JSmallfib User Guide

PRO version (commercial)

for Joomla! versions 1.6/1.7/2.5

Version 1.3.2, April 2012

About JSmallfib

JSmallfib is a simple but effective file browser/manager embedded in a normal Joomla! article page. Depending on access levels, users can read or upload files, create folder trees, etc. The main repository can be set either within the Joomla! directory, or, for greater security, outside the web root. Actions such as file uploads/downloads, and file/folder renaming, removing can optionally be logged.

This version of the plugin is native in Joomla! 1.6/1.7/2.5.

JSmallfib is ideal if you require simple light-weight file browsing/uploading functionality (to store and share documentation, for example, or to easily manage your website files directly from the front-end). However, you can easily configure the plugin to adapt it to more complex configurations, such as when using the same article page to display different repositories to different users or groups of users. This User Guide will describe how to properly configure JSmallfib to satisfy different requirements.

All the files used and managed by JSmallfib are stored in the file system as normal files and folders, without the support of the Joomla! database. This way files and folders are easily accessed separately via FTP, for example, without the need for reconfiguring the software.

The Quick Start section of this guide shows how easy it is to start working with JSmallfib. However, this software has grown to become fairly flexible, with many options available for customization. The purpose of this document is to describe in detail all the functionality of JSmallfib.

Originally released in June 2009, JSmallfib has been downloaded tens of thousands of times from more than 170 countries all over the world. The plugin is currently translated in Danish, Dutch, French, German, Italian, Portuguese, Spanish and Swedish in addition to English.

JSmallfib benefits greatly from the contributions of many Joomla! users. Some volunteer code, some write translations, many suggest improvements to the program, or point out issues to be solved. It is not possible to thank all of them here, but some are mentioned in the Change Log and the Wish List sections, along with their contribution to this project.

What's new in the commercial version of JSmallfib

The main additional features of the commercial version over the standard one are listed below:

- Flash upload module based on SWFUpload. This allows the plugin to perform multiple file uploads, each with its own progress bar and the ability to stop the upload in progress. The new module also allows administrators to filter file uploads by extension name and/or by file size.

- More logging options: these include the ability to store action logs in JSON format, to specify the log directory, to indicate whether to store all logs into a single file or use a separate file per action (default in the non-commercial version), and to send log emails in HTML format.

- 'Trusted-authors' feature, so that only articles written (and last modified) by specific users will display a repository if a {jsmallfib} command is contained in the article. To this end, a backend parameter is available to list the user IDs or group IDs of these 'trusted' users.

- More styling options (such as box shadows, round corners), and features an additional icon set, with the ability to modify the size of the file icons (up to 128px). The new icon set was partly based on the Nuvola set at www.stdicon.com, extended with proprietary graphics to make the overall look of the repository more modern. Over 100 file extensions are recognized by the new set (although some similar extensions will display with the same icon, such as for image files, videos, 3D models, word-compatible documents, and the like).

- Ability to hide the credits icon displayed in the front end (located in the bottom right side of the repository). This is an image displaying the wording 'jsmallfib'; mouse hovering it will display the name of the plugin, of the developer (with copyright wording) and the installed version of the software.

Index

About JSmallfib.................... 2

What's new in the commercial version of JSmallfib.................... 3

Download and installation.................... 7

- System requirements.................... 7
- Installation instructions.................... 7

Quick start.................... 9

Backend configuration.................... 10

Tutorial.................... 11

- Using a specific repository (inside or outside the web root).................... 11
- Specifying permissions to access the repository.................... 12
- Displaying a separate repository for each individual user.................... 14
- Displaying a separate repository for each group of users 15
- Hiding certain files and/or folders.................... 17
- Linking files directly (without showing the options popup box).................... 18
- Hiding the navigation links.................... 18
- Enabling/disabling or resizing image thumbnails.................... 18
- Changing formats or hiding file size and/or date.................... 19
- Modifying the default sorting options or hide/display the file filter box.................... 19
- Allowing unzip of compressed files.................... 20
- Hiding or displaying the info link.................... 21
- Configuring the Flash upload module.................... 21
- Modifying the overall look and feel of the display.................... 22
- Displaying a description title for a specific repository.................... 23
- Disabling/enabling the option to archive files.................... 23
- Managing the actions logs (including email options).................... 24
- Locating repositories independently of the 'Default JSmallfib Path'.................... 26
- Defining 'trusted authors' for articles containing repositories.................... 27
- Modifying the default CHMOD settings (advanced users).................... 27
- Hiding the credits icon.................... 27

Changing the icons used by JSmallfib........ 28

Modifying a string of text or creating a new translation for JSmallfib........28

Command syntax........30

Repository options........ 30

Access levels options........32

Other options........ 33

Frequently asked questions........35

Does JSmallfib have a file size limit when uploading? How can I increase the limit?........35

JSmallfib displays the repository correctly by I cannot navigate in it. 36

I have installed JSmallfib, but I get 'This line is replaced by a file repository ...' 36

JSmallfib does not properly display special characters, such as accented letters 37

JSmallfib does not recognise the starting folder........ 37

I have placed two JSmallfib commands in the same article, but only one works. Is it normal?. .37

JSmallfib crashes on pages with image files listings. What's the problem here?........37

Support........38

Wish list........ 39

Change log........ 40

29 March 2012 - Version 1.3.2........ 40

17 March 2012 - Version 1.3/Version 1.3.1........ 40

24 March 2011 - Version 1.2.1........ 40

25 January 2011 - Version 1.2........41

17 August 2010 - Version 1.1........41

14 June 2010 - Version 1.0.37........42

18 May 2010 - Version 1.0.36d........42

07 May 2010 - Version 1.0.36........42

05 January 2010 - Version 1.0.35........43

19 October 2009 - Version 1.0.34........43

15 October 2009 - Version 1.0.33b........43

11 September 2009 - Version 1.0.32........ 44

04 September 2009 - Version 1.0.31........ 44

16 July 2009 - Version 1.0.30........44

02 July 2009 - Version 1.0.29........45

24 June 2009 - Version 1.0.28....45
17 June 2009 - Version 1.0.27....45
15 June 2009 - Version 1.0.26....45
Copyright & Licence.... 46

Download and installation

You can purchase the latest version of JSmallfib from Smallerik. You will receive a link from where you'll be able to download the most recent package. Links to purchased software remain valid for a certain amount of time (check with the Smallerik site for exact details), during which you'll benefit from any update to the software.

If upgrading from a previous version, JSmallfib installs over an existing package. This works also if you are upgrading from the non commercial version (see the following section on Installation for more details), saving you from having to manually reconfigure the plugin, because all existing configuration settings will be preserved. Check out the system requirements, and please refer to the installation instructions if you need more detailed information on this procedure.

System requirements

JSmallfib will run on any webserver supporting PHP. However, it is suggested to use at least version 5.2 of PHP.

To enable thumbnail functionality for recognised image files, you need to have PHP compiled against the GD libraries, with version at least 1.8.

Installation instructions

For a new installation, once downloaded to your PC the latest package (zip file) from Smallerik , you should do the following:

1. Go to the Joomla! backend application (the admin site)
2. Click on Extensions → Extension Manager
3. Choose the 'Upload Package File' method ans select the zip file package from your PC.
4. Once the upload is executed successfully, go to Extensions → Plug-in Manager

You'll now see, in the plugin list, a line representing the newly installed JSmallfibPro – Simple File Browser plugin. This will have a red cross in the status column, indicating that the plugin is not enabled (which is the normal Joomla! default for newly installed plugins). Enable it by clicking on the cross, which should turn to a green tick sign.

Click on the plugin name in the list to display the configuration page, where you'll see all available paramaters. Hovering with your mouse onto each parameter's name will display information on how to use that particular parameter.

For upgrading an existing installation: you do not need to uninstall the existing package before installing the new one. You can just follow the instructions as per a new installation, keeping in mind that after installation the existing parameters will be kept unchanged (including the 'enabled' parameter, so the plugin will remain enabled if it was before the upgrade).

If you have been using the free version of JSmallfib, the commercial version will install as an upgrade on top of the former. This way you won't lose any of the current settings, although you'll benefit from the new functionality in terms of the new backend options that will be installed.

Also, the command syntax used in the article pages by the free version is fully compatible with the commercial version, so you won't need to alter the commands inside the existing articles either.

To uninstall the plugin:

1. Go to the Joomla! backend application (the admin site).
2. Click on Extensions → Extension Manager
3. Click on the Manage tab link in the line just below he page title.

You'll now see a list of currently installed plugins. You should find a line for JSmallfibPro – Simple File Browser (you may need to look for it on one of the following pages, if more than one page is listed). On this line you can see the version number of the currently installed package. Click on the tick box at the left of the plugin name, then click on the Uninstall link on the top right corner of the page (you may need to scroll to the top of the page to see it).

Quick start

Once the plugin is installed **and enabled** (please refer to the installation instructions if required), all you need to do to see it in action is create a new article (or edit an existing one) and place in it the special command (do not leave spaces between the opening brace and the jsmallfib keyword):

```
{jsmallfib}
```

Save the article, which, in the front end, will now display the contents of the top level repository folder specified by the Default JSmallfib Path parameter in the plugin configuration page within the administrator site. If you have left the default value after installation, all files and folders of the repository will be stored inside a folder named jsmallfib_top located under your Joomla! installation. JSmallfib will create this folder when first accessing the repository.

Please **notice that the repository will only be displayed when the article is viewed directly** and not if this is displayed in a list of articles (such as when using a blog mode or in the featured article list of the home page).

Backend configuration

The next step after the quick start described above is to explore the configuration and customization options in the plugin configuration area, where all backend parameters are displayed.

To access this page, login to your Joomla! Administrator site, then go to Extensions → Plugin Manager. From the list of currently installed plugins, find the line referring to JSmallfibPro – Simple File Browser (keep it mind that the list could have multiple pages) and click on its name to display the configuration page.

On the left side of this page you'll see some standard Joomla! parameters, including the 'enabled' switch. On the right side you have all the plugin-specific backend parameters. These are grouped into a series or tabs. Opening each tab will reveal the relevant parameters, and hovering your mouse on each parameter's name will pop up specific information about that parameter, such as its usage and whether the parameter can be overriden from the front-end by using a command option inside the individual article's JSmallfib command.

For example, the parameter storing the default permission level for registered users can be overridden by using the reg:x option in the JSmallfib command contained inside an article (we'll discuss about command options in the following sections; in this case x is an integer number from 0 to 6, representing a permission level). Overriding backend parameters allows the administrator to specify different settings for repositories displayed from different articles. You'll see how to do that in the following chapters.

Tutorial

The quick start section above shows how easy it is to start using JSmallfib. However, this plugin has a lot more features. As discussed in the Backend configuration chapter, you can get hints on how to use individual backend parameters from the backend configuration page (in the admin site, go to **Extensions → Plugin Manager**, then click on the **JSmallfibPro – Simple File Browser** line). In that page, hovering your mouse on each parameter's name will display more information about it.

The following sections will describe some common and less common tasks you may want to know how to perform with JSmallfib.

Using a specific repository (inside or outside the web root)

When you create a Joomla! article containing a JSmallfib command such as

```
{jsmallfib}
```

and view the article directly (not in a list of articles such as in the featured-article list of the home page or in a category list in blog mode, for example), and JSmallfib is correctly installed and enabled, the browser will display a file repository located inside the folder specified by the backend parameter **Default JSmallfib Path** in the plugin configuration page.

With the **Type of path** backend parameter you can specify if the **Default JSmallfib Path** has to be placed inside the web root folder (for a relative path), or has to be taken as an absolute path, which can for example be located outside the web root. The **Default JSmallfib Path** is automatically created, if it doesn't exist already, the first time an article is viewed.

If you are going to have more articles displaying different repositories, these will normally be located inside subfolders within the **Default JSmallfib Path**. To this end, each command will need to specify the intended repository. This is achieved with a command option inside square brackets, such as in

```
{jsmallfib [Documents]}
```

In this case a folder named Documents inside the Default JSmallfib Path will be created and displayed as a repository in the article containing this command. Users accessing this repository will not be able to access any area outside the Documents folder when navigating the repository created in this article.

If you create another article and place a JSmallfib command such as

```
{jsmallfib [Images]}
```

then a new folder named Images will be created in parallel to the Documents folder, both inside the Default JSmallfib Path. Users browsing this last repository will not be allowed to see anything outside the Images folder.

JSmallfib allows you to place repositories independently of the Default JSmallfib Path. This could be useful, for example, if you require to display two repositories that do not share a common folder. For details on ho to achieve this, please refer to the section named 'Locating repositories independently of the Default JSmallfib Path', where you'll see how to use the relpath() and abspath() command options.

Specifying permissions to access the repository

JSmallfib uses a 'layered' permission system. This means that sets of functionality (view, rename, upload, download, override, delete, etc) are grouped together in 'levels' of access. There are currently 7 levels of access, namely:

0. May not access repository at all
1. Can only navigate the repository (without downloading files)
2. Can navigate the repository and download files
3. Can create folders and upload files
4. Can delete files, but not folders
5. Can delete files and empty folders
6. Can additionally view logs

Normally JSmallfib will apply, to each repository, the default values set in the backend configuration page. In particular, separate permissions will be applied to visitors (non registered users) and registered users. These default values are respectively defined in the backend parameters found under the Permissions Settings → Default Access Levels section.

However, you may want to override the default permissions for individual repositories. This can be achieved including special options inside the JSmallfib command. These options come in the form of pairs of values separated by a colon. The first value is normally a user ID, while the second is one of the access levels shown above. You should leave no space to the left and right of the colon sign.

Permissions for indivual users

Overriding default permissions for individual users is simply done by adding a permission option in the command in the form user_ID:permission_level. For example, the command

```
{jsmallfib [Documents] 42:6 12:1 24:3}
```

will allow user with ID 42 (normally the Administrator) full access (level 6) to this repository, while users with IDs 12 and 24 will have respectively access levels 1 and 3. Notice that permission options can be located anywhere inside the JSmallfib command, and may be separated by spaces, commas, or even surrounded (each one separately) by square brackets. So the following commands are equivalent to the one above:

```
{jsmallfib [Documents] 42:6,12:1, 24:3}
{jsmallfib [Documents] [42:6] [12:1] [24:3]}
```

Remember that **the square brackets around the repository name are mandatory**, while the ones around the permission options are optional.

If you wish to set a permission option for all non registered users, you can use 0 (zero) in place of the user ID (this is because Joomla! assigns user ID 0 to unregistered users).

Using the keyword reg in place of the ID refers to all registered users. Therefore the command

```
{jsmallfib [Documents] 0:1 reg:3 42:6}
```

defines level 1 access for all visiting users, level 3 for all registered users except user ID 42, who is granted level 6.

Permissions for entire groups of users

JSmallfib allows repositories to be present different access levels to different groups of users. To this end, JSmallfib links to the Joomla! User Group IDs. When setting a specific permission level for a User Group, simply use the relevant ID on the left side of the permission option, but prefix this ID with the letter g (for 'group'). The command

```
{jsmallfib [Documents] g6:3 g4:1 42:5}
```

assigns permission level 3 to users in group ID 6, level 1 to users in group ID 4, while user ID 42 has again permission level 6.

In all cases, notice that the backend parameter Default Visitor Access Rights implicitly defines the minimum access level. If this parameter is set to 3, for example, then a command such as

```
{jsmallfib [Documents] 12:1}
```

will **not** assign permission level 1 to user ID 12, as one might think, but the default visitor level 3 will be assigned instead. This makes sense as it would then be possible for user ID 12 to logout, access the repository as a non registered user, and get a higher permission level that if he/she was logged in.

Displaying a separate repository for each individual user

With JSmallfib you can have a single article display different repositories for different users. To give each user access to an individual repository, you can use the command

```
{jsmallfib [USERBOUND]}
```

The keyword USERBOUND must be enclosed in square brackets, as it replaces the indication of a specific repository folder (such as [Documents]). When JSmallfib sees this keyword, it will automatically use a folder named after the user who is currently viewing the article. This has the effect of using a separate repository for each user. Users will be allowed access to their own repositories transparently, and will not be allowed to access or even see the 'parallel' repositories of the other users.

Permissions applied to the USERBOUND repositories are set by the value of the backend parameter 'Default Personal Access Rights and cannot be overridden by a command option.

As an administrator, you may want to be able to access (with full access rights, for example) all individual personal areas. Remember that the administrator (usually linked to user ID 42) will be treated just like any other user by the USERBOUND command. Accessing a page containing the previous command will display a separate repository for the administrator, too. However, if you create a new article, containing the command

```
{jsmallfib 42:5}
```

this will allow user ID 42 (the administrator) full access to the Default JSmallfib Path folder, which is located just above the personal-folder areas, making those areas fully accessible to the administrator.

Finally, you may want to customize the actual folder names used in the personal areas. These are named by default Personal area for user ID x (where x is the user ID), but this setting can be changed with the backend parameters Userbound prefix, Userbound parameter, Userbound suffix, located in the Permissions Settings → Userbound repository access levels.

Anything from Area for user named John Smith to Username jsmith - personal area is possible. You can identify users not only by their IDs but also via their names, usernames (as used for login), or by both ID and username.

Displaying a separate repository for each group of users

If you want a group of users to have access to the same repository (each group accessing a separate repository), you can use the command option

```
{jsmallfib [GROUPBOUND]}
```

This is similar to what happens with the USERBOUND functionality. The keyword GROUPBOUND must be enclosed in square brackets, as it replaces the indication of a specific repository folder (such as [Documents]). When JSmallfib sees this keyword, it will automatically use a folder named after the Joomla! User Group to which the user who is currently viewing the article belongs. Users will be allowed access their group's repository transparently, and will not access or even see the 'parallel' repositories of the other groups.

Permissions applied to the GROUPBOUND repositories are set by the value of the backend parameter Default Group Access Rights and cannot be overriden.

Just as in the case of the USERBOUND functionality, you may want, as administrator, to be able to access (with full access rights, for example) all group areas. If you create a new article, containing the command

{jsmallfib 42:5}

this will allow user ID 42 (the administrator) full access to the Default JSmallfib Path folder, which is located just above the group-related folder areas, making those areas fully accessible to the administrator.

Finally, you may want to customize the actual folder names used in the group areas. These are normally named Shared area for group ID xx (where xx is of course the relevant user group ID), but this setting can be changes with the backend parameters Groupbound prefix, Groupbound parameter, Groupbound suffix located in the Permissions Settings → Groupbound repository access levels. Anything from Area for group Friends to User Group ID 33 - shared area is possible. You can identify user groups not only by their ID but also via their name.

When using this option, **users that belong to more than one group will have the ability to switch between the relevant repositories**, by clicking on a link displayed on the top of the repository.

Notice that JSmallfib has the ability to nest USERBOUND repositories inside usergroup-related (GROUPBOUND) paths. Let's examine a situation when this functionality becomes useful (you will need to refer to the section 'Locate repositories independently of the Default JSmallfib Path for more details on relpath() and abspath() command options).

Imagine you have a recruiting company, and you have 5 recruiters. Each recruiter has 15 applicants who need to upload their CVs. What you want to do is have each recruiter have a user group, so that when new applicants are added they are assigned to the recruiter. Each applicant can upload a CV, but only see their own. The recruiter can access the group and all CVs of his/her applicants. An administrator would have top level access to all the files.

JSmallfib would allow you to satisfy the above requirements with the following steps:

1. Enable the backend parameter **Enable overriding of Default Path**;
2. Create a Joomla article for all the people to be recruited, writing inside it the command:

   ```
   {jsmallfib 42:6 [USERBOUND] relpath(jsmallfib_top/GROUPBOUND)}
   ```
3. Create another Joomla article for all your recruiters, placing inside it the command:

   ```
   {jsmallfib 42:6 [GROUPBOUND]}
   ```
4. You now need to create user groups, placing in each group a recruiter and all his/her 'recruitees'. You also need to limit access to the article at point 3 to your recruiters only, while everyone (registered) should be able to access the article at point 2.
5. For the Administrator you could use yet another reserved article with the simple command:

   ```
   {jsmallfib 42:6}
   ```

   to give the administrator (assuming for admin the default ID 42) full access to all folders inside the default JSmallfib path.

Notice that in step 2 above you could of course use the command option abspath() if you prefer to use a path specified in absolute terms (needed when using repositories located outside the web root, for example). Also, you may probably want to modify the jsmallfib_top indicated above to match your current setting of the **Default JSmallfib Path** backend parameter.

Hiding certain files and/or folders

JSmallfib can hide certain files and/or folders from view. A list of files is specified in the backend parameter named **Hidden Files**. This is a comma-separated list of files. Listing file extensions is supported in the format *.ext and you may also specify files by prefix, such as in images*, which will hide all files whose name begins with the string images.

Folders may be also hidden from view if they are listed in a comma-separated list in the backend parameter **Hidden Folders**.

Linking files directly (without showing the options popup box)

With the default configuration, clicking on a file name in a repository (unless permissions do not allow to view/download files) will popup a box with the option to open or dowload/save the file. The backend parameter Method to link files allows the user to modify this standard behavior by linking files directly; image or PDF files, for example, may be directly opened by the browser in either the same or a new window.

However, using a popup box is the recommended method, because this allows JSmallfib to log the opening/downloading of files. Also, notice that if the current position is outside the web root, the popup box option will be used regardless of the backend configuration.

Hiding the navigation links

JSmallfib will normally display, on the top left side of the repository, information on the current folder, with a navigation link chain to all preceding folders, up to the top level area, which is either the folder defined by the backend parameter Default JSmallfib Path – or the one indicated as the argument of the command options relpath() or abspath(), if used – or a specific repository folder, if its name is used in the command as an option in square brackets.

The backend parameter Display navigation will get JSmallfib to hide this information altogether.

Enabling/disabling or resizing image thumbnails

When JSmallfib lists an image file (recognized image files have extensions .jpg, .jpeg, .gif, .png) it displays a square thumbnail whose size is by default set to 30×30 pixels. With backend parameter Size of thumbnails you can modify this size for all repositories. This parameter may be overridden by individual commands using the option thumbsize:x (where x is the required size in pixels – do not enter px, but only an integer number).

A value of 0 (zero) means no thumbnail is required (the relevant file icon is displayed in place of a thumbnail).

Please notice that **thumbnails are not displayed if the current position is outside the web root**.

Changing formats or hiding file size and/or date

You can modify file size separator by changing the backend parameter **File size separator**. Its value could be set as a '.' (dot, default configuration) or a ',' (comma).

File date and time refer to the last modification of the file. The date format is modified by acting on backend parameter **File date format**, which provides for a variety of commonly used date formats. You can also choose to hide the time (acting on parameter **Display time**), or just the seconds (through **Display seconds**).

You can also choose to hide altogether the file size and/or its last modification date (and time) from view in the repository, by appropriately setting the backend parameters **Display size** and **Display date.**

Modifying the default sorting options or hide/display the file filter box

Files and folders are sorted by default by name in descending order. This configuration can be modified through the backend parameters **Default sort element** and **Default sort order**.

Backend parameters of course apply to all repositores. A way of configuring each individual repository is by using the command options **sortby:x** and **sortas:y**, whereby a repository can have its own sorting arrangement.

In these cases *x* can take values of

1 – for sorting by file name
2 – for sorting by file size
3 – for sorting by modified date/time

and *y* is either

1 – for sorting in ascending order (e.g. 3, 4, 5,… or A, B, C, …)
2 – for sorting in descending order (e.g. 5, 4, 3,… or D, C, B, A)

An example of a repository which specifies sorting by the largest file first is

{jsmallfib [Documents] sortby:2 sortas:2}

JSmallfib uses by default natural sorting. This means that files img_1.png, img_2.png, …, img_10.png are sorted by default as

img_1.png
img_2.png
...
img_10.png

while if the Natural sorting option is disabled they are displayed as

img_1.png
img_10.png
img_2.png
...

This is of course valid for folders as well as files. You can override the backend parameter using the sortnat:z, where z is 0 (zero) for non-natural sorting, 1 (one) for natural sorting.

JSmallfib also allows filtering of files by showing an input box at the top right side of the repository. You can filter the file names based on patterns; you can use more that one pattern in a semicolon-separated list, such as .jpg; .pdf; .doc

If you wish to disable file filtering, you can disable the backend parameter Allow filtering.

Allowing unzip of compressed files

When JSmallfib detects a compressed file, it gives users with access rights equal to at least 4 (delete files) the option to uncompress it remotely on the current position of the repository.

Supported formats for compressed files are bz2, bzip2, gz, gzip, tar, tbz2, tgz and zip.

You can disable this option either by default for all repositories, by appropriately selecting the backend parameter Allow unzip, or, for an individual repository, by adding the option unzip:0 in the JSmallfib command (use option unzip:1 to enable file decompression).

Hiding or displaying the info link

With the default configuration JSmallfib displays a 'info' link on the top right side of the repository. This link may be hidden by using the backend parameter **Show info link**.

Configuring the Flash upload module

This commercial version of JSmallfib features an upload module based on the SWFUpload Flash upload tool. This is enabled by default, although it can be disabled from the **Upload Settings** tab in the back end.

Notice that even though the Flash upload module is enabled, the user always has the option (in the front end) to switch to the standard HTML upload method, in case his/her browser does not support Flash. However, if the Flash module is disabled from the backend, users will only have the option to use the standard HTML upload method.

In the **Upload Settings** tab of the plugin configuration page you have access to this module's settings. They are only supported by the Flash upload module, and would have no effect if the **Upload type** parameter is set to **HTML upload**.

Using these parameters, you can limit file upload to a maximum size (by appropriately setting the **Max file size** and **Max file size unit** parameters). To set no limits, leave the default value of 0 (zero). However, keep in mind that you still have the limitations imposed by the server configuration, so you need to make sure that the server settings (please refer to the first FAQ in the relevant section, named 'Does JSmallfib have a file size limit when uploading?') are configured so as to allow the upload of files of the required size.

Another option, named **Upload**, allows you to enable the uploading of multiple files at once. This is enabled by default, but can be disabled here.

When uploading files with the Flash module, you will see, in the front end, a drop-down list of options close to the upload button. These options define the required behavior of JSmallfib when uploading files which already exist on the current folder. These options are:

- ask what to do

- leave them
- override them
- archive them

If the first option is chosen, then JSmallfib will ask what to do if it finds that one or more files in the uploading batch have an equivalent file with the same name in the current folder. It will display a table with the list of such files, each with a drop-down list where the user can specify whether each file should replace the existing one, or whether that particular file should not be uploaded. If the Permission Settings → Allow file archiving parameter is enabled, the user will also have the option to archive the existing files before the upload (see section named 'Disabling/enabling the option to archive files' below for further details on file archiving functionality of JSmallfib).

The above parameter only defines the default behavior. The user, before any upload, will have the option to select a different option from the drop-down list displayed in the front end.

A further option of the Flash upload module, named File upload limit, defines the maximum number of files that can be uploaded at once (in case of multiple file upload).

The File type parameter provides for file upload filtering based on extensions. The default value of *.* means that all extensions are allowed, while a semicolon-separated list of extensions, such as *.png;*.jpg;*.gif will limit upload to those specific file types. In this case, the parameter File types description allows the administrator to display, in the upload selection box, a description of the file type allowed. For example, in the above example of PNG, JPG, GIF extensions, a suitable description would be Image files.

Modifying the overall look and feel of the display

With JSmallfib it is possible to act, from the backend configuration page, on most of the parameters which define the styling (CSS style sheets) of the display. You can modify the width of the main display (this parameter may be overridden with the command option width:x) to the colors and styles of the various borders and backgrounds used. These parameters are listed on the Look & Feel Settings tab of the plugin configuration page.

This commercial version JSmallfib expands the Main options (apply to all boxes) section of the Look & Feel Settings tab, to include options to switch to a different icon set (in fact, the new icon set, named Smallerik, is enabled by default in this version) and to add a drop shadow to the boxes used by the application.

The new icon set is based on the Nuvola set at www.stdicon.com, extended with proprietary graphics to make the overall look of the repository more modern. Over 100 file extensions are recognized by the new set (although some similar extensions will display with the same icon, such as for image files, videos, 3D models, word-compatible documents, and the like).

The new drop shadow, which can be tweaked in terms of color, width, blur, is intended to increase the ability to graphically integrate the JSmallfib repositories with the specific template used in the Joomla installation.

Displaying a description title for a specific repository

Each repository can have its own description title. All you need to achieve this is to add a command option named desc(*description*) such as in the command

```
{jsmallfib [Documents] desc(Project documentation)}
```

Disabling/enabling the option to archive files

When uploading a file, if a file with the same name already exists, the user is given the option to cancel the upload, override the existing file, or archive the existing file before overriding it. Archives are invisible subfolders that contain the original copies of overridden files. If a normal folder contains archived copies of the files listed, the repository will display a link to visit the archive for that folder. In the archive area, copies of the original files are kept, with names modified to include the date and time they were archived. Archived files can be opened in the archive or restored to their original folder (the folder immediately above the archive) by clicking on the link below each archived file.

It is possible to disable the archiving functionality by appropriately setting the backend parameter Allow file archiving, which is listed in the Permissions Settings → Other permissions section of the plugin configuration page.

Managing the actions logs (including email options)

By default, the JSmallfib plugin does not perform any logging function. However, by appropriately setting the relevant backend parameters, located within the Activity Logs Settings tab in the plugin configuration page, it is possible to log any of the following actions:

- file uploads
- file downloads
- removal of folders
- removal of files
- restoring files
- creation of new folders
- renaming folders
- renaming files
- uncompressing files

Logs are displayed by clicking on the link on the top-right corner of the repository. This link is only visible to users who have an access level of 6.

In the file system, logs are kept in files within the Joomla! logs folder. Each file contains the logs for a specific action and for a specific repository. An example of a log file is

logs/jsmallfib_log_f1079588ddb79f5973a2d73cfa167b15_uploads.txt

or, for a JSON format log file (see below)

logs/jsmallfib_log_f1079588ddb79f5973a2d73cfa167b15_uploads.json

for a log of file uploads; the string f1079588ddb79f5973a2d73cfa167b15 is used to uniquely identify the repository to which this log file refers. From version 1.3.2 of the plugin, this log ID is now visible under the title of the log display page.

Version 1.3.2 also introduces the following options:

- you can ask JSmallfib to use the HTML format when sending log emails (if enabled)
- you can specify a JSON format for the log files (see below for the structure used)
- you have the ability to log each action in a separate file or to log all actions for that repository in the same file
- you can specify the log directory, which can be either inside or outside of the web root (by indicating whether the path specified should be intended as relative to the Joomla! root or as an absolute path

If using the JSON format, log files will be stored with the following structure for each recorded action:

```
{
   "timestamp":"2012-03-26 12:32:38",
   "new_element":"test new name",
   "info":"",
   "action_type":7,
   "action_result":0,
   "element":"test",
   "location":"\/smallerik\/jsmallfib_top",
   "userid":"42",
   "username":"Super User",
   "user_username":"admin",
   "remote_address":"x:y:z"
}
```

where the above will be written in a single line of the file. The elements in blue are the values (they change depending on the actual action logged), while the elements in black are the parameters used. Apart from the self-explaining ones, the parameter "info" stores extra log information (e.g. in case of action failure); the "element" is the file or folder interested by the action; the "new_element" stores the new name of the file/folder (in case of a renaming action – it's blank otherwise); the "action_result" is 0 for successful actions, and takes other values in case of errors; finally, the "action_type" records the action according to the following scheme:

1 – File uploads

2 – File downloads

3 – Removal of folders

4 – Removal of files

5 – Restoring files from an archive location

6 – Creation of a new folder

7 – Renaming of a folder

8 – Renaming of a file

9 – File unzipping

Ultimately, log files are just text files, and they are created by JSmallfib whenever they are needed. Therefore, **if you need to backup a log file all you need to do is to rename that file or move it to a backup folder**, for example. A new log file with the original name will be recreated as soon as JSmallfib is requested to log a specific action. Logs can be inspected with a standard text viewer, or processed by setting up a custom script.

Locating repositories independently of the 'Default JSmallfib Path'

When using a command such as

```
{jsmallfib [Documents]}
```

the folder [Documents], top level folder for this repository, is located within the Default JSmallfib Path as defined by the relevant backend parameter in the plugin configuration page. This parameter may be overridden by a command option, allowing a Joomla! article to display files and folders in a repository located somewhere else on the server, either within or outside the web root. This is achieved with the command options abspath(*absolute_path*) as in the following examples

```
{jsmallfib [Documents] abspath(/home/users/repositories)}
```

```
{jsmallfib [Documents] abspath(c:\Users\Repositories)}
```

Similarly, the command option relpath(*new_relative_default_path*) allows the Default JSmallfib Path to be overridden with a new path expressed in terms relative to the current Joomla! installation folder, such as in

```
{jsmallfib [Documents] relpath(logs/special_docs)}
```

where the repository Documents will be located within the area logs/special_docs inside the Joomla! installation. In both cases, no space is allowed between the keyword abspath or relpath and the opening bracket.

Important: In order to be able to use the abspath() and relpath() command options, you first need to enable the backend parameter Enable overriding of default path from the plugin configuration page in the Administrator site, which is disabled by default.

Defining 'trusted authors' for articles containing repositories

Normally, any user with the right permission to write a Joomla! article will be able to write an article containing a {jsmallfib} command. This will give the user potential access to the filesystem.

Some protection is given by the backend parameter Enable overriding of default path, which, when disabled, limits the ability of a JSmallfib repository to list files outside the scope defined by the parameter Default JSmallfib Path. However, there are situations where Administrators might want to restrict the ability to create JSmallfib repositories to a subset of the registered users.

From version 1.3.2 of the plugin this is possible by appropriately defining the backend parameter Advanced Options → Trusted authors, which is a comma-separated list of trusted user IDs and/or user group IDs (when the ID is prefixed by the letter g, similarly to when specifying user groups in the permission settings).

When JSmallfib is parsing a Joomla! article, these specific IDs are evaluated before everything else. If the article was written or last modified by a user either not specifically listed, or belonging to a group (or more groups) not present in the Trusted authors parameter, then the repository contained in the article will not be displayed.

Modifying the default CHMOD settings (advanced users)

By default, on a Unix filesystem, JSmallfib will set the permissions of newly created directories to 0775, while newly uploaded files will have permissions set to 0664. It is possible to modify these values by changing the backend parameters Directory chmod and File chmod, located within the Advanced Options tab of the plugin configuration page.

Hiding the credits icon

This commercial version allows the administrator to hide the credits icon displayed in the front end (located in the bottom right side of the repository). This is an image displaying the wording jsmallfib;

mouse hovering it will display the name of the plugin and of the developer (with copyright wording), as well as the installed version of the software.

Changing the icons used by JSmallfib

JSmallfib uses icons taken from its default installation. These are located inside your Joomla! installation in folder /plugins/content/jsmallfib/media/<*icon-set*>/extensionsIcons (for the file icons) and in folder /plugins/content/jsmallfib/media/<*icon-set*>/navigationIcons (for the other icons), where <*icon-set*> needs to be replaced by the icon set used. Current values are original and smallerik for the respective icon sets implemented. These folders contain PNG image files that can be modified or replaced for customization purposes.

In particular, changing a file icon is as simple as replacing the PNG image file named *ext*.png, where *ext* is the file extension to which the icon refers. For example, icons for PDF and Word files are named pdf.png and doc.png respectively.

Modifying a string of text or creating a new translation for JSmallfib

All text strings used by JSmallfib are kept in different translation files (one for each supported language). JSmallfib comes packaged with a number of different translations: apart from English, it is translated in Danish, Dutch, French, German, Italian, Portuguese, Spanish and Swedish. The relevant translation files are installed depending on the currently installed languages on a specific Joomla! installation. In the Administrator site, go to Extensions → Language Manager to see which languages are installed for the Site and Administrator areas. For example, if you have installed English and Italian on Site (the front-end site), then you will have installed the following two files:

\administrator\language\en-GB\en-GB.plg_content_jsmallfib.ini

\administrator\language\it-IT\it-IT.plg_content_jsmallfib.ini

This way it is possible to switch from one language to another in the display of JSmallfib. Also, in this case, front end users who have set their language preference to be English will see the repositories in English, while Italian users (that is users whose language preference is Italian) will see them displayed in Italian.

Let's now see how an individual translation file looks like. It is a text file encoded using UTF-8, and it contains a number of lines, each representing a text string (no end-of-line breaks are allowed in a text string). Each line is made of two parts separated by a = (equal) sign. The right side is the actual text string to be displayed, and is enclosed in double quotation marks. For example, in JSmallfib, the strings used in the English translation to display the different access levels in the backend configuration page look like:

```
MAY_NOT_ACCESS_REPOSITORY_AT_ALL="0. May not access repository at all"
CAN_ONLY_NAVIGATE_THE_REPOSITORY_NO_DOWNLOAD="1. Can only navigate the repository (no download)"
CAN_ONLY_NAVIGATE_THE_REPOSITORY_AND_DOWNLOAD="2. Can only navigate the repository (and download)"
MAY_CREATE_FOLDERS_AND_UPLOAD_FILES="3. May create folders and upload files"
MAY_DELETE_FILES_BUT_NOT_FOLDERS="4. May delete files, but not folders"
MAY_DELETE_FILES_AND_EMPTY_FOLDERS="5. May delete files and empty folders"
MAY_VIEW_LOGS="6. May additionally view logs"
```

You are more than welcome to create a new translation for JSmallfib. In this case, the best option is to use an existing translation as a template, and then rename it to xx-YY.plg_content_jsmallfib.ini. You will then need to repackage the plugin and reinstall it.

If you would like your translation to be included in the next official release of JSmallfib, please do not hesitate to contact support at web@smallerik.com

Command syntax

Here you find the syntax for all available options that can be used inside a JSmallfib command within a Joomla! article. Notice that only one command can be placed inside an article. You will need to use two different articles to display two repositories.

JSmallfib commands are normal strings contained within the curly braces { and }. The opening brace must be immediately followed (no space is allowed here) by the keyword jsmallfib. Example:

{jsmallfib}

When the above command is placed inside a normal Joomla article, and the article is saved and viewed in the front end directly (not in a list of articles such as on the frontpage or in a category list in blog mode, for example), a file repository is displayed. This repository will point to the default area (as defined in the backend configuration parameter Default JSmallfib Path) and will be available to users according to the permissions set in the backend plugin configuration page.

To customize the configuration you may access and modify the available parameters in the backend plugin configuration page. There, hovering with your mouse on each parameter's name will display more information on how to use them. Some backend parameters' values may be overridden by individual repositories, using specific options in the JSmallfib command.

The following is a list of all the available options:

Repository options

option syntax	description	examples

[repository_folder]	This option is the name of the repository to be used. It must be surrounded by square brackets and it is the name of the folder inside the Default JSmallfib Path that will be used as the top level folder for this repository. repository_folder must be the name of a single folder and cannot contain any path indication. If you require this folder to be located into a path other than the default one as specified in the backend parameters, use this option in conjunction with one of the options below abspath(xxx) or relpath(xxx)	{jsmallfib [Documents]}
abspath(path)	where path is an absolute path that overrides the Default JSmallfib Path parameter in the backend. This option makes JSmallfib independent of the default path set in the backend and allows two separate articles to contain repositories located in areas that do not share a common default path. No space is allowed between the keyword abspath and the opening bracket. This option needs to be enabled using the backend parameter Enable overriding of Default Path.	{jsmallfib [Documents] abspath(C:\Repositories)}
relpath(path)	where path is a path, relative to the web root, that overrides the Default JSmallfib Path parameter in the backend. This option makes JSmallfib independent of the default path set in the backend and allows two separate articles to contain repositories located in areas that do not share a common default path. No space is allowed between the keyword relpath and the opening bracket. This option needs to be enabled using the backend parameter Enable overriding of Default Path.	{jsmallfib [Documents] relpath(/files/docs)}
[USERBOUND]	When the repository_folder is set to the keyword USERBOUND surrounded by square brackets, then JSmallfib will create a separate repository for each user. The user-related areas are folders located inside the default JSmallfib path and the folder names created	{jsmallfib [USERBOUND]}

	depend on what is defined in the relevant backend parameters, where you can specify whether to use a user ID, a user name, a username (used for login), a user ID - username, as well as (optionally) a prefix and a suffix	
[GROUPBOUND]	When the repository_folder is set to the keyword GROUPBOUND surrounded by square brackets, then JSmallfib will create a separate repository for each Joomla! User Group. These group-related areas are folders located inside the Default JSmallfib Path (which must be a relative path) and the folder names created depend on what is defined in the relevant backend parameters, where you can specify whether to use a user group ID or its name, as well as (optionally) a prefix and a suffix. When using this option, users that belong to more than one group will have the ability to switch between the relevant repositories, by clicking on a link displayed on the top of the repository.	{jsmallfib [GROUPBOUND]}

Note that you have the ability to nest USERBOUND repositories inside usergroup-related (GROUPBOUND) paths. Please refer to the above section 'Displaying a separate repository for each group of users' for an example of when this functionality becomes useful.

Access levels options

option syntax	description	examples
user_ID:permission_level	This option, which may be repeated for any number of user IDs inside a command, assigns the user identified by ID user_ID the access right permission_level. Each pair may be enclosed in square brackets, or separated by commas, although this is not required. Currently allowed permission levels are integer numbers from 0 to 6, according to the following list:	{jsmallfib 42:6 12:4}

	0. May not access repository at all 1. Can only navigate the repository (no download) 2. Can navigate the repository and download files 3. May create folders and upload files 4. May delete files, but not folders 5. May delete files and empty folders 6. May additionally view logs	
0:permission_level	This is a special case of the above option, used to refer to all visiting users (non registered users). It overrides the equivalent backend parameter. Permission levels are as described above	{jsmallfib 0:1}
reg:permission_level	This is another special case of the above option, used to refer to all registered users. It overrides the equivalent backend parameter. Permission levels are as described above	{jsmallfib reg:2}
g*user_group_ID*:permission_level	This option, which may be repeated for any number of Joomla! user group IDs inside a command, assigns the Joomla! user group identified by ID the access right permission_level. Each pair may be enclosed in square brackets, or separated by commas, although this is not required. These IDs are found on the right of the screen available from Users → Groups in the standard Joomla! backend site. With this option you can assign the same access level to an entire group of users, and can also specify more groups within the same command	{jsmallfib g6:3 g8:4}

Other options

option syntax	**description**	**examples**
thumbsize:*size_of_thumbnails*	This option overrides the equivalent backend default parameter. It is the size of the square thumbnail which appears in place of the file icon when an image file is found. Recognized image files have extensions .jpg, .jpeg, .gif, .png. A value of 0 means no thumbnail (the relevant file	{jsmallfib thumbsize:45}

	icon is displayed instead	
sortby:*element_to_sort*	This option overrides the equivalent backend parameter that specifies which is the default element to be sorted. Allowed values for *element_to_sort* are 1 for file and folder names, 2 for file size, 3 for file modified date and time	{jsmallfib sortby:2}
sortas:*sorting_type*	This option overrides the equivalent backend parameter that specifies which is the default sorting type for the default sorted element. Allowed values for *sorting_type* are 1 for ascending, 2 for descending	{jsmallfib sortby:2 sortas:2}
unzip:*x*	This option overrides the backend parameter Allow unzip. A value of 0 disables the ability to unzip compressed files, while 1 enables it for the specific repository	{jsmallfib unzip:0}
width:*main_width*	This option overrides the backend parameter Main display width, thus allowing to customize the width of a specific repository. The value of *main_width* is in pixels (no px is needed after the value)	{jsmallfib width:480}
desc(*description*)	This option sets a description as a title just above the repository	{jsmallfib [Documents] desc(Project documentation)}

Frequently asked questions

Does JSmallfib have a file size limit when uploading? How can I increase the limit?

Three PHP configuration parameters do influence the maximum file size for uploads. These are memory_limit, post_max_size, and upload_max_filesize.

A small PHP script contained in file limits.zip will allow you to test your configuration with respect to these settings. You can find this file under the Tools area of Smallerik site. Download and unzip the file onto any folder within your web root and call it from your browser. It would be something like:

```
my_website_url_here/limits.php
```

This will display on your browser something like:

```
The setting of PHP configuration parameter 'memory_limit' is [128M]
The setting of PHP configuration parameter 'post_max_size' is [8M]
The setting of PHP configuration parameter 'upload_max_filesize' is [8M]
```

Of course the actual values you'll see might be different. The important thing is that memory_limit is greater than post_max_size and the latter is greater than upload_max_filesize, which in turn should be greater that the actual maximum file size you want to upload on your site!

So, if you have access to the php.ini file (the configuration file for PHP), you can try to change all of the three parameters there. Otherwise, another way to achieve the same thing is to add the following two lines in file .htaccess in your main web root folder:

```
php_value upload_max_filesize 20M
php_value post_max_size 20M
```

where 20M can of course be changed to whatever you prefer, provided this is less than memory_limit you saw with the limits.php script. Normally this last value is large enough not to be changed, otherwise you could include a further line in the .htaccess file (or directly change the relevant parameter in php.ini).

Unfortunately these parameters cannot be changed from within an application (so JSmallfib cannot do this for you automatically).

JSmallfib displays the repository correctly by I cannot navigate in it. Clicking on any folder will not do anything.

This problem used to happen to JSmallfib versions prior to 1.0.28 for users having the PHP setting magic_quotes_gpc enabled, but might still happen on later versions depending on specific server configuration.

A PHP script contained in file quotes.zip helps detect this settings. You can find this file under the Tools area of Smallerik site. Download and unzip this file and put it in the main web root, then call it from the browser (it should be:

```
my_website_url_here/quotes.php
```

If the first parameter is 1 then magic_quotes_gpc is on.

To sort it out, if you don't have access to PHP parameters directly (through file php.ini, which would be the best option), you can create a file .htaccess in the main web root with the following content:

```
SetEnv MAGIC_QUOTES 0
SetEnv PHP_VER 5
```

If the .htaccess is there already you just need to add these two lines and all should work.

I have installed and enabled JSmallfib, created an article with the right command, but all I get is a line saying 'This line is replaced by a file repository when the article is viewed'

You are getting the message This line is replaced … because you are viewing the article in a list of articles (such as when viewing in blog mode in a Joomla page, or when displaying featured articles in the home page). To properly display the repository, each article needs to be viewed directly, for example by clicking on its title when viewing articles in a list (you may have to enable clickable titles in the article parameter configuration).

So, when a single article is viewed, JSmallfib will display the repository.

JSmallfib does not properly display special characters, such as accented letters, in file names. What can I do about that?

JSmallfib has a backend parameter named Encode to UTF-8 and located in the Display Options → General display options section. The default value is Yes; however, if strange characters appear in your file names in place of special characters such as accented letters, it is possible to disable this setting to solve this display problem.

JSmallfib does not recognise the starting folder (or does something strange not described in the above questions). What can I do about that?

It might happen that the actual JSmallfib command gets messed up with html tags put there by your editor. It is therefore worthwhile to check if this is the case. To this end, open the article containing the {jsmallfib ...} code and click (if you use the standard MCE editor) on the HTML button on the second line of the icons, in order to view the article's actual HTML text. Then look at the actual code, and if you see html tags (they appear in the form of <xxx> tags) inside the {jsmallfib ...} command, you should remove them.

In fact, remove anything that you don't expect to see inside such command, then update the edited article and save it. This should solve the problem.

I have placed two JSmallfib commands in the same article, but only the first one seems to work. Is this normal?

Yes, currently JSmallfib does not support more than one instance on the same article page.

JSmallfib crashes on pages with image files listings. What's the problem here?

In pages with image files listed in the repository, JSmallfib will attempt to create and display thumbnails for those images. Not all server configurations support this functionality. In particular, PHP needs to be compiled against the GD libraries (at least version 1.8).

Before trying to work on thumbnails, JSmallfib now checks if an appropriate server configuration is available, so this crash problem should not happen any more.

Support

Should you have a problem, or cannot figure out how to use a specific function, or again would like to know if a specific requirement of yours is already available in the current version of JSmallfib, please read through the information provided in this document. It could also be helpful to hover with your mouse onto the backend parameters' names, to display hints on usage that might help understand the functionality.

However, if you feel that your specific question or issue is not covered, you can ask for support via email at web@smallerik.com

Wish list

The following list outlines some of the features users and development would like to see in future versions of JSmallfib. Without guarantee on time schedules, these are the potential candidates for new functionality; it will of course change as new items are added and existing ones implemented:

- progress bar when uploading files (available in the commercial version of JSmallfib)
- multiple-file upload capability (available in the commercial version)
- more command options, to override more of the equivalent backend parameters
- set up a separate permission system (possibly to work in parallel with the standard one) where each function can be set individually (e.g., delete folder, but not rename, etc.)
- it would be nice to have a set of file extensions that JSmallfib opens directly (David Carson)
- get JSmallfib to read a repository on a shared network drive (George Dontas)
- get JSmallfib to work with K2 (Cristian Belgun George and others)
- ability to display descriptions for files (Simon Watson) based on database information
- ability to configure font family and size (Stefano Codarini)
- ability to search files - either through the built-in Joomla! functionality or with a special function in JSmallfib (Andreas)
- direct editing of text files (Jesse Juhlke)
- ability to move files between folders (Jenn)
- ability to share a file (Raven C)
- introduce optional limit to folder size (also per user or per group)
- icon displaying if a folder has content added to it (Lyndsey Jackson)

Change log

This is the full change log of the commercial version of the JSmallfib plugin software.

29 March 2012 - Version 1.3.2

- Added more logging options, including the ability to store action logs in JSON format, to specify the log directory, to indicate whether to store all logs into a single file or use a separate file per action, and to optionally send log email in HTML format.
- Introduced a 'trusted-authors' feature, so that only articles written (or last modified) by specific users will display a repository if a {jsmallfib} command is contained in the article. To this end, a backend parameter is available to list the user IDs or group IDs of these 'trusted' users.
- This version also fixes a bug with the removal of image files not logged if one image file remained in the directory after the remove action.
- The front-end log page now displays, below the page title, the log ID used by the log file(s).

17 March 2012 - Version 1.3/Version 1.3.1

- CSS styling now more insulated, to improve compatibility with the templates used and other extensions. Also, JSmallfib now uses the template default font ans font size
- Various bug fixes (version 1.3.1 fixes a warning on servers running PHP 5.3)

24 March 2011 - Version 1.2.1

- Fixed problem with JSmallfib not properly sorting folders
- Introduced a new sorting option (enabled by default) to provide for *natural order string comparison*
- Now the keyword GROUPBOUND may be included in the default path (either from the backend or using the relpath() or abspath() command options), so that more flexibility is provided for special cases

25 January 2011 - Version 1.2

- JSmallfib package now upgradable - no need to uninstall current version and no need to re-enter configuration parameters after installation
- Introduced ability to unzip compressed files remotely (based on code volunteered by Francisco José Esteban)
- USERBOUND and GROUPBOUND functionality now available even outside the web root or when using absolute paths
- Now JSmallfib will attempt to create the top level folder even if this is expressed as an absolute path
- Introduced options to hide the navigation bar on the top left side of the repository and the file size and/or date (based on code volunteered by Ben Sandberg, who also made the plugin upgradable)
- Improved sorting functionality (ascending/descending now working as expected)
- New USERBOUND folder naming option 'user ID and username'
- Fixed fatal error due to missing GD libraries needed for thumbnails (in rare server configurations)
- relpath() and abspath() command option now work as expected when a repository option is used
- Main display width now configurable as a command option (width:x)
- Fixed annoying 'plus' icon getting sometimes in the way where not needed, as well as some other minor bits and pieces …

17 August 2010 - Version 1.1 (with 1.1c, introducing a fix on unix permissions and thumbs visible only in relative paths)

This release addresses some security concern related to the display of the full server path. This information is now never transmitted to or displayed by the browser.

Other modifications include:

- Thumbnail functionality for image files (JPG, PNG, GIF); it may be disabled in the backend by setting the default thumbnail size to 0 (zero) or in the front end by adding the command option [thumbsize:0]
- USERBOUND and GROUPBOUND repositories are now made more flexible from the backend
- The default path may now by overridden using the command parameter abspath() or relpath()
- Default chmod for files and folders may now be set from the backend
- Introduced a new backend parameter to define the file size separator
- Introduced a new file filtering functionality, to filter files to be displayed in the front end (may be disabled using a backend parameter)

14 June 2010 - Version 1.0.37

- Added optional email notification of logs. This can be specified individually for each log type, so you can get an email notification on uploads only, for example
- Introduced command option overrides for default registered users access level [reg:x], default sort element [sortby:x] and default sort order [sortas:x]: their usage is explained in the description of the relevant backend parameters (mouse-hovering on the relevant parameters' names)
- Some CSS styles are now renamed to improve compatibility with other extensions
- In case of warnings (such as access not allowed), the software now reproduces the original article and not just the warning

18 May 2010 - Version 1.0.36d

- This release fixed a bug with the sorting functionality, which is improved by allowing a new backend parameter to specify the default sorting element (name, size or last-modification date). Also, the sorting arrows are now properly displayed, appearing only in the sorted column.

07 May 2010 - Version 1.0.36

- Introduced the GROUPBOUND parameter, whose functionality is similar to the USERBOUND one. It is now possible to display a different repository for each Joomla! user group [user groups were contact categories at the time of the release, as user groups were introduced with Joomla! version 1.6 in January 2011] from the same article

- Introduced a new backend parameter to change the default sort order for files and folders
- Introduced a new parameter to hide the files last-changed time
- Fixed a few bugs and introduced some small enhancements: special characters are now supported by the repository default path name; fixed username not appearing in the uploads log; deprecated **ereg()** PHP function now replaced; fixed problem with some files corrupted during download, only on certain configurations (I had a few reports on this issue); the username now shown when navigating a **USERBOUND** repository, in place of the less user-friendly user ID; plus some other bits and pieces

05 January 2010 - Version 1.0.35

- Introduced the ability to specify a repository named **USERBOUND**, which allows an article to create a separate repository for each individual user: in this way each user sees his/her own repository when viewing the same article
- In the backend parameter to hide certain types of files, these can now also be specified using a prefix and an asterisk, such as in **xyz*** or **xyz.***
- When uploading a new version of an existing file, it is now possible to choose between ignoring upload, overriding the existing file and (depending on the backend configuration - default set to yes) archiving a copy of the existing file before upload
- Fixed the problem of a blank page displaying when trying to view a long list of files

19 October 2009 - Version 1.0.34

- Added translation to Portuguese. Special thanks to António Mendes of http://www.ammsoftware.com/mendes

15 October 2009 - Version 1.0.33b

- Important security fix
- Added translation for Danish language. Special thanks to Thomas Sydenham of http://imcculand.dk

- Added code by Marcos Ramirez to allow downloads beyond memory_limit PHP parameter. Marcos also provided an update to the Spanish translation

11 September 2009 - Version 1.0.32

- Added backend parameter to disable UTF-8 encoding when displaying file names (this fixes cases of wrong display of special characters such as accented letters)
- Fixed bug when checking for existing file when uploading
- Minor fixes to avoid variable warnings

04 September 2009 - Version 1.0.31

- Added translation for Swedish language. Special thanks to Erik Liljencrantz of http://www.eldata.se/
- This release has seen some contributed code, as follows:
- Avoid overwriting files and folders with the same name as existing ones (Erik Liljencrantz)
- Introduce error message when trying to delete a non empty folder (Erik Liljencrantz)
- Added more userfriendly navigation with clickable directories in the current directory path, now on top (Erik Liljencrantz)
- Removed links to go to top level and refresh, now replaced by Erik Liljencrantz new navigation links
- Added compatibility to Joomla! component EventList
- Now date formats are configurable in the back-end
- Fixed problems with deleting and downloading files with special characters (Erik Liljencrantz)
- Fixed problem when trying to download very large files
- Improved German translation (Thomas Schaller)

16 July 2009 - Version 1.0.30

- Added translation for Dutch language. Special thanks to http://www.mwprojects.nl/

02 July 2009 - Version 1.0.29

- Fixed bug with group permissions not correctly read depending on the order of the relevant command arguments
- Added translation for Spanish language. Special thanks to Tux Merlìn of http://www.moatsoft.com.ar/

24 June 2009 - Version 1.0.28

- Introduced more backend parameters for frontend configuration of line colors, types and thickness
- Fixed problem with magic quotes: JSmallfib now supports PHP setting magic_quotes_gpc enabled (although this feature is deprecated by the PHP team and in fact removed from PHP6)
- Added support for special characters such as ampersand & and single quote
- Last, but not least, JSmallfib is now shipped with translations for French, German and Italian lanuages (in addition to English). Special thanks for the translations go to Cédric Belpaire of http://www.sos-smi.com/ for the French translation and Thomas Schaller of http://www.hpssweb.de/ for the German translation

17 June 2009 - Version 1.0.27

- Fixed problems with PHP4 generating errors such as unexpected T_STATIC, expecting T_OLD_FUNCTION or T_FUNCTION or T_VAR

Added ability to support files and folders containing special characters (such as accented letters)

15 June 2009 - Version 1.0.26

- Fixed a bug with icons not displaying when using the option search-friendly URL in Joomla! Global configuration, and added option to hide help link (which is then replaced by a small 'js' icon on the bottom-right corner)

Copyright & Licence

Enrico Sandoli and Smallerik are not affiliated with or endorsed by the Joomla Project or Open Source Matters.