEVEL1=Group 1
LEVEL2=Group 2
LEVEL3=Group 3
```

2.11. NinjaBoard Integration

Displayed in the Plugins Manager as: Akeeba Subscriptions - NinjaBoard Integration

This integration plugin allows you to add and remove users from NinjaBoard groups based on their subscription status. This allows you to control view/post/attachments access to your NinjaBoard forums based on a user's subscription status and can form the basis of a commercial support forum, like the one we use at AkeebaBackup.com.

Please note that this is a smart integration, exactly as described in the JUCA plugin earlier in this documentation; instead of adding users when a subscription goes active and removing them when a subscription goes inactive, Akeeba Subscriptions plugins take a look at all of the user's active and inactive subscriptions and decide which groups the user should be added to or removed from.

The plugin has these options:

Add to NinjaBoard groups This is a list for assigning subscription levels to NinjaBoard groups. When a user has an active subscription to the specified level, he'll be added to the NinjaBoard group mentioned on the right hand of the equal sign. Each assignment is done like this:

```
LEVEL1=Group 1,Group 2,Group 3
```

Where LEVEL1 is the Title of the subscription level and Group 1, Group 2 and Group 3 are the names of the NinjaBoard groups you want to add a user to when he subscribes. If you want to assign multiple subscription levels to multiple groups, you have to do something like that (separate multiple lines with a single press of the ENTER key on your keyboard):

```
LEVEL1=Group 1,Group 2,Group 3
LEVEL2=Group 2
LEVEL3=Group 4,Group 1
```

You can also use the numeric IDs of subscription levels or NinjaBoard groups instead of their titles. However, we consider working with numeric IDs very counter-intuitive.

Remove from NinjaBoard groups Likewise, this is the list of the NinjaBoard groups to remove a user from when his subscription to the specified level is no longer active. Do note that you can specify a different set of groups to remove the user from than the ones you are adding him to. For example:

```
LEVEL1=Group 1,Group 2
LEVEL2=Group 2
LEVEL3=Group 1
```

In this example, together with the rules displayed in "Add to NinjaBoard groups", when a subscription to the LEVEL1 level expires, the user will still belong to NinjaBoard's Group 3, as it's not removed from his account; only Group 1 and Group 2 are removed. Likewise, if the user has an expired LEVEL1 subscription and an active LEVEL2 subscription

he'll now belong to Group 2 and Group 3: Group 3 because it's not removed when his subscription expires and Group 2 because his active LEVEL2 subscription suggests that he should be granted Group 2 access despite his expired LEVEL1 subscription.

If you find this hard to understand, join the club. ACLs are not meant to be easy to setup or understand. It's a power feature and, as such, poses a great difficulty to even the most seasoned web site integrators. Take your time and experiment on a dev copy of your site before going live.

2.12. Tienda Integration

Displayed in the Plugins Manager as: Akeeba Subscriptions - Tienda Integration

This integration plugin allows you to add and remove users from Tienda shopper groups based on their subscription status. This allows you to set special discounts to subscribers (e.g. a discount club) or other Tienda tricks which are based on the shopper groups a user belongs to.

Please note that this is a smart integration, exactly as described in the JUGA plugin earlier in this documentation; instead of adding users when a subscription goes active and removing them when a subscription goes inactive, Akeeba Subscriptions plugins take a look at all of the user's active and inactive subscriptions and decide which groups the user should be added to or removed from.

The plugin has these options:

Add to Tienda groups This is a list for assigning subscription levels to Tienda groups. When a user has an active subscription to the specified level, he'll be added to the Tienda group mentioned on the right hand of the equal sign. Each assignment is done like this:

```
LEVEL1=Group 1,Group 2,Group 3
```

Where LEVEL1 is the Title of the subscription level and Group 1, Group 2 and Group 3 are the names of the Tienda groups you want to add a user to when he subscribes. If you want to assign multiple subscription levels to multiple groups, you have to do something like that (separate multiple lines with a single press of the ENTER key on your keyboard):

```
LEVEL1=Group 1,Group 2,Group 3
LEVEL2=Group 2
LEVEL3=Group 4,Group 1
```

You can also use the numeric IDs of subscription levels or Tienda groups instead of their titles. However, we consider working with numeric IDs very counter-intuitive.

Remove from Tienda groups Likewise, this is the list of the Tienda groups to remove a user from when his subscription to the specified level is no longer active. Do note that you can specify a different set of groups to remove the user from than the ones you are adding him to. For example:

```
LEVEL1=Group 1,Group 2
LEVEL2=Group 2
LEVEL3=Group 1
```

In this example, together with the rules displayed in "Add to Tienda groups", when a subscription to the LEVEL1 level expires, the user will still belong to Tienda Group 3, as it's not removed from his account; only Group 1 and Group 2 are removed. Likewise, if the user has an expired LEVEL1 subscription and an active LEVEL2 subscription he'll now belong to Group 2 and Group 3: Group 3 because it's not removed when his subscription expires and Group 2 because his active LEVEL2 subscription suggests that he should be granted Group 2 access despite his expired LEVEL1 subscription.

If you find this hard to understand, join the club. ACLs are not meant to be easy to setup or understand. It's a power feature and, as such, poses a great difficulty to even the most seasoned web site integrators. Take your time and experiment on a dev copy of your site before going live.

2.13. Delete users on subscription expiration

Displayed in the Plugins Manager as: Akeeba Subscriptions - Delete users with expired subscriptions

When this plugin is enabled, when all of the subscriptions of a user expire, his Joomla! user record will be permanently deleted from the site. This is equivalent to using Manage Users in the back-end to delete a user. We consider this plugin to be suitable for a tiny minority of sites which want to allow access to their content only to registered users but wish to receive payment to register users.

Warning

DELETING USERS IS AN IRREVERSIBLE PROCESS! ONCE A USER IS DELETED, ALL INFORMATION ASSOCIATED WITH HIM (INCLUDING SUBSCRIPTIONS INFORMATION) IS GONE. FOREVER. YOU WILL NOT BE ABLE TO RETRIEVE THEM EVER AGAIN.

This is how this feature is supposed to work. When every information pertaining to a user with an expired subscription vanishes to thin air do not complain that it is unacceptable or a bug; it's not; it's how this is designed to work.

2.14. VirtueMart Integration

Displayed in the Plugins Manager as: Akeeba Subscriptions - VirtueMart Integration

Important

This plugin only works with VirtueMart 1.x. It will NOT work with VirtueMart 2.x.

This integration plugin allows you to add and remove users from VirtueMart 1.x shopper groups based on their subscription status. This allows you to set special discounts to subscribers (e.g. a discount club) or other VirtueMart tricks which are based on the shopper groups a user belongs to.

Please note that this is a smart integration, exactly as described in the JUGA plugin earlier in this documentation; instead of adding users when a subscription goes active and removing them when a subscription goes inactive, Akeeba Subscriptions plugins take a look at all of the user's active and inactive subscriptions and decide which groups the user should be added to or removed from.

The plugin has these options:

Add to Virtue-
Mart groups This is a list for assigning subscription levels to VirtueMart groups. When a user has an active subscription to the specified level, he'll be added to the VirtueMart group mentioned on the right hand of the equal sign. Each assignment is done like this:

```
LEVEL1=Group 1,Group 2,Group 3
```

Where LEVEL1 is the Title of the subscription level and Group 1, Group 2 and Group 3 are the names of the VirtueMart groups you want to add a user to when he subscribes. If you want to assign multiple subscription levels to multiple groups, you have to do something like that (separate multiple lines with a single press of the ENTER key on your keyboard):

```
LEVEL1=Group 1,Group 2,Group 3  
LEVEL2=Group 2
```

```
LEVEL3=Group 4,Group 1
```

You can also use the numeric IDs of subscription levels or VirtueMart groups instead of their titles. However, we consider working with numeric IDs very counter-intuitive.

Remove from VirtueMart groups Likewise, this is the list of the VirtueMart groups to remove a user from when his subscription to the specified level is no longer active. Do note that you can specify a different set of groups to remove the user from than the ones you are adding him to. For example:

```
LEVEL1=Group 1,Group 2
LEVEL2=Group 2
LEVEL3=Group 1
```

In this example, together with the rules displayed in "Add to VirtueMart groups", when a subscription to the LEVEL1 level expires, the user will still belong to VirtueMart Group 3, as it's not removed from his account; only Group 1 and Group 2 are removed. Likewise, if the user has an expired LEVEL1 subscription and an active LEVEL2 subscription he'll now belong to Group 2 and Group 3: Group 3 because it's not removed when his subscription expires and Group 2 because his active LEVEL2 subscription suggests that he should be granted Group 2 access despite his expired LEVEL1 subscription.

If you find this hard to understand, join the club. ACLs are not meant to be easy to setup or understand. It's a power feature and, as such, poses a great difficulty to even the most seasoned web site integrators. Take your time and experiment on a dev copy of your site before going live.

2.15. VirtueMart 2 Integration

Displayed in the Plugins Manager as: Akeeba Subscriptions - VirtueMart 2 Integration

Important

This plugin only works with VirtueMart 2.x. It will NOT work with VirtueMart 1.x.

This integration plugin allows you to add and remove users from VirtueMart 2.0 shopper groups based on their subscription status. This allows you to set special discounts to subscribers (e.g. a discount club) or other VirtueMart tricks which are based on the shopper groups a user belongs to.

Please note that this is a smart integration, exactly as described in the JUGA plugin earlier in this documentation; instead of adding users when a subscription goes active and removing them when a subscription goes inactive, Akeeba Subscriptions plugins take a look at all of the user's active and inactive subscriptions and decide which groups the user should be added to or removed from.

The plugin has these options:

Add to VirtueMart groups This is a list for assigning subscription levels to VirtueMart 2.x groups. When a user has an active subscription to the specified level, he'll be added to the VirtueMart 2.x group mentioned on the right hand of the equal sign. Each assignment is done like this:

```
LEVEL1=Group 1,Group 2,Group 3
```

Where LEVEL1 is the Title of the subscription level and Group 1, Group 2 and Group 3 are the names of the VirtueMart 2.x groups you want to add a user to when he subscribes. If you want to assign multiple subscription levels to multiple groups, you have to do something like that (separate multiple lines with a single press of the ENTER key on your keyboard):

```
LEVEL1=Group 1,Group 2,Group 3
LEVEL2=Group 2
```

```
LEVEL3=Group 4,Group 1
```

You can also use the numeric IDs of subscription levels or VirtueMart groups instead of their titles. However, we consider working with numeric IDs very counter-intuitive.

Remove from VirtueMart groups

Likewise, this is the list of the VirtueMart 2.x groups to remove a user from when his subscription to the specified level is no longer active. Do note that you can specify a different set of groups to remove the user from than the ones you are adding him to. For example:

```
LEVEL1=Group 1,Group 2
LEVEL2=Group 2
LEVEL3=Group 1
```

In this example, together with the rules displayed in "Add to VirtueMart groups", when a subscription to the LEVEL1 level expires, the user will still belong to VirtueMart Group 3, as it's not removed from his account; only Group 1 and Group 2 are removed. Likewise, if the user has an expired LEVEL1 subscription and an active LEVEL2 subscription he'll now belong to Group 2 and Group 3: Group 3 because it's not removed when his subscription expires and Group 2 because his active LEVEL2 subscription suggests that he should be granted Group 2 access despite his expired LEVEL1 subscription.

If you find this hard to understand, join the club. ACLs are not meant to be easy to setup or understand. It's a power feature and, as such, poses a great difficulty to even the most seasoned web site integrators. Take your time and experiment on a dev copy of your site before going live.

2.16. RedShop Integration

Displayed in the Plugins Manager as: Akeeba Subscriptions - RedShop Integration

This integration plugin allows you to add and remove users from RedShop shopper groups based on their subscription status. This allows you to set special discounts to subscribers (e.g. a discount club) or other RedShop tricks which are based on the shopper groups a user belongs to.

Please note that this is a smart integration, exactly as described in the JUGA plugin earlier in this documentation; instead of adding users when a subscription goes active and removing them when a subscription goes inactive, Akeeba Subscriptions plugins take a look at all of the user's active and inactive subscriptions and decide which groups the user should be added to or removed from.

Important

RedShop allows a user to belong to exactly one shopper group.

The plugin has these options:

Add to RedShop groups

This is a list for assigning subscription levels to RedShop groups. When a user has an active subscription to the specified level, he'll be added to the RedShop group mentioned on the right hand of the equal sign. Each assignment is done like this:

```
LEVEL1=Group 1
LEVEL2=Group 2
LEVEL3=Group 3
```

Where LEVEL1 is the Title of the subscription level and Group 1 is the name of the RedShop group you want to add a user to when he subscribes.

You can also use the numeric IDs of subscription levels or RedShop groups instead of their titles. However, we consider working with numeric IDs very counter-intuitive.

Remove from RedShop groups Likewise, this is the list of the RedShop groups to remove a user from when his subscription to the specified level is no longer active. Do note that you can specify a different set of groups to remove the user from than the ones you are adding him to. For example:

```
LEVEL1=Group 1  
LEVEL2=Group 2  
LEVEL3=Group 1
```

2.17. Sample Fields

Displayed in the Plugins Manager as: Akeeba Subscriptions - Sample Fields

Since Akeeba Subscriptions 1.0.0 it is possible to have custom fields in the subscription page by means of plugins. These fields are stored as JSON inside each user's params field in the #__akeebasubs_users table. Developers can easily create plugins to add as many fields as they want. The "Sample Fields" plugin is a demonstration of this feature which also acts as a self-documenting tutorial for developers. It adds two custom fields (Gender and Age Group).

2.18. Automatic Country and City fill

Displayed in the Plugins Manager as: Akeeba Subscriptions - Automatic Country and City fill

Since Akeeba Subscriptions 1.0.0 it is possible to have plugins which can override user information (e.g. name, address) and can also be notified when user's information is being saved. This allows for very advanced integrations. One possibility is to automatically fetch such information from external services and component, e.g. using a geolocation service or fetching user information from Community Builder, JomSocial, etc. The other possibility is to provide a "reverse integration", e.g. update Community Builder or JomSocial fields based on the information the user entered in the subscription page, essentially making the user believe that all of his information is managed in a single place.

This particular plugin acts as a demonstration of this plugin system. It will query the user's IP address against the free hostip.info service and populate the country and city fields, unless the user has overridden them, i.e. he has bought another subscription in the past.

2.19. Custom SQL scripts

Displayed in the Plugins Manager as: Akeeba Subscriptions - Custom SQL scripts

This plugin can run arbitrary SQL commands when a subscription becomes active or inactive. Please note that this plugin is very naive. For starters, it will run the SQL commands once for every active and inactive subscriptions it encounters. If you have two disabled subscriptions and five enabled subscriptions on a specific level it will run the deactivation SQL twice and the activation SQL five times. Moreover, the plugin fires: when a new subscription is created; when the subscription status changes for any reason (payment status changes, manual disable/enable, automatic enable/disable due to Valid From/To dates being reached) and when you click on the Run Integration button.

In the plugin's parameters you can define the SQL commands to run. In all cases, you can do something like this:

```
LEVEL1=INSERT INTO #__foobar (`user_id`,`something`) VALUES( '[USERID]', 'myvalue' );
```

The leftmost part is the subscription level for which the SQL command will run. The rightmost part is the list of SQL commands, separated by semicolons. The `[USERID]` variable will be replaced with the user's numeric ID for the user whose subscription Akeeba Subscriptions is currently processing.

3. Other plugins

Akeeba Subscriptions comes with more standard Joomla! plugins which accomplish a variety of tasks, from email communications to content filtering.

3.1. Subscription expiration control

Displayed in the Plugins Manager as: **System - Akeeba Subscriptions Expiration Control**

Akeeba Subscriptions will automatically expire your users' subscriptions when they visit a page with an Akeeba Subscriptions module on your site after their subscription expiration date. However, if you want subscription expiration to be performed automatically, you need to publish this plugin. It will fire once per day, expiring subscriptions past their expiration date. We strongly recommend publishing this plugin at all times.

3.2. Subscription emails

Displayed in the Plugins Manager as: Akeeba Subscriptions - Emails

When this plugin is published, an email will be sent out to the user when any change in the status of the subscription has taken place. For a complete list of the events on which an email is sent you can take a look at the language file, e.g. `administrator/language/en-GB/en-GB.plg_akeebasubs_subscriptionemails.ini`. Please note that if you are running a multilingual site, the site's active language at the time the subscription was registered is saved as the user's default language. As long as translation files for that language exist for our email plugin, then all emails the user receives will be in that language (the user's default language).

If you want to customise the email messages, you can easily do so! First, you will need to find your language's language code. In order to do that, just take a look at `administrator/language`. You will see a lot of directories in there. `en-GB` is the English (United Kingdom) language and, depending on which languages you have installed on your site, you will see more directories e.g. `es-ES` for Spanish (Spain), `de-DE` for German (Germany), `el-GR` for Greek (Greece) and so on. Note that down. Whenever you see `xx-XX` below, replace it with your own language's language code.

Go inside the `administrator/language/xx-XX` directory. Find the `xx-XX.plg_akeebasubs_subscriptionemails.ini` file and copy it to `xx-XX.plg_akeebasubs_subscriptionemails.override.ini`. Now, edit any of the subscription email texts. The Emails plugin will use your override file instead of the default when sending out emails!

Note

This plugin uses Joomla!'s mail system. You have to ensure that your site can send out emails (e.g. use the Mass Mail feature to send a test email to Super Administrators) before activating this plugin.

We strongly recommend publishing this plugin at all times.

3.3. Subscription expiration notification

Displayed in the Plugins Manager as: System - Akeeba Subscriptions Expiration Notification

This plugin will notify a user before his subscription's expiration, based on the notification settings of each subscription level.

Note

This plugin uses Joomla!'s mail system. You have to ensure that your site can send out emails (e.g. use the Mass Mail feature to send a test email to Super Administrators) before activating this plugin.

We strongly recommend publishing this plugin at all times.

If you want to customise the email messages, you can easily do so! First, you will need to find your language's language code. In order to do that, just take a look at `administrator/language`. You will see a lot of directories in there. `en-GB` is the English (United Kingdom) language and, depending on which languages you have installed on your site, you will

see more directories e.g. es-ES for Spanish (Spain), de-DE for German (Germany), el-GR for Greek (Greece) and so on. Note that down. Whenever you see xx-XX below, replace it with your own language's language code.

Go inside the administrator/language/xx-XX directory. Find the xx-XX.plg_system_asexpirationnotify.ini file and copy it to xx-XX.plg_system_asexpirationnotify.override.ini Now, edit any of the subscription email texts. The Emails plugin will use your override file instead of the default when sending out emails!

3.4. Content restriction

Displayed in the Plugins Manager as: Content - Akeeba Subscriptions Restricted

This content plugin allows you to show parts of your content only to specific subscribers. This is a very powerful feature that allows you to alter the displayed content based on the user's subscription status.

Note

It can not "hide" entire articles, e.g. not show them at all to non-subscribers. At best, only the title will be shown. If you want to completely hide articles and block unsubscribed user access to entire categories or sections in Joomla! 1.5 you will have to use an ACL solution, like Dioscouri's JUGA, NoixACL or similar.

If you have Joomla! 1.6 or later you can use the bundled Joomla! 1.6 User Group integration plugin to automatically assign subscribers to Joomla!'s groups and then use Joomla!'s ACL feature to fine-tune which users have access to which articles.

Note

The following documentation adds a space between the curly brackets ({ and }). This is done in order for Joomla! not to process the documentation text as plugin instructions. When you use the plugin, please DO NOT add a space between the curly braces and its contents. Thank you!

Its syntax is:

```
{ akeebasubs expression }Your content{ /akeebasubs }
```

This means that "Your content" will only be displayed to users whose subscriptions satisfy the *expression*. In the simplest form, expression is just the name of a subscription level, e.g.

```
{ akeebasubs LEVEL1 }Your content{ /akeebasubs }
```

If you want to present the content to someone who holds an active subscription to both LEVEL1 and LEVEL2, you can use the && (logic and) operator:

```
{ akeebasubs LEVEL1 && LEVEL2 }Your content{ /akeebasubs }
```

Likewise, you can use the || (logic or) operator to present the content to anyone who has a subscription to any of LEVEL3 or LEVEL4 levels:

```
{ akeebasubs LEVEL3 || LEVEL4 }Your content{ /akeebasubs }
```

Important note: The logical OR consists of two "pipe" symbols. These are not the same as capital I! On a Mac's keyboard, the pipe symbol is present next to the ENTER key, accessible by pressing SHIFT-|. On most Windows/Linux machines, this key is usually somewhere near the ENTER key and is most likely accessible by pressing SHIFT-| as well.

You can also negate the logic. For example, you can present the content to anyone having a subscription to neither LEVEL1 nor LEVEL3 by using the ! (logical not) operator:

```
{ akeebasubs !LEVEL1 && !LEVEL3 }Your content{ /akeebasubs }
```

If you specify multiple operators, the precedence is NOT, AND, OR. This means that the exclamation mark is processed first, the ampersands second and the bars last. Example:

```
{ akeebasubs !LEVEL1 || LEVEL2 && !LEVEL3 }Your content{ /akeebasubs }
```

This means that the content will be presented to users who do not have a subscription to LEVEL1. Also, if they do have a LEVEL1 subscription and they are also LEVEL2 (but not LEVEL3) subscribers the content will also be shown to them.

Tip

Since Akeeba Subscriptions 1.0 Stable you can also use a pseudo-level marked by a single star (*) which matches all active subscriptions. For example, to show something to all subscribers use:

```
{ akeebasubs * }Content to show to all subscribers{ /akeebasubs }
```

Additionally, you can use the "not" modifier (exclamation mark) to show content only to non-subscribers:

```
{ akeebasubs !* }Content to show to non-subscribers{ /akeebasubs }
```

You can use this trick to easily show different messages to subscribers and non-subscribers, no matter if you have decided on the final list or naming of your subscription levels.

3.5. The Akeeba Subscriptions Link (aslink) plugin

Displayed in the Plugins Manager as: Content - Akeeba Subscriptions Link

The purpose of the aslink plugin is to produce URLs to subscription pages if you give it the name of the subscription level, its URL or its numeric ID.

Note

The following documentation adds a space between the curly brackets ({ and }). This is done in order for Joomla! not to process the documentation text as plugin instructions. When you use the plugin, please DO NOT add a space between the curly braces and its contents. Thank you!

The aslink plugin only produces a URL, not a link. In order to create a link, select the text you want to link, click on your WYSIWYG editor's link button and when it asks you for a URL, enter the plugin code, as shown below.

Syntax:

```
{ aslink MYLEVEL }
```

Returns the URL to a subscription level called "MYLEVEL", e.g. <http://localhost/component/akeebasubs/new/mylevel?layout=default>.

```
{ aslink mylevel }
```

Returns the URL to a subscription level whose slug is "mylevel", e.g. <http://localhost/component/akeebasubs/new/mylevel?layout=default>.

```
{ aslink 2 }
```

Returns the URL to a subscription level whose numeric ID is 2, e.g. <http://localhost/component/akeebasubs/new/mylevel?layout=default>.

In order to find the numeric ID of a level, please go to your site's back-end, Components, Akeeba Subscriptions, Subscription Levels and click on the name of the subscription level you want. Note the URL in the browser. It's something like `http://localhost/administrator/index.php?option=com_akeebasubs&view=level&id=2`. The number after `&id=` is the numeric ID of the level. In this example, the numeric ID is 2.

Chapter 4. Akeeba Subscriptions' modules

Akeeba Subscriptions comes with a variety of modules to display customized lists of subscription levels or a list of a user's subscriptions. Make sure you visit your Module Manager page and have a look at them. You can create interesting effects, like a customized and categorized subscriptions list page, out of those modules.

1. List of active subscriptions

Displayed in the Modules Manager as: Akeeba Subscriptions - List of active subscriptions

This is extremely simple plugin will merely list the current user's active subscriptions as an unordered list. It is meant as an example of how you can create custom modules for Akeeba Subscriptions.

2. List subscription levels

Displayed in the Modules Manager as: Akeeba Subscriptions - List subscription levels

This extremely powerful module allows you to present a list of all or specific subscription levels, using either layout, in any module position on your site. Combined with Joomla!'s { loadposition } plugin you can use this module to assemble the perfect presentation page of your subscriptions, grouped any way you please.

The module has only two options:

- | | |
|---------------------|---|
| Subscription Levels | Choose which subscription level(s) to display in the module. This is a multiple-selection list. You can use CTRL-click (Windows, Linux) or CMD-click (Mac OS X) to select multiple items on the list. |
| Layout | Choose which layout you want to use to display your subscription levels to the users. You can choose between the Default or the Awesome layout, the same layouts you can choose in menu items. |

Chapter 5. Developers' information

Akeeba Subscriptions was designed to be very extensible right from its inception. Everything can be accomplished using plugins which can be installed through the familiar Joomla! extensions installer.

The plugins can belong to one of two groups:

- akeebasubs** This is the plugin group where all integrations live. Plugins in this group can implement a plethora of methods which allows them to be notified when a subscription's status changes, when the user information is read/written and much more.
- akpayment** This plugin group is used by Akeeba Subscriptions payment plugins. Plugins in this group implement special methods which allows them to be notified of the user's intention to pay for a subscription, as well as process the payment notification by the payment processor and, in turn, let the component know.

The rest of this chapter documents the various plugin events in each group. References to existing plugins are made so that you can understand how things really work.

1. The "akeebasubs" plugin events

1.1. onAKSubscriptionChange

Prototype: `public function onAKSubscriptionChange($row, $info)`

Synopsis: This event is called whenever a subscription is created or modified.

Sample plugin: `plugins/akeebasubs/subscriptionemails.php`

The `$row` variable contains a `JTable` descendant which gives you access to all of the information in the subscription record (`#__akeebasubs_subscriptions` table's fields). The `$info` variable contains a hash array with useful information, including a copy of table before and after the modification, the modified rows and the kind of modification (new for new record or modified for a modified existing record) that took place.

You can not change the information in `$row` or pass data back to the component in any way. In fact, this event is called immediately after all information has been written to the database.

1.2. onAKUserRefresh

Prototype: `public function onAKUserRefresh($user_id)`

Synopsis: This event is called when the user clicks on the "Refresh Integrations" button in the back-end of the Akeeba Subscriptions component. It is called once per each subscriber. The `$user_id` variable contains the numeric Joomla! user ID of the subscriber.

Sample plugin: `plugins/akeebasubs/juga.php`

You might consider it odd that the event is called per user instead of per subscription. However, the integrations in Akeeba Subscriptions are "smart". This means that instead of triggering functionality based on the enabled status of an individual subscription they work based on the enabled status of all subscriptions. This allows you to have rules like "if the user has SUB1 and SUB2, do this" or "if the user has SUB1 and had a SUB2 but it has now expired, do that". Looking at the sample plugin's code you can understand how the iteration of the user's subscriptions is made and how a plugin can determine which subscription levels the user belongs to.

1.3. onSubscriptionFormRender

Prototype: `public function onSubscriptionFormRender($userparams, $cache)`

Synopsis: This event is called when the subscription form (front-end, view=level) is displayed. It returns an array of extra fields which will be rendered in the form, right below the email field and before the address fields.

Sample plugin: `plugins/akeebasubs/samplefields.php`

The input of this event's method is two variables:

\$userparams This is a copy of the user's record, as stored in the `#__akeebasubs_users` table, in object format. You can access the data in the user record as, for example, `$userparams->address1` in order to get the Address 1 field's contents.

The most important -for you- part of this object is `$userparams->params` which stores all the extra fields' values. For example, if you have defined a field named `foobar`, you can get its stored value as `$userparams->params->foobar`.

Important

This object contains the data stored in the database. When the user modified this data in the interface and submits the form (but the form is invalid) the information in this object IS NOT UPDATED. In this case, use the `$cache` variable.

\$cache When the user submits the subscription forms but it is invalid (errors don't allow it to pass the validation), Akeeba Subscriptions stores the user's modified data in the `$cache` array. The information in this array may be different than the information in `$userparams` array.

Getting the values for your custom fields

To cut a long story short, if you have a field named `foobar`, here is how to get its value:

```
if(array_key_exists('foobar', $cache['custom'])) {
    $current = $cache['custom']['foobar'];
} else {
    $current = property_exists($userparams->params, 'foobar') ? $userparams->params->foobar
}
```

Returning custom field definitions

The return of this event is an array which consists of one or more hash (key/value pair) arrays. Each of the hash arrays defines exactly one extra field and has the following keys:

| | |
|--------------------------|---|
| <code>id</code> | String. Required. The field's HTML id. |
| <code>label</code> | String. Required. The label which appears to the left of the field. |
| <code>elementHTML</code> | String. Required. The HTML of your field. |

Important

The name value of your field **must** be `custom[id]`, where `id` is the HTML id you defined in the `id` key above. Also, the HTML id **must** be defined and **must** be equal to

the HTML id you defined in the id key above. If you don't do that, your extra field might never be saved.

| | |
|--------------|--|
| invalidLabel | String. Optional. This is the label which appears in green on the right of the field when the field is valid. The HTML element containing this string is named <code>id_valid</code> , where <code>id</code> is the HTML id you defined in the id key above. |
| validLabel | String. Optional. This is the label which appears in red on the right of the field when the field is invalid. The HTML element containing this string is named <code>id_invalid</code> , where <code>id</code> is the HTML id you defined in the id key above. |
| isValid | Boolean. Optional. If you defined the <code>invalidLabel</code> and/or <code>validLabel</code> fields, pass a boolean here indicating if the field is currently valid so that the correct label is shown when the page loads. |

Javascript validation

Akeeba Subscriptions does not make any attempt to validate the custom fields and update the display status of the valid/invalid labels. It is your responsibility. In this event, you can call `JFactory::getDocument()->addScriptDeclaration($scriptData)` where `$scriptData` is the Javascript validation code.

In order to make it easier for you, Akeeba Subscriptions offers an architecture similar to an observer pattern in the Javascript code. You can have Javascript functions called when Akeeba Subscriptions' Javascript code is fetching the contents of the fields before sending a data validation request through AJAX and another function which is called when the validation results are being processed. You can add the former to the stack using `addToValidationFetchQueue()` and the latter using `addToValidationQueue()`.

Please look at the very well documented sample plugin for more information regarding all of the above.

1.4. onValidate

Prototype: `public function onValidate($data)`

Synopsis: This event is called when Akeeba Subscriptions is validating the subscription form.

Sample plugin: `plugins/akeebasubs/samplefields.php`

The `$data` parameter is an object with all the data keys. You are interested in `$data->custom` which contains the values of the custom fields. You can get the value of a custom field named "foobar" using `$data->custom['data']`.

The return value is a hash array with the following keys:

`custom_validation` A hash array of booleans, showing the validation status of each field.

`isValid` Set to false if any of the validation errors above means that the entire subscription form should be deemed invalid, otherwise set it to true to not cause the entire form to become invalid. The latter is useful when you have an optional field whose validation status doesn't play a role in the user's ability to complete the subscription transaction.

1.5. onAKUserGetData

Prototype: `public function onAKUserGetData($userData)`

Synopsis: This plugin is called when Akeeba Subscriptions is fetching the user data the first time in the current session that the user is visiting the Subscription (view=level) page

Sample plugin: `plugins/akeebasubs/autocity.php`

This method is called whenever a user starts a new subscription and Akeeba Subscriptions wants to fetch user data. You can use it to fetch user information from additional sources and return them in an array. The values in the array will replace the values stored in the user's profile.

The `$userData` variable contains an object with already fetched user information from the `#__akeebasubs_users` table.

Returns a key/value array with user information overrides.

1.6. onAKUserSaveData

Prototype: `public function onAKUserSaveData($row)`

Synopsis: This plugin is called when Akeeba Subscriptions is saving user's data to the database.

Sample plugin: `plugins/akeebasubs/autocity.php`

This method is called whenever Akeeba Subscriptions is updating the user record with new information, either during sign-up or when you manually update this information in the back-end.

The `$row` variable contains a `KRowDatabaseDefault` object with the information to be stored in the `#__akeebasubs_users` table.

2. The "akpayment" plugin events

2.1. onAKPaymentGetIdentity

Prototype: `public function onAKPaymentGetIdentity()`

Synopsis: Returns the internal name and the human readable description of this payment plugin.

Sample plugin: `plugins/akpayment/paypal.php`

Returns a hash array:

`name` The internal payment processor plugin name. If you have `plg_akpayment_foobar` then this must be `foobar`.
`title` The name of the payment processor as will appear on the subscription page, e.g. "Foobar Payments", "Credit Card", etc.

2.2. onAKPaymentNew

Prototype: `public function onAKPaymentNew($paymentmethod, $user, $level, $subscription)`

Synopsis: This method is called whenever the user asks for a payment to be made for his new subscription.

Sample plugin: `plugins/akpayment/paypal.php`

2.3. onAKPaymentCallback

Prototype: `public function onAKPaymentCallback($paymentmethod, $data)`

Synopsis: This method is called whenever the payment processor posts back to Akeeba Subscriptions to notify us of a payment having been processed.

Sample plugin: `plugins/akpayment/paypal.php`