Macrotone Issue Tracker Design Guide

Geoffrey S. Chapman, Macrotone Consulting UK Ltd.

Macrotone Issue Tracker Design Guide:

by Geoffrey S. Chapman

Publication date November 2014

Abstract

This book covers the use of the Macrotone Issue Tracker component which allows you to control issues or problem reports on your Joomla!™-powered web sites.

Table of Contents

List of Figures

List of Tables

Preface

Joomla

Issue Tracker

Component

Author:	G S Chapman
Date:	13th November 2014
Version:	1.6.5

DOCUMENT HISTORY

Version	Date	Changed By:	Remarks
1.3	26/02/2013	G S Chapman	Split from the original documentation the design guide is now a separate document for release 1.3.0
1.3.1	18/04/2013	G S Chapman	Updated for release 1.3.1.
1.5.0	04/12/2013	G S Chapman	Updated for release 1.5.0
1.6.0	25/02/2014	G S Chapman	Updated for release 1.6.0
1.6.2	14/06/2014	G S Chapman	Updated for release 1.6.2
1.6.5	13/11/2014	G S Chapman	Updated for release 1.6.5

PURPOSE OF DOCUMENT

This documentation describes the design of the release 1.6.5 version of the Joomla Issue Tracker component by Macrotone Consulting Ltd.

This document also contains preliminary information upon (possible) forthcoming feature enhancements. This information is not a commitment to implementation but rather a guide to current thoughts as to possible future features and how they might be implemented.

Part I. Macrotone Issue Tracker

Table of Contents

Chapter 1. Overview

This document concentrates upon the planning, design and creation of the Joomla Issue Tracker component for Joomla! 2.5/3.3. The component has its origin in a 'Problem Management System (PMS)' developed many years ago on an Oracle Database using Oracle Forms. It migrated to an Oracle Application Express (APEX) based application and incorporated ideas from a sample APEX application. This version takes the sample and implements it upon a MySQL database using Joomla 2.5/3.3 as the interface. As time and circumstances permit some of the original program features may/will be re-implemented onto new platform.

The completed Issue Tracker component is available on the Macrotone Consulting Ltd Website:

http://www.macrotoneconsulting.co.uk/index.php/Macrotone/joomla-extensions.html

Wikipedia defines an Issue Tracking system (ITS, trouble ticket system, support ticket or incident ticket system) as a Computer Software package that manages and maintains lists of Issues (computers), as needed by an organisation. Issue tracking systems are commonly used in an organisation's Customer Support / Call Centre to create, update, and resolve reported customer issues, or even issues reported by that organisation's other employees. An issue tracking system often also contains a Knowledge Base containing information on each customer, resolutions so common problems, and other such data. An issue tracking system is similar to a "bug tracker", and often, a software company will sell both, and some bug trackers are capable of being used as an issue tracking system, and vice versa. Consistent use of an issue or Bug Tracking System is considered one of the "hallmarks of a good software team".

This implementation is suitable for a small or medium sized organisation to record and efficiently respond to reported problems / issues.

Where ever possible the development tries to lever the supplied features provided in the Joomla core. This enables the same 'look and feel' as supplied by the standard Joomla components, ensuring user familiarity and making for easier implementation.

Chapter 2. Design

This section describes the design criteria used for the component. It provides the rationale behind the design and helps to understand how the component works.

One of the criteria in the design is to make use of as much of the supplied Joomla functionality and provide familiarity of use with the standard Joomla components.

Planning and Analysis

Effective project and program management is the key to completing on time and within budget. Within every project or program there are always multiple issues that need to be tracked, prioritised, and managed. Additionally there are often additions and improvements made as the project goes through its life cycle.

A company, whatever its size, may have many projects that must be completed on time for the company to be successful and profitable. Missed project deadlines can result in lost revenue. The company's project leads use various methods to track issues, including manually recording statuses in notebooks, organising issues in text documents, and categorising issues by using spreadsheets.

By using a Joomla component it is possible to easily record and track issues in one central location. This approach offers each project/program lead access to just the data they need and makes it easier for management to determine if critical issues are being addressed.

Planning and Project Analysis

The first action is to define the application requirements. These are then used to design a database and an outline that describes how the user interface accepts and presents data.

The project leads establish requirements that define the information that must be tracked, security requirements, data management functions, and how to present data to users. The wish list is enclosed below for completeness; note however that the current implementation upon Joomla does not provide all of these facilities at present.

Gather the Necessary Data

Each project lead may track information slightly differently. It is agreed that the application should include the following information:

- Summary of the issue
- Detailed description of the issue
- Who identified the issue
- The date on which the issue was identified
- Which project/program/product the issue is related to
- Who the issue is assigned to
- Visibility of issue (General or Restricted)
- A current status of the issue
- Priority of the issue
- Target resolution date

- Actual resolution date
- Progress report
- Resolution summary

> **Note**
>
> Not all of these requirements are met with the current release. As demand and circumstances permit, the missing requirements will be added with later releases.

Define Security Requirements

The project leads are concerned about everyone having access to all the information, they agree upon the following access rules:

- Each team member and project lead is only assigned to one project at a time.
- Each team member and project lead must be assigned to a project.
- Managers are never assigned to a specific project.
- Only managers can define and maintain projects and people.
- Everyone can eater new issues.
- Once assigned, only the person assigned or a project lead can change data about the issue.
- Management needs views that summarise the data without access to specific issue details.

Joomla ACL permissions can be configured to achieve a number of these requirements. Others will be implemented as time permits.

Select Data Management Functions

The project leads also determine how information will be entered into the system. For this project, users must be able to:

- Create issues
- Assign issues
- Edit issues
- Create projects
- Maintain projects
- Maintain people information
- Maintain project assignments

Certain of these aspects need to be under control. For example only certain people should be permitted to create projects and assignments, whilst creating issues is much more liberal.

Select Data Presentation Functions

Once the data is entered, users need to view the data. The team decides that users must be able to view the following:

- All issues by project

- Open issues by project
- Overdue issues, by project and for all
- Recently opened issues
- Unassigned issues
- Summary of issues by project, for managers
- Resolved issues by month identified
- Issue resolution dates displayed on a calendar
- Days to Resolve Issues by person

Reports will be developed and added based upon requirements once the product has been released and upon a priority basis.

Define Special Function Requirements

The project leads also determine that the application must support the following special functions:

- Notify people when an issue is assigned to them
- Notify the project lead when any issue becomes overdue

Some of these have not yet been implemented.

Handling of date fields.

Records often contain date/time fields which are used to record the date/time when an event has occurred. These include items such as when an issue is opened or updated. When it is closed and also various auditing performed upon the record. Prior to release1.6.2 dates (and times) entered into the various forms were set up such that they were converted into user time when the form was saved. The details of the specific timezone were not saved thus when redisplayed the time might appear to be incorrect. This was most noticeable with the resolution date field, which when the user was present in an area with a large timezone difference from UTC/GMT could be confusing. Especially since the default audit dates as set up by database triggers were using server local time'. A user would enter the date in the required field which would be as for example '2014-06-05', but because the value was picked up (by a EUROPE/LONDON timezone user) it would be handles internally as if it were '2014-06-05 23:00:00'. For a Latvian user with a timezone of 'EUROPE/RIGA which has an additional time difference of 2 hours (3 if one includes daylight saving time [DST]), the time would appear to be '2014-06-05 21:00:00'.

Even our own system where daylight saving time (DST) was in effect could cause a 1 hour difference in the local time and the actual time. For some areas it would be an even greater apparent time discrepancy. Release 1.6.2 addresses this problem and all date/time values are now stored in UTC time. Then when the values are displayed to the user they are presented in the time in the users own time zone. For guest users the server time zone is used. To aid this conversion a helper routine has been implemented and is described in more detail later in this document.

Cron tasks

The topic of implementing a scheduled activity with Joomla is achieved usually by the use of a daemon named 'cron'. This is primarily due to most(?) Joomla systems running upon a UNIX based operating system. This is not always the situation and there are a number of Windows based systems available, but there is adequate documentation about Windows systems created by Microsoft and it is not necessary to repeat that information. With UNIX systems being so common we have included this section because an understanding of 'cron' is important in its implementation and also in the design of specific

activities that leverage its facilities. Some of this information is also included in the Users Guide but is also included here for completeness.

Cron is basically a time-based job scheduler that runs in the background on UNIX systems. Typically a task requiring repetitive actions to be carried out on a regular (pre-determined) basis would be ideal candidates for using cron.

Issue Tracker release 1.5 introduced a number of tasks that could be automated. The number is likely to increase over time, but initially includes the generation and sending of reports of a) Summary of issue numbers over a specified time period, and b) overdue issues by assignee, which is sent to each assignee. Release 1.5 also enables the automatic raising of issues and optionally updates via email. See below for details of the criteria used for this implementation.

Joomla systems can offer a number of different mechanisms to perform repetitive actions, of which the one to be used is very dependant upon the facilities offered by the hosting supplier.

Pseudo-cron

It is possible to make use of a ‘bespoke’ plugin which can be invoked when someone accesses one or more pages upon a site and performs some predefined tasks. This is really a 'real' cron since there is no real way in which one can guarantee that there would be a site visitor at any predetermine time, and has an unavoidable side effect of slowing the site access for the specific visitor that happen to visit the site at or immediately after the desired time. For this reason we have not implemented such a mechanism, however for some other ‘short’ running tasks it might be suitable, so although not suitable for our implementation can in some circumstances provide a reasonable solution.

Front-end access

This mechanism makes use of the front end of the site and provides a task that may be invoked by a web browser, but more commonly by a utility accessing a web page such as wget or curl. Such tasks typically when invoked in this way do NOT provide progress details.

The front-end access URL's for cron tasks are not usually designed to be run from a normal web browser, but from an unattended cron script, utilising a server side executable as a means of accessing the function. Typically utilities such as wget, curl or lynx are used, which can be thought of as applications which simulates the behaviour of the browser. They typically try to access the cron supplied site URL so that the accessed page actions occur.

Normal web browsers tend to be "impatient". If a web page returns a bunch of redirection headers, the web browser thinks that the web server has had some sort of malfunction and stop loading the page. It will also show some kind of "destination unreachable" message. Remember, these browsers are meant to be used on web pages which are supposed to show some content to a human. This behaviour is normal. Most browsers will quit after they encounter the twentieth page redirect response, which is bound to happen. Using browsers such as Firefox, Internet Explorer, Chrome, Safari, Opera or any other browser are **not** intended to work with the front-end cron based features. They are **NOT** meant to work by design.

Command line utilities, by default, will also give up loading a page after it has been redirected a number of times. For example, wget gives up after 20 redirects, curl does so after 50 redirects. Depending upon the actual task being executed it may be advisable to configure the command line utility with a large number of redirects. The number will depend upon the task itself.

Tip

There are several locations upon the web that provide a free service enabling of scripts supporting CRON? There is a free service from Webcron that provides a simple interface, that we have tested out ourselves. There is also a paid service that Webcron.org provide that fully supports a number of front-end features and is reasonably cheap - you need to spend about 1 Euro for 1000 runs. Just make sure you set up your Webcron [http://webcron.org/] CRON

job time limit to be at least 10% more than the time it takes for the script to execute upon your site. Don't know how much is that? No problem! Just run the script from your site's front end and average out the run times and add about 10% more to the required value.

It should be noted that IF your site uses any redirection component such as sh404SEF (and possibly .htaccess) that the URL specified in the Webcron service should be the 'redirected' URL and NOT the initial URL, otherwise a 301 error is seen.

One feature that is often included is the supply of a 'Secret key' or 'Pass Phrase'. It is a character based string which will allow the CRON job to ascertain that it has the right to request to run is acceptable. It can be thought of as an additional security feature.

Tip

Use only lower- and upper-case alphanumeric characters (0-9, a-z, A-Z) in your secret key. Other characters may need to be manually URL-encoded in the CRON job's command line. This is error prone and can cause the backup to never start even though you'll be quite sure that you have done everything correctly.

Most hosts offer a CPanel of some kind. There will be a section named something like "CRON Jobs", "scheduled tasks" or the like. The help screen they provide should describe how to set up a scheduled job. The missing part would be the command to issue. Simply putting the URL in there is unlikely to work.

Warning

If your host only supports entering a URL in their "CRON" feature, this may not work with most cron scripts. There is no workaround. It is a hard limitation imposed by the host.

Important

Be careful with any caching that may be present upon the web site. If the specific page being accessed is present within the cache it will be delivered to the caller, BUT the underlying actions will NOT be invoked. It is better to disable cache for the specific page.

We wasted a lot of time with tracking down this specific problem, and have not found it mentioned anywhere else on the web.

UNIX System

The following examples assume that the command is referencing a Joomla component named 'com_component' and that the specific cron task to be performed is configured as 'cron.xxxx'. These will be dependant upon the specific task and component being used.

You can use one of the following commands depending on your server:

```
lynx -source "http://www.yourwebsite.com/index.php?
option=com_component#38;task=cron.xxxx&secret=YourSecretKey" > /
dev/null
```

Or

```
wget -O /dev/null -q "http://www.yourwebsite.com/index.php?
option=com_componenttask=cron.xxxx&secret=YourSecretKey" > /dev/
null
```

If you access your website using an HTTPS protocol, you may want to use this command instead:

```
wget --no-check-certificate -O /dev/null
 "http://www.yourwebsite.com/index.php?
```

```
option=com_component&task=cron.xxxx&secret=YourSecretKey" > /dev/
null
```

Some hosting company block the lynx and wget command in which case you should use the curl instead:

```
curl --silent --compressed "http://www.yourwebsite.com/index.php?
option=com_component&task=cron.xxxx&secret=YourSecretKey" > /dev/
null 2>&1
```

or

```
curl -L -v "http://www.yourwebsite.com/index.php?
option=com_component&task=cron.xxxx&secret=YourSecretKey" > /dev/
null 2>&1
```

Depending upon the PATH setting in your session it may be necessary to specify the full path to the executable, and is probably a good idea to specify it as a default anyway.

```
i.e. /usr/bin/wget instead of wget.
```

Any line breaks which may be shown in the above examples are included for formatting clarity only. One should not have a line break in the command line!

Warning

Do not forget to surround the URL in double quotes. If you don't the script will fail. The reason is that the ampersand is also used in UNIX to separate multiple commands within a single command line. If one doesn't use the double quotes at the start and end of the script URL, your host will think that an attempt has been made to run multiple commands and load your site's home page instead of the front-end page URL.

Contact your host; they usually have a nifty help page for all this stuff.

Important

Using a web browser (Internet Explorer, Google Chrome, ...) or wget version 1.10 and earlier may result into an error message concerning the maximum redirections limit being exceeded. This is not a bug. Most network software will stop dealing with a web site after it has redirected the request more than 20 times. This is a safety feature to avoid consuming network resources on misconfigured web sites which have entered an infinite redirection loop. To avoid this problem (if you encounter it, there is the option --max-redirect on wget version 1.11 and above.

Warning

Any ampersands entered in the crontab entries should be written as a single ampersand, not as an HTML entity (&). Failure to do so will result in a 403: Forbidden error message and the task will not run. This is not a bug, it's the way wget works.

Wget

GNU Wget is a free software package for retrieving files using HTTP, HTTPS and FTP, the most widely-used Internet protocols. It is a non-interactive command line tool, so it may easily be called from scripts, cron jobs, terminals without X-Windows support, etc.

GNU Wget has many features to make retrieving large files or mirroring entire web or FTP sites easy, including:

- Can resume aborted downloads, using REST and RANGE

- Can use filename wild cards and recursively mirror directories
- NLS-based message files for many different languages
- Optionally converts absolute links in downloaded documents to relative, so that downloaded documents may link to each other locally
- Runs on most UNIX-like operating systems as well as Microsoft Windows
- Supports HTTP proxies
- Supports HTTP cookies
- Supports persistent HTTP connections
- Unattended / background operation
- Uses local file timestamps to determine whether documents need to be re-downloaded when mirroring
- GNU Wget is distributed under the GNU General Public License.

Downloading GNU Wget

If GNU Wget is not installed upon your private system, the source code can be found on http://ftp.gnu.org/gnu/wget/ [via http] [http://www.gnu.org] and ftp://ftp.gnu.org/gnu/wget/ [via FTP] [http://www.gnu.org]. It can also be found on one of the GNU FTP mirrors. For more download options, see the download information on the Wget Wiki [http://www.gnu.org].

CURL command

cURL is a software package which consists of command line tool and a library for transferring data using URL syntax. It is a standard command present on most (all?) UNIX systems.

cURL supports various protocols like, DICT, FILE, FTP, FTPS, Gopher, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMTP, SMTPS, Telnet and TFTP.

Command Line Alternative

If you have access a the command-line version of PHP, scheduled tasks can use a different, probably better - and faster - way of scheduling.

In this situation a script which can be run from the command-line PHP interface (PHP CLI) is used. This doesn't require the front-end in order to work; it is self-contained, and can run even if the web server is down!

In order to schedule such a task, one has to use the following command line to the host's CRON interface:

```
/usr/local/bin/php /home/USER/webroot/cli/cron.php
```

where /usr/local/bin/php is the path to the site PHP CLI executable, /home/USER/webroot is the absolute path to your web site's root, and cron.php is the name of your script located in the 'cli' subdirectory. This information can be supplied by your host provider.

Important

We cannot know the path to your PHP CLI executable or the absolute path to your web site's root. This is the kind of information that only the people who have set up your server (your host) can possibly know.

Note

The use of the PHP CGI executable is not suitable. To check which version is available run the PHP command with the --v option which will display the executable specifics.

Sometimes hosts will give you the path to the PHP CGI binary. This will NOT work with CLI scripts. CGI (Common Gateway Interface) and CLI (Command Line Interface) are two completely separate things. The former is designed to run scripts which will be served as web pages. The latter is designed to run command line scripts, usually for use with CRON.

Securing the script

We earlier mentioned the use of a secret key to be provided when the script is run. There is also another alternative for PHP-CLI scripts. This is (if using Apache as your web server - most do) is to add some rules to your .htaccess file that prohibits anyone other than the local machine from running the script.

Note

For the following rules to work, you must have an Apache web server with mod_rewrite enabled.

For example: If the file you want to run is called 'script_name', you could add the following to your .htaccess file to ensure that nobody else can run the script:

```
RewriteEngine On
RewriteCond %{REMOTE_ADDR} !^127\.0\.0\.0
RewriteCond %{REMOTE_ADDR} !^x\.x\.x\.x
RewriteCond %{QUERY_STRING} script_name [NC]
RewriteRule ^(.*)$ index.php [F,L]
```

...where x.x.x.x is the actual IP address of your server (if you have an SSL certificate, this will be unique to your hosting account - otherwise the IP address might be shared with other hosting accounts on the same server). You can find the IP address of your domain by running a ping test against your domain name (in Windows, go to Start->Run, type in 'cmd' and press enter, then type 'ping www.mydomain.com' and press enter). If you wanted to allow yourself to run the script from a browser (useful for testing), you could add your own IP address as well. Please note, you must add a backslash before each dot of your IP address.

Cron Basics

Cron is a daemon that executes commands at specified intervals. These commands are called "cron jobs." The Cron daemon is available on Unix, Linux and Mac servers. Windows servers use a Scheduled Task to execute commands.

Cron Job Command Basics

The way you set up a CRON job on your server depends on what software you are using to manage your hosting account. Some web hosting control panels (such as cPanel or Plesk) offer a graphical user interface (GUI) which makes it much easier to set up the scheduled task. If you do not have a GUI available, you will have to enter the necessary commands using Telnet or SSH.

A quick search on the Internet for 'cron' followed by the name of your web control panel should yield plenty of articles that tell you how to physically set up a new CRON job, but it is usually fairly self-explanatory if using something like cPanel or Plesk.

In any case, a CRON job is made up of 2 main elements: a command to be run, and an interval definition (i.e. something that tells it how often to run). You might also be asked to provide an e-mail address to which any error messages or other output can be sent.

The command to run is made up of 4 parts:

- An executable (i.e. a program that is capable of running PHP scripts)
- One or more configuration options (specific to the executable)
- An output path (where to put any output from the script)
- A URL (the script to be called)

In the following example, the crontab command shown below will activate the cron tasks automatically on the hour:

```
0 * * * * wget -O - -q -t 1 "http://www.yourwebsite.com/index.php?
option=com_component&task=cron.xxxx&secret=YourSecretKey"
```

The 0 * * * * part in the example above is the time when the cron job is going to happen. The following schematic tries to explain the general crontab syntax:

```
# +---------------- minute (0 - 59)
# | +-------------- hour (0 - 23)
# | | +---------- day of month (1 - 31)
# | | | +------- month (1 - 12)
# | | | | +---- day of week (0 - 7) (Sunday=0 or 7)
# | | | | |
* * * * * command to be executed
```

The frequency with which one wishes a specific script to run, should be carefully evaluated. Running it too often when there is nothing much to do is probably wasteful of system resources. On the other hand not running it frequently enough means that the expected actions are not 'up to date'. Generally once an hour is more than adequate in our tests, especially since we used to manually perform the task twice or three times a day.

Trouble shooting cron

If your cron task is not working, you should:

- Make sure your website can be triggered via your browser: Click on your cron URL. That will trigger it manually via your browser so you will see if this URL can be reached or not.
- If you use a third-party SEF component, your website may redirect the current page to the SEF one which will break the own cron service. So you can either configure your SEF component to not transform non-SEF into SEF URLs or directly write your SEF cron URL on the cron interface.
- Make sure your website can be triggered by the outside: If you have an .htaccess with password protection, your website will obviously not accept to be triggered by an anonymous user.
- If you use a security component on your Joomla website and use a remote cron facility, this component might block the cron task. If you are using RSFirewall, you could try disabling the DoS protection from the Components => RSFirewall => Configuration => Active scanner configuration options.
- Sometimes cron jobs can't trigger an URL including a "&" (see above) or can only trigger a script (instead of an URL). In that case, you should create a file with a suitable name at the root of your website including the code below:

```
<?php
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL,'http://www.yourwebsite.com/
index.php?
option=com_component&task=cron.xxxx&secret=YourSecretKey');
```

```
curl_setopt($ch, CURLOPT_RETURNTRANSFER,1);
curl_setopt($ch, CURLOPT_TIMEOUT, 10);
curl_setopt($ch, CURLOPT_FOLLOWLOCATION,true);
curl_setopt($ch, CURLOPT_AUTOREFERER,true);
echo curl_exec($ch);
curl_close($ch);
```

Then configure your cron to trigger that script.

Please don't forget to replace 'yourwebsite.com' by the URL of your own website, and also the correct component name, task parameters etc..

- Make sure that there is no system cache storing up your pages accessed by a front end based cron task. If there is the cron task will only work when the cache entries expire, which will depend upon the cache settings.
- We have seen circumstances where a cron job that executes fine in PHP with MySQL statements and echo statements, but when mail functions are added it fails with this error:

  ```
  sh: -t: command not found
  ```

  The cron daemon can be picky about paths, and in these cases it is worth using absolute physical paths instead of absolute share paths. This is mainly because the cron daemon is active before the shares are up, and why it is probably unable to process the commands/enumerate the path during the initialisation phase.

  It is also worth while adding the option to direct PHP to the initialisation file to be used, with the '-c' option, to prevent it running from shell context only. This will include the proper path to the sendmail alias and other information such as locales etc.

  ```
  * * * * * /usr/local/apache/bin/php -c /php.ini /
  path_to_cron_php_file/cron_file.php 2> /path_to_log_file/
  cron_error.txt
  ```

Joomla Warning Messages

The following messages coming from the Joomla core may occasionally be seen, and can safely be ignored:

```
Warning: session_start(): Cannot send session cookie -
 headers already sent by (output started at /...../cli/
issuetracker_imap.php:131) in /...../libraries/joomla/session/
session.php on line 532
```

The above message is coming from the JTable Joomla code, when the composed new issue record is stored into the database. It seems to occur on the first (initial) cal to the JTable store routine only.

```
Warning: mysqli_ping(): Couldn't fetch mysqli in /...../libraries/
joomla/database/database/mysqli.php on line 190
```

The above message is seen when exiting the cron task. The reason is not currently known, as we have not yet investigated the sequence of actions which cause the message to be generated.

Automatic creation and update of issue via email.

This section looks specifically at the areas/concerns around using email to both create/open new issues and also using email to update existing issues. This feature makes use of the Cron facility discussed earlier, and because it is such a involved task is worthy of a section in its own right.

Design Criteria

There are two situations where email could be useful. The first is for receiving emails about new issues and the second is for receiving updates for already existing issues.

The following design criteria is deemed necessary in the email issues functionality. They are included in no specific order of priority.

- Ensure that there is a clear record of the person who is reporting the email. This effectively means relying upon the senders email address.
- Need to ensure that the Issue Tracker is not open to spam in the received email.
- Implementing a mechanism to try and ensure that the email FROM address is genuine.
- Support different types of email formats and protocols (e.g. POP3, IMAP).
- Ensure that the installation retains control of the issues in their system (i.e. make sure no one can report a single issue that has undesirable links). An installation doesn't want their customers/users to login to their issues and find porn and other undesirable related issues.
- The reporting by email feature should be disabled by default.
- Use one dummy email account to associate with all received emails, and use the originating email address as part of the criteria for determining who is raising/updating the issue.
- When a user reports an issue via email, a confirmation email should be sent to the originating email address. This should be independent of whether we set the reporter as a unregistered or a registered account.
- When an email is received for an existing or a new issue, the assignee should be sent an email, except if the assignee has indicated that they should NOT receive emails.
- When an update to an issue is received, an email should only be sent if the user has a flag indicating that an email should be sent.
- Attachments to emails should be added as attachments in the Issue Tracker component.
- Must be able to handle mime/multi-part emails.
- Replies to existing issues to be added to the ongoing progress field.
- Inline replies are not supported. Hence, the additional details should be extracted from the top of the message
- Look for and use a “——please reply above this line——” (or administrator define string) in the incoming email for updated issue emails. Add such a string to the outgoing notification email.
- Provide an option to disallow/allow unknown (public/guest) users from submitting issues
- Unknown users should require a reply validation hash to prevent spamming of the issue tracker.
- Look up a Issue Tracker account based on an email. If an account if found use it, otherwise, there should be an option to reject the email or auto-create an unregistered account.
- Ensure email addresses are unique. i.e. Multiple users cannot have the same email address.
- One idea to avoid spam is to send back an email that requests a certain interaction that the user needs to do. This would probably stop bots but not manual spamming.

Implementation of email raise/update issues

It is possible to use a cron task to fetch emails, from a specific address, and to use the contents to create or update an issue upon the site automatically.

The component needs to be configured with the details of the mail server from which the emails are to be fetched. Once configured the task will make contact with the email server, and read all of the 'UNSEEN' messages present for the connection user.

Each message is then processed. The first step is to inspect the message to determine whether this is a 'new' issue being reported or an 'update' to an already existing issue. This is achieved by checking the received email to see if there are any 'hidden' header tags which are recognised by Issue Tracker.

Note

All email notifications sent out by Issue Tracker include two specific custom header tags that identify the issue (about which the email is concerned), by its database unique identifier and it issue number. If the sent messages are used to create a reply then these fields can be checked for when a reply is made. Unfortunately a lot of email clients strip out the headers when they reply so that means they are not present for us to check, and hence can not be relied upon.

If the 'hidden' header tags are not present then the message subject (title) is inspected for a specific issue string. Depending upon the strength of the checking specified in the component options, the specific string would be '[Issue: xxxxxxxxxx]' or '[xxxxxxxxxx]' where xxxxxxxxxx represents the issue number. If neither of these pieces of information are present then the message is assumed to represent a new problem report.

Note

Consideration was given to implementing checks for other information in the email header for a response, but unfortunately different email clients may add "re" "RE" "Re" "Antwort" "AW" "?p" "?p??t?s?" or some other localised "reply" prefix OR SUFFIX in the message subject. It was considered virtually impossible to keep a track on all of the possibilities and would present a continual coding nightmare.

To complicate the problem there is always the situation where people change the subject of the email so one is then into a complex parsing exercise. For this reason no checks are performed to match the Email Subject title other than for the string mentioned above.

The message is also checked, using the currently implements spam checks. i.e. Are there any 'prohibited' words, the number of included web links etc. Optionally Akismet is also supported for checking of Spam.

Note

A lot of the commercial ticket/issue/problem reporting systems –e.g. GitHub– work based on a customised email address per problem. Unfortunately this is not an option for a Joomla! component. The other technique commonly used is the thread ID in the mail server which again is not an option.

This leaves only with possible hidden email header entries and a reply above line, both of which are implemented in the Issue Tracker feature.

New issue emails

When the message is determined to be a new 'problem report', some additional checks are made.

The email address of the sender is then analysed. If the email address is not currently known by the system, and we are accepting emails from 'guest' users then optionally a new 'unregistered' user is created in the component.

The message subject is stored as the 'issue summary' for the reported problem.

The contents of the message text are stored in plain (not HTML) format in the description field of the issue.

A confirmation email is sent to the sender and also to the (default) assignee for the issue when the issue has been saved in the database.

Updates for existing issues.

To be a (possible) reply to an existing issue,the header tags and/or the pseudo header tags are present.

Checks are then made to ensure that the 'extracted' issue details from the message subject or header tags, together with the email address of the sender match with the details recorded in the existing issue within the database, IF it exists. If the 'requested' issue does not exists, or is 'CLOSED' or is not identified as being originally identified as by the message sender, the message is rejected.

There are options to control the strength of the email reply detection. Three options are available:

- Strict: Accepts only the exact format that email notification uses. Which means "anything [Issue: xxxxxxxxxx] anything here". The full text is required, and the issue id has to be the exact length (def 10 chars). In addition the custom header fields have to be present in the reply. Spaces before and after the issue identifier are optional.
- Balanced: As Strict only the custom header fields may or may not be present. Spaces before and after the issue identifier are optional.
- Relaxed: Accepts email notification formatted subjects but also [xxxxxxxxxx] in the subject line. Custom header fields may or may not be present. Spaces before and after the issue identifier are optional.

A lot of email clients (e.g. Thunderbird) by default *appends* replies to the original text, and it is not desirable that this information should be added to existing issues, since it will cause un-necessary duplication. For that reason the message should include a 'reply above this line' (the precise text is a configuration option) string. If a user makes a mistake and does not enter text in the correct manner an empty reply may be generated, and in this situation the message should be discarded.

The message contents are appended to the issue progress field, immediately preceded by a line separator and the date and sender Name (extracted from the message) in HTML format.

Note

The script only fetched 'UNSEEN' messages from the mail server. In the event of an error being encountered upon processing the message, i.e. An issue may be in the state of being updated (edited), in which case the issue could not be updated, then the message would be left upon the mail server, even if it is configured to remove messages after processing.

Tip

The act of retrieving the email message from the server will often cause the email server to mark the message as 'SEEN' or 'READ', which means that if the next 'cron script' run tries to retrieve the message it will not be 'fetched'. To reprocess a message it is necessary to enter the email server and mark the desired messages as 'UNSEEN' or 'UNREAD', in which case they will be picked up by the 'cron script' and re-processed upon its next run.

Caution

If the option to delete processed emails is set, then all emails that have been successfully processed will be automatically removed from the email server and will be unavailable for re-processing.

Custom Fields

The aim was to produce a feature known as custom fields within the Issue Tracker component. This change was to be incorporated within the combined (Joomla 2.5/ 3.x branch) which at the time was the release 1.5 branch. This is a 1.6 feature.

The requirement is to be enable the addition of additional informational fields in the Issues details.

- These additional fields need to be of different forms such as a drop down, radio, integer, link, URL, text fields etc.
- These fields are to be populated upon the edit of the issue and stored within the database 'linked' to the specific issue and visible and editable from the front and back ends of the site.

Possible implementations

There are two possible ways in which this could have been addressed. The first would be to add additional 'customisable' fields within the issue itself.

Additional table fields

The site administrator could then tailor these fields themselves. However the downside of this is not knowing how many fields to provide and the specific field types required. It is almost certain that if one were to provide say 4 customisable text fields, someone would want five etc. This is one of the reasons why this is probably not the way to go!

New custom tables

A better and more extensible solution is to create a single extra table column and store all of the customisable fields within this one 'column' in JSON format.

This is the preferred solution and the one chosen to implement.

There are two additional tables required. The first is the 'custom field' table where the additional fields criteria are specified.

This includes such criteria as:

- field name, field type (i.e. Drop down, Radio, text etc.). The default value the field is to take if not specified. (Null would be acceptable).
- The required optional values if it were a check box or a option list. i.e.. "apple, pear, banana"
- The tool tip text to aid the person entering the details.
- The name of the group to which the custom field applies. (See below).
- Validation rules that might apply to the field?
- Published state of the field (whether to show it or not!)
- Access rules that might apply to the field. (Who to show it to.)

The second table is the 'field group' table, which contains the details of the various 'custom groups' to which a 'custom field' would apply.

- A custom group applies to a 'project', and potentially a project could have multiple custom groups. The initial implementation will be for a single custom group per project. Sub-projects would inherit the 'custom groups' of the parent.

- A custom field would have a relative position (ordering column) within a custom group which would be used to determine display order.

Each of these tables has their own controllers, models and views (list and singular) on the back end for the definition and modification of the field and the field group.

The issue class method has changes to the 'save' and 'get' method to handle the new 'custom field column'. This applies to the front end and back end methods.

The projects and issues table have an additional field adding to provide linkage to the custom field and custom field group.

The back end project list display is changed to display the 'custom group(s)'. Front end possibly project list probably doesn't need changing.

The front and back end Issue list display are in the initial implementation unchanged with the custom fields only being shown in the single item views,. This may or may not be a possible future enhancement to the feature? [It would require breaking out the JSON column to extract the required 'custom field' to add to the list display.]

New text strings are created to handle the additional table fields and form display fields.

The displays have been modified to enable the display of the 'custom group' fields. There again are two ways of achieving this:

- The first would be to let the site administrator define a 'template override' which would enable them to code where they wanted the field(s) to be displayed.

- The second would be to provide an additional parameter on the 'custom group' definition to enable them to specify 'after' which field, the custom group should be displayed. This is not as 'customisable' as the template override option, but it does not prevent a site administrator using the template override option. This is the method implemented although the 'ordering column is used to define the order in which the various fields are displayed within a group, rather than provide a new option..

The issue display effectively has a number of 'visible blocks'. i.e. Summary (Title) and Description, a Progress block, and a Resolution block. The custom group effectively becomes an addition display block.

The former 'product details' request on the front end form has been migrated to make use of the custom fields feature, thus enabling the actual usage to be more easily tested and controlled going forward, as well as removing some unnecessary duplication of coding.

There are changes to add additional field(s) added to the issue and project tables to accommodate these Custom Field tables. The issues table accepts one additional field named 'custom_fields' and the projects table has an additional field named 'customFieldsGroup'.

The display of the custom field types makes a call to the a javascript routine which controls the fields shown upon the screen. Some of the field types have their own unique display of fields, others are adaptable with the ability to add additional criteria. Examples of the latter are the 'select', 'multi-select and 'radio button' field types where a list of possible options can be created. These are added by javascript which runs in the client browser.

Also implemented is the use of AJAX calls within the Issue display in the back end and the 'Raise an Issue form' in the front end which is invoked when the associated Project is selected by the raiser/editor and/or changed from that which is currently defined or is the default.

Progress History

The update of the progress history is intended to make it more logical to view the actions performed upon an issue in a chronological order. This is achieved but having a single 'progress' table in which

is stored the progress records for all issues. Each record in the table relates to one specific activity performed. Each record will obviously possess a unique identifier but also an incrementing counter for each issue.

The records are shown as a separate panel in the issue display (front end and back end) in a tabular format similar to that already present for issue attachments.

The intent is that only the 'Issue administrators' will be able to edit 'old' progress records, and a link will be provided upon each progress record for the issue administrator to click upon to edit the record. After editing the record the intent is to return to the issue display itself. This may change in later revisions subject to experience.

Each progress record has controls upon who can view that specific record. The 'access' field will enable only users within the assigned group to view the specific record. This will enable 'issue administrators' or 'staff' members to be able add 'private notes' to an issue recorded in the progress history, in this way avoiding the need to implement a separate 'notes' feature.

This also changes the former storing of 'additional' information entered into an issue by the raiser on the front end such that this is also entered and stored as a 'progress' record, instead of as formerly being appended to the end of the issue description.

An additional feature to add details of emails sent to a progress record is also under consideration and may be implemented in a later release.

For simplicity the existing front end form where one enters new progress information is retained, and upon saving the data stored in the new progress table.

There is no facility (or currently any need?) to edit previous progress records from the front end. They can be edited in the back end. Changing (or deleting) of a progress record does not cause the sending of an update notification email.

When an issue is trashed the individual progress records are not impacted, and it is only when the issue record is 'deleted' (i.e. the trash is 'emptied') that the progress records are deleted. The progress records can be edited by an issue administrator (in the back end), and the published state and access modified as required. These fields can also be specified when a progress record is created, with the defaults being that the progress record is unpublished and the access set to 'registered' users.

An individual progress record can have its own 'access control group', an individual published state, and also an individual privacy setting. This enables the site to use the progress records in any way they deem suitable. So for example a site might use the progress access control group setting such that the issue identifier never sees certain progress records, which are not in their assigned access groups. In this way internal update actions might be available to different support staff associated with an issue and only viewable by those support staff.

Use of the individual 'privacy' setting would enable only the issue identifier to 'see' progress records marked as private, so in the event of an issue being published the general public would not see these 'private' marked progress records, even if the progress record itself is 'published'.

Release 1.6.4 introduced a change in the email sending of progress records such that only the 'last' progress record is attached to a 'user update' email. If the last progress record is not within the users assigned groups then no update email would be generated.

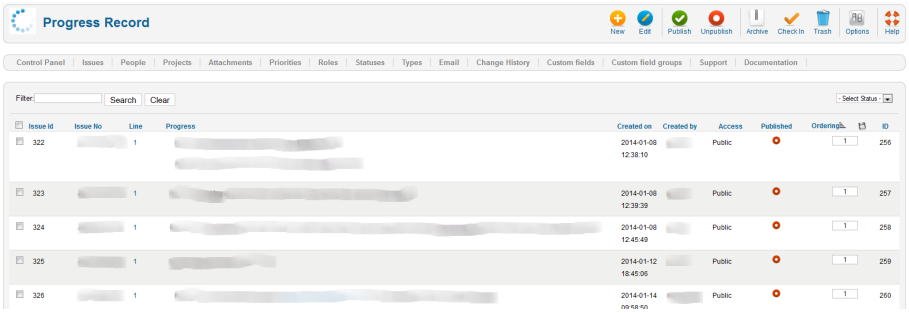
Pactions View

With the design of the partitions table and its implementation into the component it was decided to create a hidden view in the back end of the component. As is common with Joomla views back end displays are generally of a form where there is a 'list' of items and then individual 'item' displays accessed from the 'list' display. Progress records do not quite fall into this 'category'. This is because any given Issue record may have many different progress records associated with it. These 'historic

progress records' are displayed as a small grid as part of the Issue detail display. These 'small progress grid records' can be edited in the back end, but no mechanism is given for removing the records individually. This is mainly by design, with the reason being that each progress record builds up the 'historic knowledge base' of information about issues/problems encountered and it is not (usually) desirable to lose such information. If the Issue record is deleted then all associated progress records (along with all attached files) are automatically removed at the same time.

We recognised that occasionally it may be desirable to remove some possibly contentious progress records, and that is why we created the additional list display available in the back end of the site. This 'list' display is of all of the progress records. It provides a sortable and filterable list of every entered progress record together with the ability to edit them, delete them, publish/unpublish, etc. All of the classic tasks are familiar to the existing 'list' displays. Since this is a display that is not likely to be required very often (if at all) there is no immediate link to the display. Instead it is necessary to enter the URL directly in the browser address bar. The name of the view is 'pactions' (progress actions). It is a fully functioning view as explained above.

Figure 2.1. Back End Pactions view



One benefit of the view is that it allows a quick and easy way to view progress records and if necessary edit them to correct layout or grammatical errors in the text.

Issue Reports

There is a requirement to generate some graphical reports for the component. This is a future release feature, possibly using Google Open Graph API, or alternatively JQuery Graphics UI. The choice will depend upon the features and ease of implementation of each. The preferred choice (currently) looks like being the JQuery UI since that is the direction that Joomla is moving towards and the libraries may well be implemented by the time the coding starts taking place.

Important

This is a feature that is targeted towards a 1.7 or 1.8 release.

Report Export

Release 1.6.4 introduced the ability to generate a CSV file that can be loaded into a spreadsheet to allow a site to create their own reports based upon the information exported in the CSV file.

Note

This is a Joomla 3 feature only and is not implemented for a Joomla 2.5 site.

The feature is accessed in the back end Issue List view, via a small icon on the icon bar. When pressed it will generate a popup modal window where the administrator can choose a suitable name for the export file. The contents of the exported file will be based upon the contents displayed in the list view. This means that any filters that might have been applied in the list view will also be used in

the extracted file. The choice of file name will be based upon the type of filter that may have been applied. If more than one filter has been applied then the first is usually assumed. If an attempt is made to choose a filename upon a filter that is not in use then a default file name based upon the 'site name' will be used instead.

The contents of the generated CSV file have been populated such that all references to sub tables have been resolved and the reference fields contain the referenced values.

Some of the fields are actually a combination of different possible tables. So for example the 'custom fields' are all contained within a single 'spreadsheet' cell for the specific issue. Likewise the progress details are all extracted from a separate database table and are presented in JSON format with a single cell for a specific issue. It is left to the spreadsheet user to use the information in any way they deem suitable.

Multi-Lingual support

Important

This feature is unlikely to be implemented in the short term, but is a longer term feature, possibly in release 2.0, for which no release date has been specified.

Likely changes include:

- An extra language field on the subsidiary tables and then in the drop down picker, filter out the strings we do not need.
- Modification of all list views to add a language column, plus sort and filter abilities.
- Each table to have an additional identifier so that the same 'type' is any language is connected. i.e A defect in one language is the same type as a defect in another language.
- Need to decide what language back end displays in (en-GB) ?
- This is going to be a major change impacting all over the place.
- First step is to create the additional language field in all our tables.
- At the same time need an extra id fields so that a specific 'type' in different languages can be associated.
- An index on the language field will also help.
- Then add extra code in the model and views.
- Modify select to include language selection for site. Need to think how to make the upgrade easy as well.
- Can we assume that there is only one language set in use, probably yes so make that the default and then copy it to make it the current language default.

Import/Export Facility

There is a requirement to be able to extract the information upon a site, possibly as a CSV format file, suitable for transferring and reloading into another site. There are some considerations as to the versions of both the source and destination components.

Release 1.6.4 introduced the ability to generate a CSV file that can be loaded into a spreadsheet to allow a site to create their own reports based upon the information exported in the CSV file. This is was intended to be a feature that can be used to load the details back into Issue Tracker. The details

have been extracted such that the fields are text items, except where they are specifically something like a date field.

Important

The import/export feature is targeted towards a 1.8 or 1.9 release.

Alternative Alias identifiers

One request that came from the forum was the ability to have an alternative format for the Issue Alias, which is used to identify the issue when it is raised.

By design the alias is composed of a ten character string of random characters with the leading character used to identify the source of the issue, and whether it was raised in the front end or the back end of the site. This can be considered a little over kill perhaps and can be difficult to tell someone the value over the phone without perhaps causing unnecessary problems.

For these reasons there are two alternative alias formats implemented. These all retain the same ten character string length but provide a simpler mechanism.

The first consists of the same leading character prefix followed by the unique id of the issue preceded by zeros.

The second format is the unique issue identifier (id) but this time followed by null characters.

There is one complication with the new formats and that is that they contain the unique issue identifier which means that when an issue is created the alias is not known, until the issue is saved. Hence there is a need to 're-read' the issue from the database to set the alias.

It is possible to mix the formats, but only one can be in use at any one time. So for example one may start creating issues in one format and then decide to change to one of the alternatives. Existing issues will remain unchanged and continue to use the 'old' format but any new issues created will assume the new format.

Note

This may create what may be considered a confusing URL in that the issue identifier may appear to occur twice in the URL, where as in actual fact it is two fields (the id and the alias) which both happen to have the same value.

Ability to associate types with projects.

Ability to link 'types' to 'projects', so when you select a project in the FE, the types field only displays the linked types, not the whole list. There should be an option to enable this linking, because people may want all types to be available to all projects, and one doesn't want to have to duplicate the list of issue types for each project!

Issue types, such as 'Defect', 'Enhancement', 'Documentation' etc. currently have no specific association and can be chosen for any project. To implement this feature we would require additional linkage in the 'create issue form' such that the list is generated based upon the selected project. This could be done with some Javascript that would make an Ajax call to the system to retrieve the 'new' list of possible issue types. We would therefore need to implement a new controller task to return to the browser the available options. In addition we would need, on the server the 'association list' which would relate each 'type' to one or more projects. This sounds like an ideal candidate for a database link table. The table would consist of a couple of fields, a project field which is a 'link' to the projects table, and a types field which is a link to the 'types' table. There would be one entry in the table for each type that is associated with a project. There would need to be a back end screen to enable the admin

to be able to create the association. Not seen such an idea on any other Joomla component before so it could be interesting to implement.

Another idea could be that each project has an additional (JSON) field that would contain the list of issue types that are applicable for that specific project. We could have a multi select drop down list in the back end for each project that the administrator would select. The default would be that all types are available. In the front end the drop down list selection for the issue type would have to have an additional parameter such that the project id is passed to the drop down selector code so that the type information for that project could be selected.

The final implementation (release 1.6.5) is as follows:

There is an addition field named itypes added to the it_projects table. This is accessed in the back end Projects view, and requires that we add the new field to the itprojects.xml model file. A new model field file named issuetrackertypes.php is created which is used to provide the possible entries from the it_types table. The project view itself enables the administrator to add all the required 'issue types' that are applicable for the project. In the back end projects model class there are two changes required. The first is in the project save routine where we JSON encode the returned values and store them in the table field. The second is in the get method where the record is fetched from the database and the issue types field is JSON decoded.

There are a couple of language strings to provide the table label and description.

The changes in the front end are that we need a new Javascript file that will call the server and populate the select options for the type field. This requires a controller task that is called by the Javascript to return the determined values for the specific project. The javascript is initiated by the user choosing a new project in the front end from the dropdown project list. This is coded in the model field issuetrackerprojectfe.php file. There is already a call in the file to invoke the display of any Custom fields for the project, and this is just an additional call.

There are specific checks such that it if the project 'itypes' field is empty of contains the entry of '0', that the defined types for the component 'default project' file is used. An additional check is that the 'default type' is marked as the selected value providing that default is one of the types acceptable for the project chosen.

New Create Issue Button on List Views.

Button to display/hide the new issue form on the same page as the issue list (when a user is logged in), either below the list or above it (option?)

To achieve this would require using one or more templates which would be pulled in by the issue list display. Using a template means that we can optionally place it either before or after the existing list display.

We may need to look at the 'form' again to see if we need to split that down further to minimise the need for 'duplication' in our coding.

SMS Integration

This was raised as a possible enhancement in the Forum and is being developed. (the initial release is for 1.6.5) The following is an outline of the implemented and the possible drawbacks.

There are a large number of different SMS providers available around the world and it would not be worthwhile providing an interface to them all especially in a 'free' product such as Issue Tracker. Instead a better approach is to implement an interface to an existing Joomla SMS component using an API and thus pass the message along to the SMS component to forward to the relevant SMS provider.

We did not have any experience with any of the current Joomla SMS components but from reading around it was suggested that the preferred component might well be AcySMS. As of October 2014

Acyba (component authors) did not provide an API but they did have a simple PHP file which it was suggested might be used.

It has to be remembered that using any SMS mechanism will involve some extra element of cost as the SMS providers will expect payment for delivery of the message to the recipient. This varies depending upon the SMS provider used, but as of October 2014, Clickatell (one of the global providers) charge about 2.7p per message. There are also costs in our developing the interface to ensure that it would work, and also probably involve purchasing the AcySMS component from Acyba for our development work. It is not expected that we would at any time in the near future have any requirement to use an SMS interface for our own purposes, so such a development would be solely as a result of some form of donation, to cover our expected costs.

The following changes are required.

- Change the Joomla user interface such that the additional profile option to force(?) the user to supply a suitable telephone number that would be used for the sending of any message. This could be achieved by using the User Profile plugin (Joomla standard) but it doesn't really validate the phone number provided.

  In the first implementation this will not be provided. Instead it will be necessary for the system admin to manually add the required phone detaiils for the users in the IT People table. This will ensure that the correct number (and format) is provided. The expected format is that expected by AcySMS which includes a leading '+' and teh country code, followed by the actual users phone number.

  We would look to implement some form of checking o number submission at some time in the future.

- Modify the system plugin such that the information is also synchronised with the it_people table.

  See above item for more information. This will not be available in the 1.6.5 version.

- Modify the it_people table to accept the phone details, and also a new field such as to control whether the SMS interface is to be invoked for this specific user.

  A new phone number field is provided in the IT People table along with a flag to indicate SMS messages are acceptable. This is implemented in release 1.6.5.

- Modify the front end issue creation such that the raiser is provided with the ability to specify a telephone number. Where the person raising the issue is a guest (public) user then we need to capture the telephone number. This is either saved with the user details in the it_people table for an unregistered user or it is stored in the progress field for the issue if unregistered users are not enabled in the component.

  Not implemented in release 1.6.5. There is a question as to whether this is a desirable feature or not? The question is whether we can trust the guest user supplied phone number without requiring the guest user to validate the number.

- New component option to specify whether SMS integration is required.

  The email component configuration has been changed to be 'messages', which will cover both Email and SMS. Each of the various options instead of having a Yes/No option has been modified to have No/Email/SMS/Both options.

  An extra field to select the SMS sender provide has been added, which will be empty if AcySMS is not installed, along with a parameter to specify the maximum length of an SMS message. Any generated message with a length exceeding this value is truncated. Most (all?) SMS providers set some limit upon the length of an SMS message. A default of 160 characters is defined.

- Modify the method involved with the sending of email notification such that SMS are sent instead of or as well as the email notification.

The message sending method has been modified to accept the four possible messaging options (described above).

- Three new 'message' templates have been provided for SMS usage: sms_create, sms_update and sms_close. These are required due to the 'restricted' size of a SMS message, which means that the supplied email templates are not suitable. There is no distinction between user, assignee, or admin usage, as the message is likely to be terse.

- Implement the Acyba code to invoke the interface to the AcySMS component.

  A new method as been written to act as the interface between Issue Tracker and AcySMS.

- A mechanism for testing is required. AcySMS provide a 'test' interface, which effectively does nothing, but it does have a randomly generated failure mechanism. This enables the testing of, the display of errors that may be generated from the SMS provider. All of our development and testing has made use of the 'test' interface, with only the final test prior to the release being upon a 'real' interface.

  The interface checks for various settings and displays messages in the component log to assist administrators in determining where problem may be encountered.

- Do we require any interaction with the AcySMS component to ensure that the message is send successfully(?) or do we just have to assume that it went successfully?

  Depending upon the SMS provider used, there may or may not be a delivery report available from the SMS provider which would be visible in the AcySMS back end. Our testing using Clickatell certainly was visible,although there was some initial confusion as to how this was configured within Clickatell.

The link to the Acyba Developer documentation is as follows: https://www.acyba.com/acysms/387-acysms-developer-documentation.html

It consists (as of October 2014) only of the following piece of information which is reproduced below for convenience.

```
if ( !include_once ( rtrim(JPATH_ADMINISTRATOR,DS) .DS.
 'components' .DS. 'com_acysms' .DS. 'helpers' .DS.
 'helper.php' ) ) {
 echo 'This code can not work without the AcySMS Component';
 return false;
 }

$messageClass = ACYSMS::get('class.message');
$senderProfileClass = ACYSMS::get('class.senderprofile');
$message = $messageClass->get($message_id);

if(empty($message->message_senderprofile_id)){
    echo 'No sender profile found with this ID';
    return ;
}

$gateway = $senderProfileClass->getGateway($message-
>message_senderprofile_id);
if(!$gateway->open()){
    return;
}

$phoneHelper = ACYSMS::get('helper.phone');
```

```
$phone = $phoneHelper->getValidNum($receiver->receiver_phone);
if(!$phone){
    echo 'Invalid phone number';
    return;
}

$status = $gateway->send($message->message_body,$phone);
$gateway->close();
```

The above code was used as the basis of the interface with a few extra checks implemented along the way.

One thing to note is that within AcySMS it generally is not shown when messages are dispatched, apart from when the SMS provider provides a delivery report. Not all SMS providers do, so there is no easy way to determine whether a message has been dispatched with out such a report.

Possible enhancements include:

- Make use of 'canned' messages held within AcySMS and just pass substitution fields (tags) to be placed in the AcySMS messages. This is not something that is in the initial 1.6.5 release. No work has been performed to determine how easy or difficult this may be to implement.
- Accept entry of phone number on Issue creation for guest users, so that they may recieve SMS messages. Requires handling both situations where a user is created as an un-registered user and where the user's details are stored in a progress record. See above comments for whether this is desirable or not.
- Add improved checks on entered phone fields in the users form entry. In the 1.6.5 implementation this is not very sophisticated.

Front End Download of Attachments

This was also an enhancement suggested in the Forum.

It would be easy enough to add a link to the filename, just as is currently done in the back end, but there is a need to think about any possible security concerns here as well. For example if this is a 'private' issue do we want to allow downloads. I suspect not. Do we want guest users to be able to download? Again suspect not. Issue admin and staff, probably yes. Could add an option to enable the adding of the link, but this would tend to be 'over riding' so that it would apply to all attachments. Might need to add some additional flags to the attachment to control who can download it as a better solution.

One suggestion was that it might be possible to let users choose while raising an issue? The intent being that there would an option, that the user uses to specify who can download attachments. The raiser of the issue/ticket would decide who can download their attachments - everyone, registered or staff and admin only. The only possible problem with this is whether we can really rely upon the user to specify the 'visibility'. One way would be to assume that the default is that any attachment is always 'private' unless specifically marked as 'public' by the submitter. At least this way we would have a basic level of security (of the attachment) in place without requiring the user to do anything at all.

There should also be a component parameter override option to enable or disable this front end download ability.

The criteria is such that all 'issue administrators', 'issue staff' and the issue creator should be able to download any attachments associated with the issue. If the issue is marked as 'private' then no other person should be able to download any attachments. If the issue is public and the (new) component option to enable registered users to download attachments is set (default is not to allow registered users to download attachments) then registered users are permitted to download the attachments. A link is provided around the file name to enable the download. A new controller task is invoked to permit the download to occur. If a download is not possible then the link is not available.

The download is possible from two possible places in the front end. The first is from the actual issue form. Since the form is only displayed to the issue admin, staff and issue creator, no further checks are required and the link is always displayed. The second is on the issue display screen. It is here that we have to ensure that all the checks have been performed.

Chapter 3. Database Design

In this section we specifically concentrate upon the underlying database objects.

Designing the Database Objects

Once the database requirements are defined, the next step is to turn these into a database design and an outline that describes how the user interface accepts and presents data. Given the requirements described above the three main tables can be created:

- **Projects** tracks all current projects
- **People** contains information about who can be assigned to handle issues
- **Issues** tracks all information about an issue, including the project to which it is related and the person assigned to the issue

Further detail notes upon the main tables is provided later in this document.

Subsidiary tables are created to provide reference data. Sample reference data is supplied on installation. The provide samples should be tailored by the site administrator to reflect actual ‘real’ life terms used by the organisation. Once obvious example is for foreign language sites to be able to tailor the codes for their own use.

- **Status** provides the base issue status codes. i.e. Open Closed etc.
- **Priority** provides the list of priorities an issue may take. i.e. Low, High etc.
- **Role** contains the list of business roles individuals may have. i.e. Manger, Lead etc.
- **Types** define the ‘issue type’. i.e. Whether the issue is a ‘Defect’ or perhaps and ‘Enhancement Request’, or related to product ‘Documentation’. The list is extensible and should be tailed for the specific site requirements.
- **Emails** defines the templates used for the notification messages.
- **Triggers** defines the database triggers used to populate the enhanced auditing (change history) feature.
- **Change History** holds the data populated by the database triggers that comprise the enhanced audit/change history feature.

In addition to the tables, we also need to create additional database objects, such as procedures and triggers, to support the tables. System generated primary keys will be used for all tables so that all the data can be edited without executing a cascade update.

Database Objects - Procedures, Views, Triggers

Our experience tends to indicate that very few (if any) Joomla components make use of the underlying database to its full advantage, preferring instead to implement features at the application layer. This is in our opinion missing a very important 'trick', since the database is 'closest' to the data itself. We have used our extensive database expertise to therefore make use of some of the database features with this component to provide the change data.

Important

This component makes use of database features to provide enhanced functionality specifically the use of database routines, database views and database triggers. Not all host providers allow their clients to create these types of database objects. It is advisable to check your

specific privileges before installing the component otherwise you will not be able to use the component if you cannot create database triggers.

Attempts to install the component upon a system without the permissions are handled gracefully with a message indicating that certain features will be unavailable when the component is installed.

Database Triggers

The component has a basic check upon record changes inbuilt using PHP code within the component. This is backed up with some database triggers to capture situations where changes occur 'outside' of the component. If the Joomla database connection user has not been granted the database privilege to create database triggers, then this 'back up' ability will not be available. This should not be a disadvantage to most installations.

The enhance auditing feature requires and makes use of database triggers to provide the component with 'audit/change history' data. If the Joomla database connection user has not been granted the database privilege to create database triggers, then this feature will not be available within the component.

The 'basic' Joomla installation provides some tables with a record of who made the last change to an item (i.e. an article or a web link etc.) and when they made the change. Unfortunately it doesn't inform one of what was changed. It might have been something as simple as a correction of a typological error, or it may have been some thing much more extensive. Some sites require much more information about any given change, especially sites that may house data that could be considered 'sensitive'. The enhanced auditing feature addresses this requirement and provides a more comprehensive and extensive audit over all of the changes that have occurred upon table records.

This feature makes use of the underlying features of the database and creates 'database triggers' that record the changes after they are made in the database. In this way there is no need to change any of the Joomla libraries or core code, which avoids any problems if/when the core code is changed between releases.

The following description is taken from Wikipedia:

A database trigger is procedural code that is automatically executed in response to certain events on a particular table or view in a database. The trigger is mostly used for maintaining the integrity of the information on the database. For example, when a new record (representing a new worker) is added to the employees table, new records should also be created in the tables of the taxes, vacations and salaries.

Triggers are commonly used to:

- audit changes (e.g. keep a log of the users and roles involved in changes)
- enhance changes (e.g. ensure that every change to a record is time-stamped by the server's clock)
- enforce business rules (e.g. require that every invoice have at least one line item)
- execute business rules (e.g. notify a manager every time an employee's bank account number changes)
- replicate data (e.g. store a record of every change, to be shipped to another database later)
- enhance performance (e.g. update the account balance after every detail transaction, for faster queries)

The examples above are called Data Manipulation Language (DML) triggers because the triggers are defined as part of the Data Manipulation Language and are executed at the time the data is manipulated. Some systems also support non-data triggers, which fire in response to Data Definition Language (DDL) events such as creating tables, or runtime or and events such as logon, commit and rollback. Such DDL triggers can be used for database auditing purposes.

The following are major features of database triggers and their effects:

- triggers do not accept parameters or arguments (but may store affected-data in temporary tables)
- triggers cannot perform commit or rollback operations because they are part of the triggering SQL statement (only through autonomous transactions)

MySQL

MySQL 5.0.2 introduced support for triggers. MySQL supports these trigger types:

- Insert Trigger
- Update Trigger
- Delete Trigger

Note

MySQL allows only one trigger of each type on each table (i.e. one before insert, one after insert, one before update, one after update, one before delete and one after delete).

Note

MySQL does NOT fire triggers outside of a statement (i.e. API's, foreign key cascades)

Whether a specific database user can create these triggers will depend upon the permissions granted to the database user. The database user that has to have the correct permission in the Joomla environment is the user that is used to perform all database connects as defined in the Joomla installation itself.

Note

Different databases implement database features such as database triggers in different way, and often with slightly different syntax. For that reason the current version only supports MySQL databases, the most commonly used database for Joomla systems.

Database Routines (Procedures)

A stored procedure is a subroutine available to applications that access a relational database system. A stored procedure is actually stored in the database data dictionary.

Typical use for stored procedures include data validation (integrated into the database) or access control mechanisms. Furthermore, stored procedures can consolidate and centralise logic that was originally implemented in applications. Extensive or complex processing that requires execution of several SQL statements is moved into stored procedures, and all applications call the procedures. One can use nested stored procedures by executing one stored procedure from within another. This is more efficient that handling the actions at the application layer, since the database is 'closer' to the actual data being manipulated.

Stored procedures are similar to database functions. The major difference is that functions can be used like any other expression within SQL statements, whereas stored procedures must be invoked using the CALL statement.

The component uses MySQL procedures to perform the installation (and uninstall) of the sample data used to demonstrate initial component usage. If the Joomla database connection user has not been granted the database privilege to create database routines then the 'sample data' will not be available within the component. This should not be a disadvantage to most installations since the sample database is not important to the functioning of the component.

Database Views

A view is the result set of a stored query on the data, which the database users can query just as they would in a persistent database collection object. This pre-established query command is kept in the

database dictionary. Unlike ordinary base tables in a relational database, a view does not form part of the physical schema: as a result set, it is a virtual table computed or collated from data within the database, dynamically when access to that view is requested. Changes applied to the data in a relevant underlying table are reflected in the data shown in subsequent invocations of the view.

Views can provide advantages over tables:

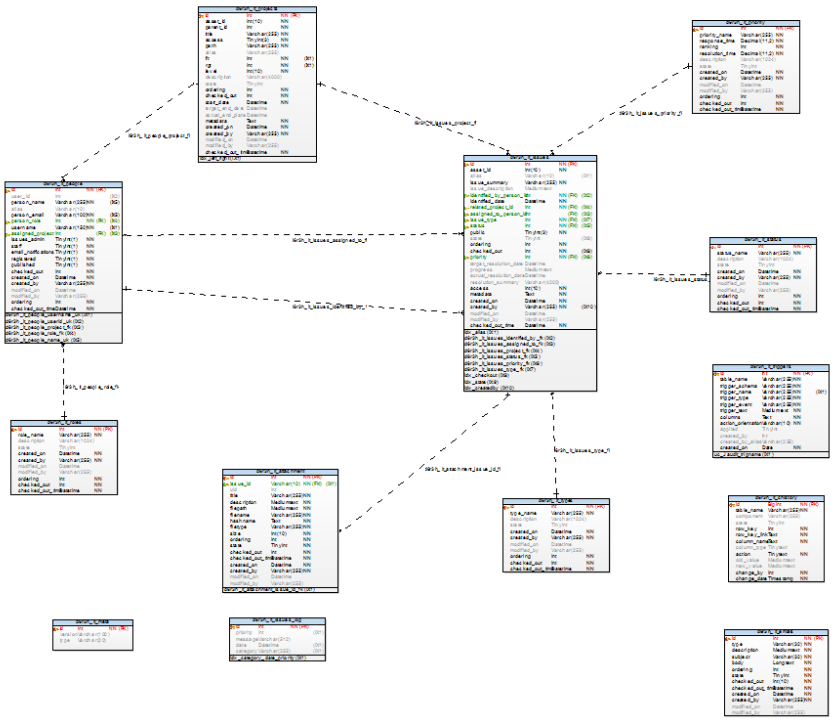
- Views can represent a subset of the data contained in a table; consequently, a view can limit the degree of exposure of the underlying tables to the outer world: a given user may have permission to query the view, while denied access to the rest of the base table.
- Views can join and simplify multiple tables into a single virtual table
- Views can act as aggregated tables, where the database engine aggregates data (sum, average etc.) and presents the calculated results as part of the data
- Views can hide the complexity of data; for example a view could appear as Sales2000 or Sales2001, transparently partitioning the actual underlying table
- Views take very little space to store; the database contains only the definition of a view, not a copy of all the data which it presents
- Depending on the SQL engine used, views can provide extra security

Issue Tracker uses views to simplify the query used by the Finder (Smart Search) plugin. If the Joomla database connection user has not been granted the database privilege to create database views, then the finder plugin uses a longer, and slightly more complex query to create 'index' entries. This though slightly slower, should not be a disadvantage to most installations.

Data Model

The data model designed is graphically shown in Figure 3.1, “Data model for Issue Tracker 1.6”.

Figure 3.1. Data model for Issue Tracker 1.6



Projects Table

Each project must include project name, project start date, target date, and actual end date columns. These date columns help determine if any outstanding issues are jeopardising the project end date.

Table 3.1. Projects table description

Column Name	Type	Size	Not Null?	Constraints	Description
id	Number	n/a	Yes	Primary key	A unique numeric identification for each project (or sub-project). Populated by an auto-sequence.
asset_id	Number	n/a	No	None	FK to the #__assets table.
parent_id	Number	n/a	No	None	Key to parent project.
title	Varchar	255	Yes	Unique key	A unique alphanumeric name for the project. Formally named project_name.
alias	Varchar	10	Yes	None	Project Alias. Used to mask primary key of issue from random selection.
description	Varchar	4000	No	None	Description of the project. Formally named project_description.
lft	Integer	11	Yes	None	Nested table left.
rgt	Integer	11	Yes	None	Nested table right.
level	Integer	10	Yes	None	Nested table level.
access	TinyInt	3	Yes	None	Required for nested table.
path	Varchar	255	Yes	None	Required for nested table.
state	Number	4	No	None	Indicates that project is visible to front end.
assignee	Integer	11	No	None	A specified default assignee for the project. Default 0.
itypes	Text	4000	No	None	A JSON encoded string containing the list of the 'types' for this project. The default is all published types. Added in 1.6.5,
customfieldsgroup	Integer	11	Yes	None	Custom field group associated with this project.
ordering	Number	n/a	No	None	Order in which projects are displayed
checked_out	Number	11	Yes	None	Joomla field record locking
checked_out_time	Date	n/a	Yes	None	Joomla field record locking
start_date	Date	n/a	Yes	None	The project start date.
target_end_date	Date	n/a	Yes	None	The targeted project end date.
actual_end_date	Date	n/a	No	None	The actual end date.
metadata	Varchar	n/a	Yes	None	Associated metadata for the record.
created_on	Date	n/a	Yes	None	Date the record was created.
created_by	Varchar	255	Yes	None	The user who created the record
modified_on	Date	n/a	Yes	None	The date the record was last modified.
modified_by	Varchar	255	Yes	None	The user who last modified the record.

Note

1) It is possible to define sub-projects as well. Each sub-project is linked through the parent_id field. In the Joomla displays the full project name is displayed for every sub-project. i.e. For a project with three levels of sub-projects the project name is displayed as:

Project Name Level 1 - Level 2 - Level 3

Where 'Level 1', 'Level 2' and 'Level 3' are the defined project_name values for the sub projects. This means that the displayed 'combined project name may become quite long, and for that reason it is suggested that the 'project name' is kept to a minimum length sufficient to uniquely identify it.

2) The ordering column permits the easy display of sub-projects below the main project in the list displays, using standard Joomla functionality.

People Table

Each person will have a defined name and role. Project leads and team members will also have an assigned project.

In order to associate the current user to a person, a username column is added to the people table. This allows flexibility when deciding on the authentication mechanism and also allows for an authorisation scheme that can determine who the person is that has logged on and if they have access to the application.

As a standard, audit columns are present on each table. They do not need to be identified during analysis because they are added consistently to each table just before implementation.

Table 3.2. People table description

Column Name	Type	Size	Not Null?	Constraints	Description
id	Number	n/a	Yes	Primary key	A numeric ID that identifies each user. Populated by an auto incremental sequence.
user_id	Number	n/a	No	Unique Key	Value obtained from the '#__users' tables.
alias	Varchar	10	Yes	None	Not currently used
person_name	Varchar	255	Yes	Unique key	A unique name that identifies each user.
person_email	Varchar	100	Yes	None	User email address.
person_role	Integer	n/a	Yes	Check constraint (MySQL)	The role assigned to each user. Foreign key to the it_roles table.
username	Varchar	150	Yes	Unique Key	The username of this person. Used to link login to person's details.
phone_number	Varchar	300	No	Unique Key	User phone number for receiving SMS messages if configured. Added in 1.6.5.
assigned_project	Number	n/a	No	None	The project this person is assigned to.
issues_admin	Number	1	No	None	Whether the person is an administrator.
staff	Number	1	No	None	Whether the person is a company staff member.
email_notifications	Number	1	No	None	Whether the person has requested email notifications.
sms_notify	Number	1	No	None	Whether the person has requested SMS notifications. Added in 1.6.5.
registered	Number	1	No	None	Whether the person is registered in Joomla.

Column Name	Type	Size	Not Null?	Constraints	Description
published	Number	1	No	None	Indicates issue is visible in front end.
ordering	Number	n/a	No	None	Specifies ordering of issues.
checked_out	Number	11	Yes	None	Joomla field record locking
checked_out_time	Date	n/a	Yes	None	Joomla field record locking
created_on	Date	n/a	Yes	None	Date the record was created.
created_by	Varchar	255	Yes	None	The user who created the record.
modified_on	Date	n/a	Yes	None	The date the record was last modified.
modified_by	Varchar	255	Yes	None	The user who last modified the record.

Note:

For the current implementation the design has been simplified. User data is usually much more elaborate and in our implementation the user information is populated automatically from the Joomla users table. Also, users typically work on more than one project at a time.

The minimum amount of information is kept about the user, sufficient only to be able to notify them if they request to be kept informed.

1. The roles that are assigned to a user are dynamic, A separate role table with a foreign key that relates to the people table.

The user information is automatically inserted into the table when a new user is registered upon the Joomla site. This is achieved via a Joomla plug-in, installed when the component is installed. The plug-in also handles the deletion of users when they are removed from the Joomla users table. Only issues assigned or raised by the deleted user are handled as specified in the ‘options: delete mode’ setting. See below for more details of options.

In release 1.1.0 and earlier no facility existed to create new users in the component. This was deliberate since the users are expected to be ‘registered Joomla users’ and there is no concept of ‘Issue Tracker users’ as a distinct entity outside of Joomla. In release 2.0 the ability to create users who are known to Issue Tracker but not registered is introduced. This means that it is possible to identify a non-registered user with an issue that they may view in the front end. The registered and user_id fields were introduced to meet this change.

The presence of the published field may seem strange since there is a security question about whether people's details should be made available to the wider public. However the web site might not be available on the web and instead be resident on an internal company intranet. In this situation the ability to control what is displayed becomes more obvious.

The staff field is used to define users who can be assigned to ‘work’ an issue.

Issues Table

It was decided to track separate issues assigned to each person. Issues will also provide a simple audit trail. The audit trail will track who created the issue, when it was created, as well as who modified the issue last and on what date that modification was made.

Table 3.3. Issues table description

Column Name	Type	Size	Not Null?	Constraints	Description
id	Number	n/a	Yes	primary key	A unique numeric ID that identifies an issue. Populated by an auto sequence.
asset_id	Int	10	No		ACL asset reference field.

Column Name	Type	Size	Not Null?	Constraints	Description
alias	Varchar	10	No	None	Issue Alias used to mask primary key of the issue from random selection.
issue_summary	Varchar	255	Yes	None	A brief summary of the issue.
issue_description	Varchar	4000	No	None	A detailed description of the issue.
identified_by_person_id	Number	11	Yes	foreign key to People ‘id’	The user who identified the issue.
identified_date	Date	n/a	Yes	None	The date the issue was identified.
related_project_id	Number	n/a	Yes	foreign key to Projects table	Project related to the issue.
assigned_to_person_id	Integer	11	No	foreign key to the People table ‘user_id’	The person who is assigned to ‘work’ the issue.
issue_type	Integer	11	Yes	Foreign key to the it_types table.	The type of issue. i.e. Defect, Enhancement etc.
status	Integer	11	Yes	Check constraint (MySQL) Foreign key to the it_status table.	The issue status. Automatically set to Open when new and set to Closed when actual resolution date entered.
public	TidyInt	3	No	None	Privacy flag to indicate the issue contains private information.
state	Number	4	No	None	Indicates issue is visible in front end.
ordering	Number	11	No	None	Specifies ordering of issues.
checked_out	Number	11	Yes	None	Joomla field record locking
checked_out_time	Date	n/a	Yes	None	Joomla field record locking
priority	Integer	n/a	to	Check constraint (MySQL) Foreign key to the it_priority table.	The priority of the issue.
target_resolution_date	Date	n/a	No	None	The target resolution date.
progress	Mediumtext	-	No	None	The progress of the issue. Varchar(4000) pre 1.3.0 release.
actual_resolution_date	Date	n/a	No	None	Actual resolution date of the issue. Will also ensure that the issue status is set to ‘Closed’.
resolution_summary	Varchar	4000	No	None	Resolution summary
access	Integer	10	Yes	None	Access group rules for the record.
custom_fields	Text	N/A	Yes	None	Custom field values.
metadata	Varchar	N/A	Yes	None	Associated metadata for the record.
created_on	Date	n/a	Yes	None	Date the record was created.
created_by	Varchar	255	Yes	None	The user who created the record.
modified_on	Date	n/a	Yes	None	The date the record was last modified.
modified_by	Varchar	255	Yes	None	The user who last modified the record.

Note

1. To be more robust there should be more comprehensive auditing carried out and recorded. This is to enable the tracking of each change to the data rather than just the last change. Tracking each change to the data would require an additional table, linked to the issues table. If the valid priorities assigned to issues need to be dynamic, we would be required to add a separate table with a foreign key that relates to the issues table.

2. The progress field is also used to store some basic information if the issue is created in the front end of the site. For unregistered users, the username, the email address and website are stored in this field. For all users the Joomla version, Database type, and Product version are stored if the option is chosen from the component options.

3. In release 1.2.0 the identified_by_person_id fields is reference to the id field of a person in the it_people table. The assigned_to_person_id is reference to the user_id field in the it_people table. This means that any person identified in the it_people table can be used as the 'identifier' of an issue, but that only registered users can have issues assigned to them. A further restriction is also used such that the assigned user must be a 'staff' member.

Progress Table

The structure of the progress table is as shown below. This table is used to store the progress records for All issues. The records each have a unique number to record the consecutive number within each issue as well as individual control over the display and access for each record.

Table 3.4. Progress table description

Column Name	Type	Size	Not Null?	Constraints	Description
id	Number	n/a	Yes	Primary key	A unique numeric identification for each progress record. Populated by an auto-sequence.
asset_id	Number	n/a	No	None	FK to the #__assets table. Note: Not currently implemented.
issue_id	Integer	11	Yes	None	The system generated unique identifier for the specific issue. FK to issue table.
alias	Varchar	10	No	None	Issue Alias. Used to mask primary key of issue from random selection. From the issue record.
public	TinyInt	3	Yes	None	Again from the issue record, whether the issue is private or public applies to the progress records.
state	Number	4	No	None	Indicates that progress record is visible to front end.
checked_out	Number	11	Yes	None	Joomla field record locking
checked_out_time	Date	n/a	Yes	None	Joomla field record locking
ordering	Number	n/a	No	None	Order in which progress records are displayed. Not currently used as the order of creation is used instead. Possibility of future usage/expansion.
progress	MediumText	n/a	Yes	None	Any progress notes on the issue resolution. The field is required to be this size since on creation any existing information in the Issues table has to be transferred over to this

Column Name	Type	Size	Not Null?	Constraints	Description
					table, and making it any smaller may truncate and hence remove important progress data for existing issues. In reality it probably does not need to be any larger than a 'Text' field.
lineno	Integer	11	Yes	None	Unique value defining the default order of the various progress updates for a specific issue.
access	TinyInt	3	Yes	None	Required for controlled group access.
created_on	Date	n/a	Yes	None	Date the record was created.
created_by	Varchar	255	Yes	None	The user who created the record
modified_on	Date	n/a	Yes	None	The date the record was last modified.
modified_by	Varchar	255	Yes	None	The user who last modified the record.

Attachments Table

The attachments table contains the details of all attachments (images or files) associated with raised issues.

Table 3.5. Attachments table description

Column Name	Type	Size	Not Null?	Constraints	Description
id	Number	n/a	Yes	primary key	A unique numeric ID that identifies a priority. Populated by an auto sequence.
issue_id	Varchar	10	Yes	Foreign key to associated issue.	Associated issue.
uid	Int	11	No	None	User id of the user attaching the file.
title	Varchar	255	Yes	None	Title for attachment.
description	Mediumtext	-	Yes	None	Description of the file attachment.
filepath	Mediumtext	1024	Yes	None	Path to the file in the system.
filename	Varchar	255	No	None	Original name of the file attachment.
hashname	text	-	No	None	Hash of file name and date string.
filetype	Varchar	255	No	None	Type of file attachment. Default `application/octet-stream`
size	Integer	10	Yes	None	Size of file attachment.
state	Number	4	No	None	Indicates issue is visible in front end.
ordering	Number	11	No	None	Specifies ordering of issues.
checked_out	Number	11	Yes	None	Joomla field record locking
checked_out_time	Date	n/a	Yes	None	Joomla field record locking
created_on	Date	n/a	Yes	None	Date the record was created.
created_by	Varchar	255	Yes	None	The user who created the record.
modified_on	Date	n/a	Yes	None	The date the record was last modified.
modified_by	Varchar	255	Yes	None	The user who last modified the record.

Priorities Table

The priorities table is a basic structure with future expansion in mind. In the release 1.1.0 only the priority_id, priority_type, state and description are used. Release 1.3.0 uses the priority ranking to implement colour coding for the associated issues. The additional fields will be implemented in a future version.

Table 3.6. Priorities table description

Column Name	Type	Size	Not Null?	Constraints	Description
id	Number	n/a	Yes	primary key	A unique numeric ID that identifies a priority. Populated by an auto sequence.
priority_name	Varchar	60	Yes	None	The name used to display on the screens.
response_time	Decimal	(11,2)	Yes	None	The target response time expressed in hours.
ranking	Int	11	Yes	None	The ranking of the priority expressed as a value between 0 and 100. Higher numbers indicate higher priority.
resolution_time	Decimal	(11,2)	Yes	None	The target resolution time expressed in hours.
description	Varchar	1024	No	None	A brief description of the issue type.
state	Number	4	No	None	Indicates issue is visible in front end.
ordering	Number	11	No	None	Specifies ordering of issues.
checked_out	Number	11	Yes	None	Joomla field record locking
checked_out_time	Date	n/a	Yes	None	Joomla field record locking
created_on	Date	n/a	Yes	None	Date the record was created.
created_by	Varchar	255	Yes	None	The user who created the record.
modified_on	Date	n/a	Yes	None	The date the record was last modified.
modified_by	Varchar	255	Yes	None	The user who last modified the record.

Note

Priority records may not be deleted whilst they are in use by any issue.

Roles Table

The roles table is used to assign different roles to the people registered upon the site. The supplied sample roles are expandable and cover the basic type of activities performed by people.

Table 3.7. Roles table description

Column Name	Type	Size	Not Null?	Constraints	Description
id	Number	n/a	Yes	Primary key	A unique numeric ID that identifies a role. Populated by an auto sequence.
role_name	Varchar	60	Yes	None	The name used to display on the screens.
description	Varchar	1024	No	None	A brief description of the role.
state	Number	4	No	None	Indicates issue is visible in front end.
ordering	Number	11	No	None	Specifies ordering of issues.
checked_out	Number	11	Yes	None	Joomla field record l1cking

Column Name	Type	Size	Not Null?	Constraints	Description
checked_out_time	Date	n/a	Yes	None	Joomla field record locking
created_on	Date	n/a	Yes	None	Date the record was created.
created_by	Varchar	255	Yes	None	The user who created the record.
modified_on	Date	n/a	Yes	None	The date the record was last modified.
modified_by	Varchar	255	Yes	None	The user who last modified the record.

Note

Issue types may not be deleted whilst they are being used by any issue.

Status Table

The status table contains a list of different statuses that an issues may be in. For example an issue is often Opened, then goes through being 'Assigned', then in being worked upon it in 'In-Progress', before being resolved, and undergoing 'QA Testing', released to the client, when it goes into a 'Waiting for Customer' before being 'Closed'. Such a mechanism is easy to create in Issue Tracker.

Table 3.8. Status table description

Column Name	Type	Size	Not Null?	Constraints	Description
id	Number	n/a	Yes	primary key	A unique numeric ID that identifies a status. Populated by an auto sequence.
status_name	Varchar	60	Yes	None	The name used to display on the screens.
description	Varchar	1024	No	None	A brief description of the status.
state	Number	4	No	None	Indicates issue is visible in front end.
ordering	Number	11	No	None	Specifies ordering of issues.
checked_out	Number	11	Yes	None	Joomla field record locking
checked_out_time	Date	n/a	Yes	None	Joomla field record locking
created_on	Date	n/a	Yes	None	Date the record was created.
created_by	Varchar	255	Yes	None	The user who created the record.
modified_on	Date	n/a	Yes	None	The date the record was last modified.
modified_by	Varchar	255	Yes	None	The user who last modified the record.

Note

Status records may not be deleted whilst they are being used by any issue.

Issue Types Table

The Issue Types relate to the different types of issues that may be raised. For example an Issue may be upon a 'Defect' of a product that is produced, or may reflect a 'Documentation' feature, or a 'Suggestion' for product improvement.

Table 3.9. Issue Types table description

Column Name	Type	Size	Not Null?	Constraints	Description
id	Number	n/a	Yes	primary key	A unique numeric ID that identifies an issue type. Populated by an auto sequence.

Column Name	Type	Size	Not Null?	Constraints	Description
type_name	Varchar	60	Yes	None	The name used to display on the screens.
description	Varchar	1024	No	None	A brief description of the issue type.
state	Number	4	No	None	Indicates issue is visible in front end.
ordering	Number	11	No	None	Specifies ordering of issues.
checked_out	Number	11	Yes	None	Joomla field record locking
checked_out_time	Date	n/a	Yes	None	Joomla field record locking
created_on	Date	n/a	Yes	None	Date the record was created.
created_by	Varchar	255	Yes	None	The user who created the record.
modified_on	Date	n/a	Yes	None	The date the record was last modified.
modified_by	Varchar	255	Yes	None	The user who last modified the record.

Email Types Table

More specifically this is really a Message Template table. This table relate to the different types of messages, emails and SMS that may be send from the Issue Tracker component. For example an email may be send when an issue is opened, updated or closed. Likewise an SMS mesage may be despatched if the AcySMS component is installed, which is used for the SMS sending interface.

Messages can be sent to issue administrators, assignees who would work the issue, or the user who raised the issue in the first place. Global options control the send of the specific message types, and a global notification override, along with optional notifications for individual users. Administrators and assignee will receive messages unless the global options are set not to do so.

The stored text format is HTML.

Table 3.10. Email types table description

Column Name	Type	Size	Not Null?	Constraints	Description
id	Number	n/a	Yes	primary key	A unique numeric ID that identifies an issue email type. Populated by an auto sequence.
type	Varchar	60	Yes	None	The messagel type name used to display to control the email sending. These are hard coded into the component and should not be changed.
description	Varchar	1024	No	None	A brief description of the issue email type.
subject	Varchar	60	Yes	None	The text to be inserted into the email header.
body	Varchar	2048	Yes	None	The text that is used for the email body. May also include tags which are substituted by the details from the issue itself with which the email is concerned.
state	Number	4	No	None	Indicates issue is visible in front end.
ordering	Number	11	No	None	Specifies ordering of issues email types.
checked_out	Number	11	Yes	None	Joomla field record locking
checked_out_time	Date	n/a	Yes	None	Joomla field record locking
created_on	Date	n/a	Yes	None	Date the record was created.

Column Name	Type	Size	Not Null?	Constraints	Description
created_by	Varchar	255	Yes	None	The user who created the record.
modified_on	Date	n/a	Yes	None	The date the record was last modified.
modified_by	Varchar	255	Yes	None	The user who last modified the record.

When installed a number of message types are automatically created in thedatabase table and these are detailed below:

Table 3.11. Message types description

Template name	Type	Description
admin_close	Email	Used when an Email message is sent to the issue administatrors for an Issue closure.
admin_new	Email	Used when an Email message is sent to the issue administartors when a new Issue is created.
admin_update	Email	Used when an Email message is sent to the issue administartors for an Issue update.
ass_close	Email	Used when an Email message is sent to the issue assignee for an Issue closure.
ass_new	Email	Used when an Email message is sent to the issue assignee when a new Issue is created.
ass_update	Email	Used when an Email message is sent to the issue assignee for an Issue update.
sms_close	SMS	Used when a SMS message is sent to a user, an assignee or the issue administatrors for an Issue closure.
sms_new	SMS	Used when a SMS message is sent to a user, an assignee or the issue administartors when a new Issue is created.
sms_update	SMS	Used when a SMS message is sent to a user, an assignee or the issue administartors for an Issue update.
user_close	Email	Used when an Email message is sent to the user (issue creator) for an Issue closure.
user_new	Email	Used when an Email message is sent to the user (issue creator) when a new Issue is created.
user_update	Email	Used when an Email message is sent to the user (issue creator) for an Issue update.

Issue Log Table

The Issue Log table is used by the logging routines to store information, of different priorities that are used to provide additional information upon the operation of the component. It is built on top of the standard Joomla JLog libraries.

Table 3.12. Log table description

Column Name	Type	Size	Not Null?	Constraints	Description
id	Number	n/a	Yes	primary key	A unique numeric ID that identifies an issue type. Populated by an auto sequence.
priority	Number	11	Yes	n/a	The priority of the message. i.e. Alert, Info, Notice, Warning etc.

Column Name	Type	Size	Not Null?	Constraints	Description
message	Varchar	512	No	n/a	The text message to store.
date	Date		No	n/a	The date the message was generated.
category	Varchar	255	No	n/a	The message category. Default 'com_issuetracker'

Table 11 - **Issues Log Table**

There is a compound index upon the category, date and priority columns in that order.

Change History Table

The Change History table contains the change records and is populated by the database triggers created and enabled in the MySQL database.

Table 3.13. Change History table description

Column Name	Type	Size	Not Null?	Constraints	Description
id	Bigint	20	Yes	Primary key	The system generated unique identifier for the record. Populated by an auto-sequence.
table_name	Varchar	256	Yes	None	Name of the table for which the change is recorded.
component	Varchar	255	Yes	None	The component to which the change record applies. The default value is 'Issue Tracker'.
state	TinyInt	4	Yes	None	State of the specific record. i.e. Published, archived, trashed etc.
row_key	Int	11	No	None	The value of the primary key for the record.
row_key_link	Text	255	No	None	The value of the alias for the record if it exists, the value of the Primary key otherwise.
column_name	Varchar	255	Yes	None	The name of the table column being recorded.
column_type	Varchar	12	Yes	Varchar	The type of column for the record. i.e. Integer, Varchar etc.
old_value	MediumText		No	None	For an DELETE or UPDATE action the former field value.
new_value	MediumText		No	None	For an UPDATE or INSERT action the new field value.
action	Varchar	12	No	None	The action of the change record. i.e.INSERT, UPDATE or DELETE.
change_date	DateTime	n/a	No	None	The date when the change was made.
change_by	Int	11	No	None	The Joomla id of the person who made the change where it can be determined otherwise the super user id.

Triggers Table

The triggers table contains details of the triggers created by the Issue Tracker component.

Table 3.14. Triggers table description

Column Name	Type	Size	Not Null?	Constraints	Description
id	Int	11	Yes	Primary key	The system generated unique identifier for the record. Populated by an auto-sequence.
table_name	Varchar	255	Yes	None	Name of the table for which the trigger applies.
trigger_schema	Varchar	255	Yes	None	The database schema to which the trigger applies. MySQL databases may have multiple schemas.
trigger_name	Varchar	255	Yes	None	The name of the trigger. This is automatically generated based upon the name of the table upon which it is apples.
trigger_event	Varchar	255	Yes	AFTER	The type of trigger. i.e. BEFORE or AFTER. Some databases may contain other trigger events.
trigger_event	Varchar	255	Yes	INSERT	The type of trigger, or more accurately when the trigger fires. i.e. INSERT, UPDATE or DELETE.
trigger_text	Medium Text		Yes	None	The actual trigger text from the specified criteria.
columns	Text	255	Yes	All	The columns to which the trigger applies.
action_orientation	Varchar	10	No	ROW	How the trigger is applied.
applied	TinyInt	4	Yes	Inactive	State of the specific trigger. i.e. Active or Inactive.
created_by	Integer	11	Yes	None	The Joomla id of the use who created the trigger text. Automatically populated.
created_by_alias	Varchar	255	No	None	The name of the person who created the trigger text.
created_on	DateTime	n/a	No	None	Date the record was changed.

Custom Field Table

This table stores the define custom fields required by a site to capture additional information that will assist in resolving raised issues.

Table 3.15. Custom Field table description

Column Name	Type	Size	Not Null?	Constraints	Description
id	Number	n/a	Yes	Primary key	A unique numeric identification for each project (or sub-project). Populated by an auto-sequence.
name	Varchar	255	Yes	None	Name for the custom field.
value	Text	n/a	Yes	None	Default value for the field.
type	Varchar	255	Yes	None	Type of the custom field.
tooltip	MediumText	n/a	No	None	Text for the tooltip of the field, if any. Probably too large a field, perhaps Varchar(255) might be more suitable.

Column Name	Type	Size	Not Null?	Constraints	Description
validation	MediumText	n/a	No	None	Validation rules for the field. Probably too large a field, perhaps Varchar(255) might be more suitable.
access	TinyInt	3	Yes	None	Required for controlled user group access.
group	Integer	11	Yes	None	Name of the group for which this field is part. Or to be precise a numeric value that relates to the entry in the Custom Field Group table. FK to the Custom Field Group.
state	Number	4	No	None	Indicates that field is visible to front end.
ordering	Number	n/a	No	None	Order in which fields are displayed
checked_out	Number	11	Yes	None	Joomla field record locking
checked_out_time	Date	n/a	Yes	None	Joomla field record locking
created_on	Date	n/a	Yes	None	Date the record was created.
created_by	Varchar	255	Yes	None	The user who created the record
modified_on	Date	n/a	Yes	None	The date the record was last modified.
modified_by	Varchar	255	Yes	None	The user who last modified the record.

Custom Field Group Table

The custom field group stores the details of all of the defined custom groups. A custom group is associated with one or more projects.

Table 3.16. Custom Field Group table description

Column Name	Type	Size	Not Null?	Constraints	Description
id	Number	n/a	Yes	Primary key	A unique numeric identification for each custom field group. Populated by an auto-sequence.
name	Varchar	255	Yes	None	Name of the custom field group.
state	Number	4	No	None	Indicates that group is visible to front end.
checked_out	Number	11	Yes	None	Joomla field record locking
checked_out_time	Date	n/a	Yes	None	Joomla field record locking
created_on	Date	n/a	Yes	None	Date the record was created.
created_by	Varchar	255	Yes	None	The user who created the record
modified_on	Date	n/a	Yes	None	The date the record was last modified.
modified_by	Varchar	255	Yes	None	The user who last modified the record.

Meta Data Table

The Meta table is used by the component itself to record version numbers of its component parts. i.e. Modules, plugins etc.

Table 3.17. Meta data component table description

Column Name	Type	Size	Not Null?	Constraints	Description
id	Int	11	Yes	Primary key	The system generated unique identifier for the record. Populated by an auto-sequence.
version	Varchar	100	Yes	None	Version number of the installed component.
type	Varchar	20	Yes	None	Type of extension.

Implementing Database Objects

This first step in building an application was to create the database objects. The tables have been defined above but there are a few other database objects we require.

Additional Database Objects Needed

To populate the primary key values of the tables needed for this application, an auto incrementing sequence is used in the MySQL implementation.

Triggers are present upon the tables for insert and update. These triggers are used to set the values of the audit fields (if they are not set by the application) and also to set curtain defined business criteria. These database triggers act as a secondary mechanism in case data is entered into the database outside of the application. (The addition of the sample data is performed in this way.)

Certain database procedures are used to encapsulate mechanisms into the database. i.e. The sample data is provided in the form of a database procedure, and the application provides a call to the procedures.

About Building Database Objects

There are several tools for creating objects in MySQL. You can:

- Use phpAdmin to create tables, views, indexes, procedures, functions, triggers etc.
- Use a product such as Toad, which provides the same abilities as phpAdmin with a cleaner interface. This is downloadable from the Quest website.
- Create a script in a standard text editor and then upload the script and use the MySQL command interface, Toad, or phpAdmin to execute it.

Default Base Data

There are a few provided records inserted into the tables upon installation.

The projects table has a Root entry created with an id value of 1, which acts as the base project from which all other projects derive. This particular 'Root' project is 'hidden' from viewing or changing within the component.

There is also a default 'Unspecified Project' created with an id value of 10. This projects acts as the project to which all issues are assigned if another project is not specified. There is a component option to change this if desired.

The people table is populated with a default 'Anonymous' user, usually with an id value of 1. In rare instances, such as where the id value is already in use, as would be the situation were the root id have such a value, the 'Anonymous' id value would then become 2. This entry is used to populate any new issues with the identifying user, unless another is chosen upon issue creation.

The default project and default person may be edited and altered in any way, and once the component options are set, can if not used as ‘default settings’ be safely removed, although this is not necessarily recommended.

There are also sample reasons, priorities, issue types, statuses, email templates etc provided that the site administrator can edit and modify to reflect site design. These may be changed and will not be generally be updated as a result of upgrading the component. See the release notes for the particular upgrade where there would be a specific mention were this to change.

Chapter 4. CSS Styling

This section describes some of the possible configuration that can be performed to change the 'look and feel' of the front end of the site. It is not the purpose of this document to describe all of the possible CSS configuration options that can be used, but instead to describe a few of the more commonly used options.

Note

It is beyond the scope of this document to go into the details of CSS styling as there are a number of resources available on the web providing details.

It is possible to change the CSS styling used on your site by using a media override. When installed the CSS stylesheets for the Issue Tracker Component are installed in the 'media/com_issuetracker' directory. Release 1.3.0 allows the site administrator to create a site override by creating a directory named media/com_issuetracker under the site template directory. One then copies the folders and files that it is desired to override in this location. Then almost like magic, Issue Tracker will now load which ever files are under the template instead of that contained in the media directory under the site's root directory.

The following specific class names are supplied in the Issue Tracker list displays (Issues, People and Projects):

- itstyle : A copy of the standard adminlist table styles. This replicates very closely the default theme used in versions 1.2.2 and earlier of Issue Tracker.
- itbluestyle : A blue themed table style.
- italtstyle : A greenish table theme.

There are also other class names available on the row and data definitions - fieldDiv fieldLabel, and fieldDiv fieldValue which are also present upon the list views and are present to permit additional styling if it should be desired.

An example of using these fields is given below. This simple example changes the 'legend' and the colour and weight of the displayed text. The classes are provided to enable a designer to modify the form to match site styling requirements.

```
/*************************************/
/* Sample style for front end FieldDev */
/*************************************/
.fieldDiv body {
        font-family:arial, verdana, sans-serif;
        font-size:0.8em;
        background:#8be;
        color:#fff;
}
.fieldDiv legend  {
        font-size:14px;
        background:#FAFAFA;
        border:1px solid #C3C3C3;
        border-radius:0 0 3px 3px;
        position:relative;
        width:auto;
        padding:2px 10px;
}
.fieldDiv dl {
        font-weight: bold;
```

```
}
.fieldDiv dt {
        font-weight: bold;
        letter-spacing: 2px;
        color: #bf2c22;
}
.fieldDiv dd {
        color: #5454d4;
}
```

The figures below shows the results of the CSS fieldDiv styling on a simple item display with and without styling.

Figure 4.1. Item without fieldDiv styling.

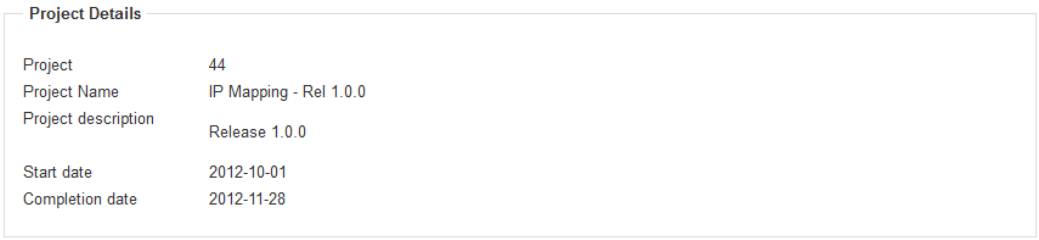
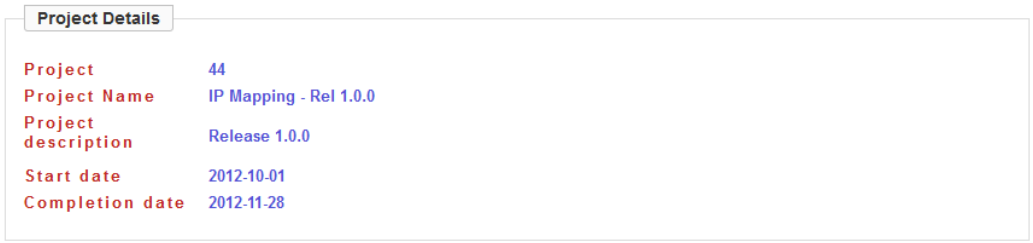


Figure 4.2. Item with fieldDiv styling.



Change the fieldDiv styling in the template override file to suit the site standards. This ensures that your changes are not lost when the component is updated.

The name of the CSS style sheet file used in the front end of your site is named 'issuetracker.css'. The supplied file contents are shown below and presents a blue theme to the listing display.

To retain the style of existing sites the default 'adminlist' remains as it was in release 1.2.2 and earlier. This style will have been defined in the site default system template.

The three themes are provided as examples of what can be achieved and all that is necessary is to specify the desired theme in the menu configuration option from the 'Advanced' tab.

If you are creating your own theme then one can either edit one of the existing themes and introduce your own settings, or create a completely new theme with its own unique name, and specify that in the menu option Advanced settings.

Caution

The supplied CSS styling examples below may change between releases and is provided as a guide only. For the version installed with your version of the component please inspect the CSS files supplied within the installation zip file.

Blue theme CSS styling

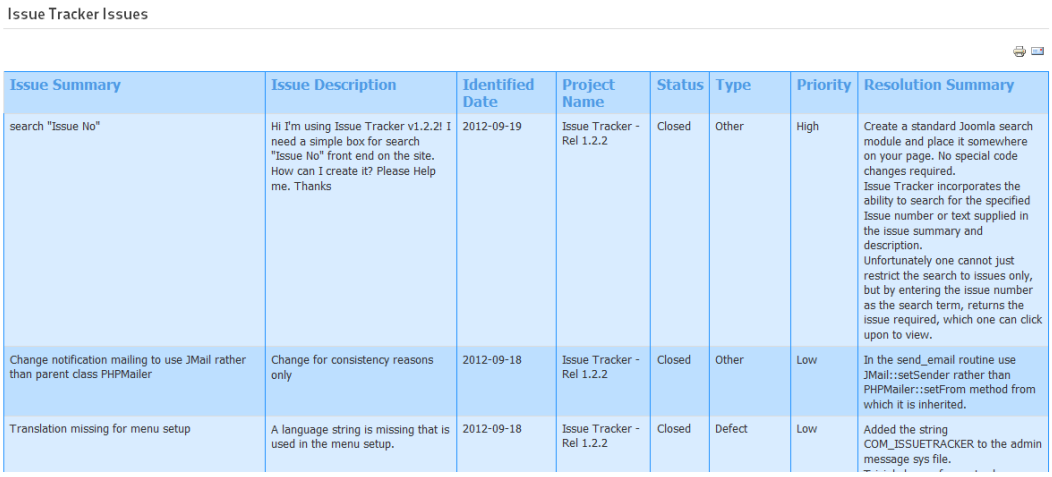
The details settings employed in the 'blue theme' are displayed below. Recent changes may not be reflected in the documentation and you are advised to view the settings directly in the CSS file for the latest version.

```
/*************************/
/* Table look and feel   */
/* Blue theme            */
/*************************/
/*
table.itbluestyle table,
table.itbluestyle td,
table.itbluestyle th {
   border-collapse:collapse;
   margin:0;
   padding:0;
}

table.itbluestyle {
   border: none;
   font-family: Garamond, Georgia, "Times New Roman", Times, serif;
}

table.itbluestyle table caption {
   background-color: transparent;
   background-repeat: repeat-x;
   background-position: bottom left;
   text-align:left;
   font-size:150%;
   text-transform:uppercase;
   line-height: 30px;
   letter-spacing:0px;
}

table.itbluestyle td,
table.itbluestyle th{
   vertical-align:top;
   font-weight:normal;
}

table.itbluestyle thead {
   border-left:1px solid #2293FF;
}

table.itbluestyle thead th {
   background: #bddfff;
   color: #FFFFFF;
   font-size: 125%;
   font-weight: bold;
   border-top:1px solid #2293FF;
   border-right:1px solid #2293FF;
   padding: 4px 0.4em 4px 0.4em;
}

table.itbluestyle tfoot td,
table.itbluestyle tfoot th {
```

```
   border-top:1px solid #2293FF;
   border-right: none;
   font-weight: bolder;
   font-size: 110%;
   padding: 0.4em 0.5em 0.4em 0.5em;
}

table.itbluestyle tbody tr.row0 td,
table.itbluestyle tbody th {
   background-color: #D9ECFF;
   border-right:1px solid #2293FF;
   border-left:1px solid #2293FF;
   padding: 0.4em 0.5em 0.4em 0.5em;
        word-wrap: break-word;
}

table.itbluestyle tbody tr th,
table.itbluestyle tbody tr.row1 td{
   background-color: #BDDFFF;
   border-right:1px solid #2293FF;
   border-left:1px solid #2293FF;
   padding: 0.4em 0.5em 0.4em 0.5em;
        word-wrap: break-word;
}

table.itbluestyle tbody th{
   font-weight: bold;
   border-left:1px solid #2293FF
}

table.itbluestyle a:link,
table.itbluestyle a:visited,
table.itbluestyle a:active {
   color:#444F66;
   background-color: transparent;
   text-decoration: underline;
}

table.itbluestyle a:hover {
   text-decoration:none;
   background-color: #1E90FF;
   color: #FFFFFF;
}
```

The figure below shows the results of the blue theme styling.

Figure 4.3. Blue theme css styling


Issue Tracker Issues

Issue Summary	Issue Description	Identified Date	Project Name	Status	Type	Priority	Resolution Summary
search "Issue No"	Hi I'm using Issue Tracker v1.2.2! I need a simple box for search "Issue No" front end on the site. How can I create it? Please Help me. Thanks	2012-09-19	Issue Tracker - Rel 1.2.2	Closed	Other	High	Create a standard Joomla search module and place it somewhere on your page. No special code changes required. Issue Tracker incorporates the ability to search for the specified Issue number or text supplied in the issue summary and description. Unfortunately one cannot just restrict the search to issues only, but by entering the issue number as the search term, returns the issue required, which one can click upon to view.
Change notification mailing to use JMail rather than parent class PHPMailer	Change for consistency reasons only	2012-09-18	Issue Tracker - Rel 1.2.2	Closed	Other	Low	In the send_email routine use JMail::setSender rather than PHPMailer::setFrom method from which it is inherited.
Translation missing for menu setup	A language string is missing that is used in the menu setup.	2012-09-18	Issue Tracker - Rel 1.2.2	Closed	Defect	Low	Added the string COM_ISSUETRACKER to the admin message sys file.

Alternative 'greenish' theme CSS styling

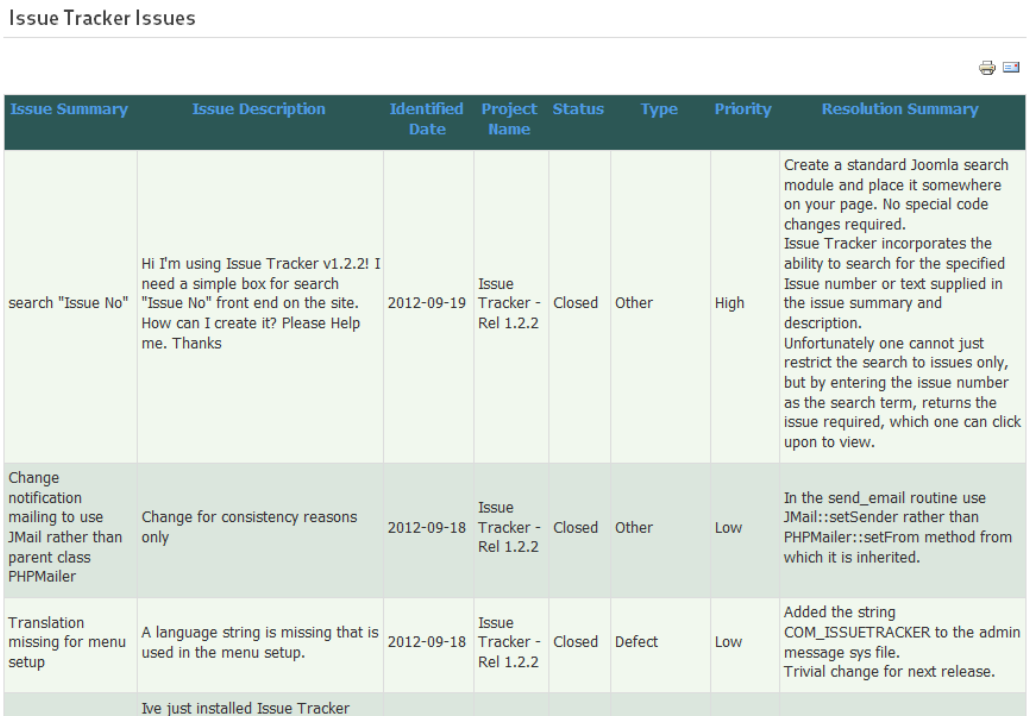
We realise that a blue theme is not to everyone's taste so we have also supplied an alternative which is commented out in the css file. The details settings employed in the 'greenish theme' are displayed below. Recent changes may not be reflected in the documentation and you are advised to view the settings directly in the CSS file for the latest version.

```
/************************/
/* Table look and feel   */
/* Alternative  theme    */
/************************/
table.italtstyle {
   border-collapse: collapse;
   border: 1px solid #839E99;
   background: #f1f8ee;
   font: .9em/1.2em Georgia, "Times New Roman", Times, serif;
   color: #033;
}

table.italtstyle caption {
   font-size: 1.3em;
   font-weight: bold;
   text-align: left;
   padding: 1em 4px;
}

table.italtstyle td,
table.italtstyle th {
   padding: 3px 3px .75em 3px;
   line-height: 1.3em;
}

table.italtstyle th {
   background: #839E99;
   color: #fff;
   font-weight: bold;
   text-align: left;
   padding-right: .5em;
   vertical-align: top;
```

```
}

table.italtstyle thead th {
   background: #2C5755;
   text-align: center;
}

table.italtstyle tr.row0 td {
   background: #f1f8ee;
   word-wrap: break-word;
}

table.italtstyle tr.row0 th {
   background: #2C5755;
}

table.italtstyle tr.row1 td {
   background: #DBE6DD;
   word-wrap: break-word;
}

table.italtstyle tr.row1 th {
   background: #6E8D88;
}

table.italtstyle td a,
table.italtstyle td a:link {
   color: #325C91;
}

table.italtstyle td a:visited {
   color: #466C8E;
}

table.italtstyle td a:hover,
table.italtstyle td a:focus {
   color: #1E4C94;
}

table.italtstyle th a,
table.italtstyle td a:active {
   color: #fff;
}

table.italtstyle tfoot th,
table.italtstyle tfoot td {
   background: #2C5755;
   color: #fff;
}

table.italtstyle th + td {
   padding-left: .5em;
}
```

The results of the styling is shown in the figure below.

Figure 4.4. Alternative theme styling



Issue Tracker Issues

Issue Summary	Issue Description	Identified Date	Project Name	Status	Type	Priority	Resolution Summary
search "Issue No"	Hi I'm using Issue Tracker v1.2.2! I need a simple box for search "Issue No" front end on the site. How can I create it? Please Help me. Thanks	2012-09-19	Issue Tracker - Rel 1.2.2	Closed	Other	High	Create a standard Joomla search module and place it somewhere on your page. No special code changes required. Issue Tracker incorporates the ability to search for the specified Issue number or text supplied in the issue summary and description. Unfortunately one cannot just restrict the search to issues only, but by entering the issue number as the search term, returns the issue required, which one can click upon to view.
Change notification mailing to use JMail rather than parent class PHPMailer	Change for consistency reasons only	2012-09-18	Issue Tracker - Rel 1.2.2	Closed	Other	Low	In the send_email routine use JMail::setSender rather than PHPMailer::setFrom method from which it is inherited.
Translation missing for menu setup	A language string is missing that is used in the menu setup.	2012-09-18	Issue Tracker - Rel 1.2.2	Closed	Defect	Low	Added the string COM_ISSUETRACKER to the admin message sys file. Trivial change for next release.
	Ive just installed Issue Tracker						

The styles are applicable to the list displays available on the front end of the site. i.e. Issues List, Projects List and People List. The menu options available under the 'Advanced' tab permits the specification of the required table style to be used. All that is necessary is that the named table class is present in the 'site template' over ride file.

Spearmint flavoured style.

After Issue Tracker release 1.3.0 was released we resolved a small problem with a 'spearmint' coloured CSS style. Whilst working on presenting a sample table style for the Issue Tracker component one particular colour combination was creating a few problems. The main problem was the colour of the links in the table header. These would be used to sort the list and due to the choice of colour scheme the link colours were taking the settings from an earlier (in the tree) DIV style.

This resulted in us temporarily shelving the table style until such time as we had to investigate further. Since the release of 1.3.0 we have revisited the style sheet. The investigation was interesting and in the interests of sharing we made it a Blog post.

The resolution was to make use of the '!important' declaration which been around since CSS1 but should still be used with caution.

Declaration

The !important declaration can be added at the end of any CSS property/value.

```
table.spearmint thead a:link { color: white !important; }
```

Application order

CSS assigns a weight to each rule which is based upon the specificity of its selector and the position in the source. The result determines which style is applied to an HTML element.

If 2 rules conflict on a single element then the following principles will be applied:

Origin of rules - If a rule between an author and a user style sheet conflicts, the user's rules will win over the author's rules.

Specificity - When 2 or more declarations that apply to the same element set the same property and have the same importance and origin, the declaration with the most specific selector will be applied.

Source order - When 2 rules have the same weight, the last rule declared in the style sheet will be applied.

There might be times when it would be useful to change the order of sequence so it is possible to break the cascading chain by using the !important CSS declaration. When the !important declaration is used on a property/value, that value becomes the most important for that property and overrides any others.

The final style result is presented below:

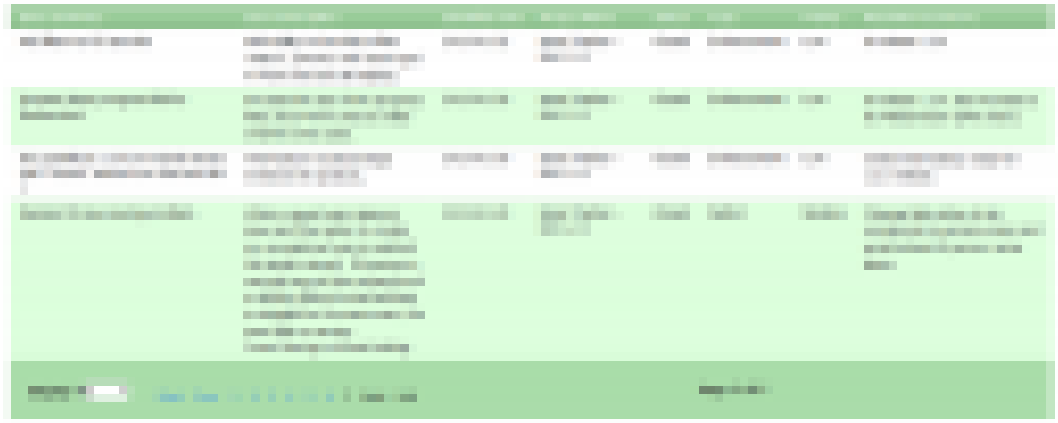
```
/* Spearmint tints */
table.spearmint {
   margin: 0 0 1em;
   background: #FFF;
   border-collapse: collapse;
   border-top: 1px solid #363;
   border-bottom: 2px solid #363;
}

/* caption = table title/heading */
table.spearmint caption {
   text-align: left;
   font: bold small-caps 120%/1.3 "trebuchet
 ms",Helvetica,Arial,Sans-Serif;
   color: #363;
   margin: .3em 0; }

table.spearmint thead a:link {
   color: white !important; }

/* reduced font size to save space */
table.spearmint tr { font-size: 90%; }
/* prevent nested tables reducing font size further */
table.spearmint tr tr { font-size: 100%; }

/* tinted rows */
/* in CSS3 selectors: tbody tr:even or tbody tr:nth-child(2n) */
table.spearmint tr.row1 { background: #DFD; }

/* table cells */
table.spearmint th, td {
   font-weight: normal;
   padding: .3em .7em;
   text-align: left;
   vertical-align: top; }

/* borders to separate body sections */
table.spearmint tbody tr:first-child th,
table.spearmint tbody tr:first-child td,
table.spearmint tfoot tr:first-child th,
table.spearmint tfoot tr:first-child td { border-top: 1px solid
 #363; }

/* tints for column headings */
```

```
table.spearmint thead { background: #9C9; white-space: nowrap; }

/* tints for totals */
table.spearmint tfoot { background: #ADA; }

/* bold text for totals */
table.spearmint tfoot th, table.spearmint tfoot td {
   font-weight: bold; }
```

Usage in Issue Tracker

To use with the Issue Tracker component add the above style to the file templates/your_template/media/com_issuetracker/css/issuetracker.css. Then on the menu item for the list display use the 'Advanced' tab and enter the name 'spearmint' to the table css style parameter. The end result will/should look similar to the following:

Figure 4.5. Spearmint theme styling



List Row colouring

This is supplied as it was a question raised in the forum and others may wish to produce a similar effect on their sites.

The request was to be able to provide a row colour style in 1.2.2 to the Issue List display based upon the issue status code. The easiest way to achieve this is to edit the site/components/com_issuetracker/views/itissueslist/tmpl/default.php file.

We illustrate the providing colours to the open and the closed issue status. All other statuses can be similarly changed and it is left to the site to provide these since the status codes is use are site dependant.

In the file add the following lines in a suitable position. (i.e. In release 1.2.2 after line 31)

```
<style>
table.adminlist tr.status-4 td { background: red; color: white; }
table.adminlist tr.status-1 td { background: lime; color: black; }
</style>
```

It is now necessary to edit the line containing the 'tr' code within the 'tbody' section of the file. (i.e. In release 1.2.2 the new line 193)

Change

```
        <tr>
```

To

```
<tr class="<?php echo 'status-'.$dataItem->status;?>">
```

Edit the styles as you require. The number in the style relates to the specific id of the status code itself. i.e. 1 = Closed, 4 = Open. The colours may be a bit bright but you may specify them as an RGB code such as #20B2AA if you want something intermediate.

Note that this is possibly not the most efficient way to add a style, which should be in a stylesheet, but it will be the easiest if you are not all that familiar with CSS etc.

Front End form

The front end 'raise issue' form makes use of some standard classes:

- formelm
- formelm-area
- formelm-buttons

These classes are probably defined in your site template. The decision to use these styles was deliberate so that the form matches the style of any other forms that may be used upon your website such as those used by the com_content forms. Issue Tracker does not itself use any special CSS of its own definition for the form.

Back End styling

The styling sheet for the back end is named 'administrator.css'. It contains all of the colour coding, and table style specific to the administrator forms. Where a class is not specified within this file, the details of the style are obtained from the site template.

Bootstrap CSS

With the implementation of Bootstrap into Joomla with Joomla 3.0 we implemented changes in the back end of the site to handle the Isis template. At the same time retaining the former CSS for the alternative Hathor template. There was however only rudimentary changes made to the front end which mainly revolved around the button displays which used the Bootstrap styling.

The rationale was that it was virtually impossible to know what template any given site was using, there being a number of different template frameworks available, any (and probably most) of which were being used somewhere. For that reason it was often necessary for a site to create a template override and for them to modify the overridden files as required to create their required display.

We ourselves have looked at Bootstrap and decided that it is the direction to move and for that reason we are developing template overrides suitable for a Bootstrap front end template. The current release of Bootstrap as of 26th April 2014 is release 3.1.1 and it seemed sensible to therefore use this version for the development of our overrides. The main focus of Bootstrap v3 was 'mobile' focused and mainly involved changes to the CSS classes,so it would be possible to back port versions for Bootstrap v2 if required.

Chapter 5. Helper Classes

General Helper

The general issue tracker helper is used to present display list options, and also is called for the generation of all of the notification messages.

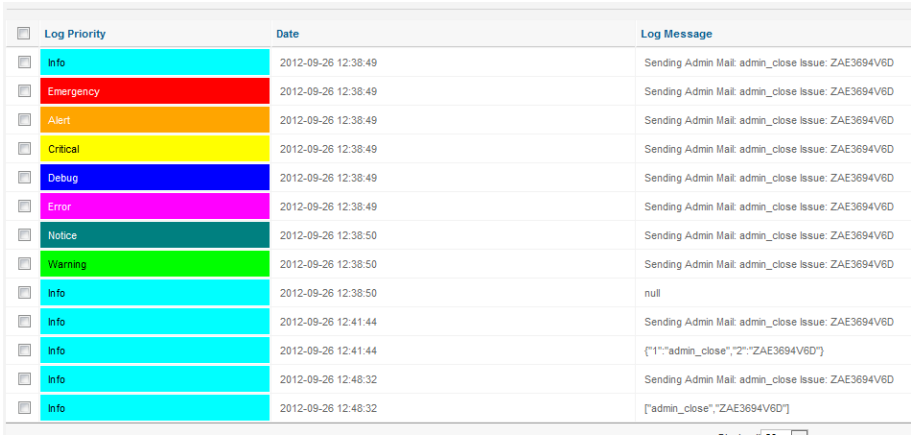
Logging Helper

Release 1.3 introduced a logging mechanism so that messages may be logged to a database table. A database table was chosen as the location since it is easier to monitor from within the application.

The logging mechanism is built upon the Joomla Jlog libraries and implements all the current logging priorities.

The implementation implements colour coding of the messages depending upon their priority as shown in the figure below:

Figure 5.1. Log Entry Display



Log Priority	Date	Log Message
Info	2012-09-26 12:38:49	Sending Admin Mail: admin_close Issue: ZAE3694V6D
Emergency	2012-09-26 12:38:49	Sending Admin Mail: admin_close Issue: ZAE3694V6D
Alert	2012-09-26 12:38:49	Sending Admin Mail: admin_close Issue: ZAE3694V6D
Critical	2012-09-26 12:38:49	Sending Admin Mail: admin_close Issue: ZAE3694V6D
Debug	2012-09-26 12:38:49	Sending Admin Mail: admin_close Issue: ZAE3694V6D
Error	2012-09-26 12:38:49	Sending Admin Mail: admin_close Issue: ZAE3694V6D
Notice	2012-09-26 12:38:50	Sending Admin Mail: admin_close Issue: ZAE3694V6D
Warning	2012-09-26 12:38:50	Sending Admin Mail: admin_close Issue: ZAE3694V6D
Info	2012-09-26 12:38:50	null
Info	2012-09-26 12:41:44	Sending Admin Mail: admin_close Issue: ZAE3694V6D
Info	2012-09-26 12:41:44	{"1":"admin_close","2":"ZAE3694V6D"}
Info	2012-09-26 12:48:32	Sending Admin Mail: admin_close Issue: ZAE3694V6D
Info	2012-09-26 12:48:32	["admin_close","ZAE3694V6D"]

The logging is controlled by a component option. Entries logged will be dependant upon the version of the code installed.

In addition to being able to store simple text strings, there is also a routine to store an array using JSON encoding.

Cron Helper

Release 1.5 introduced a new Helper class for 'cron' tasks generally, but most specifically for the functionality around using email for raising and updating issues. It makes extensive use of the Logging helper to record activities associated with the fetching and updating of issue records.

Date Helper

Release 1.6.2 introduced a new Helper class for the handling of date variables. The helper was written to make it easier to handle the various dates that are handled by the component. Suppose we have a situation where a User A fills out a form, to raise an issue. The issue stores a date variable that represents the date that the issue was opened. We wish to save this date for display when ever the issue is displayed, either back to the user or to any other authorised viewer. The best thing to do when the date is stored in the database is to convert it to UTC. You might ask why store it as UTC? The reason

is that we are then providing ourselves with a starting point and when we present the output to the user, we can add different timezones depending on the users place or origin/display.

The technique of converting the date to and from UTC means that a helper class is an ideal place to locate the code,which will be used over and over again. For example an issue has a number of dates when an issue is stored. For example we have the identified date, an estimated completion date, a closure (resolution)date and also dates used in the auditing, for when the issue was first saved, and when it was last modified.

The helper routine makes use of the Joomla JDate function. Inspection of the JDate class in libraries/joomla/utilities/date.php shows that the constructor actually expects 2 parameters -> the date and the timezone. Thus when one saves a date one would generally want to do something like this:

```
$date = new JDate($myDate, $myTimezone);
```

Quite often however the timezone would not be specified and this would result in the date variable be treated as being in UTC. The question then becomes how to properly calculate a timezone? Fortunately Joomla provides a means to do that as well. Within out helper class we have a small routine as follows:

```
   /**
    * Return the date with the correct specified timezone offset.
    * If a user timezone is specified use that otherwsie use the
 server timezone.
    *
    * param : raw date string (date with no offset yet)
    * return : JDate object
    */
   public static function dateWithOffSet($str='')
   {
      $userTZ  = self::getOffSet();
      $date    = new JDate( $str );

      $user    = JFactory::getUser();
      if($user->id != 0) {
            $userTZ  = $user->getParam('timezone');
      }

      if (empty($userTZ)) {
         $jversion = new JVersion();
         if ( version_compare( $jversion->getShortVersion(), '3.2',
 'ge' ) ) {
            $userTZ = JFactory::getApplication()->get('offset');
         } else {
            $config  = JFactory::getConfig();
            $userTZ  = $config->get('offset');
         }
      }

      $tmp = new DateTimeZone( $userTZ );
      $date->setTimeZone( $tmp );

      return $date;
   }
```

As can be seen in the code above we first try to obtain the user's timezone, and then we get the global config timezone. If the user has set a timezone in their configuration, then the value of it is passed to the DateTimeZone object. If the user on the other hand has not set a timezone, or the guest user has not set a timezone, then we use the global one. Now that we have the correct time zone we can format the date to the MySQL format and store it in the database.

To save the date in the database we now want to convert it to the correctly formatted string. Depending upon the version of PHP in use will influence how we do this. In PHP 5.4 and above method chaining is possible so we could use a function as follows:

```
$tz = 'EUROPE/LONDON';
$d11 = (new JDate($odate), $tz)->format('Y-m-d H:i:s', false,
 false);
```

In PHP 5.3 and below we have to call the method upon the variable instead. Since we do not know what versions of PHP is in use we assume that the PHP 5.3. syntax is the most appropriate as follows:

```
$tz = 'EUROPE/LONDON';
$d11 = new JDate($odate, $tz);
$d12 = $d11->format('Y-m-d H:i:s', false, false);
```

The first parameter to the format function is 'Y-m-d H:i:s' - this is the format we want our date to be saved in the db. The second parameter tells the function that we want to have the GMT/UTC time and the third parameter tells the format function that there is no need to translate the date. We can then save the date/time variable ($d12) in the database in the standard way.

Note

There are also a few other things worthy of note:

- JHtml::_('date', $myDate) will output an UTC date in the user's timezone automatically -> so there is no need to calculate the timezone oneself.
- JHtml::_('calender', myDate ...) won't convert the date to the user's timezone so one has to make sure that you provide the date with the correct timezone
- If one uses JForm calender time one can provide a filter: SERVER_UTC or USER_UTC that will handle the timezone calculations. (See note earlier in this document).

Database Triggers time

The Issue Tracker component has a fallback auditing feature such that if the 'creation date' is null when a record is saved, or if the 'modified date' is null when a record is updated, then a database trigger will automatically populate the fields when the record is saved. Prior to release 1.6.2 the triggers made use of the database function SYSDATE() which returns the current date and time in YYYY-MM-DD HH:MM:SS or YYYYMMDDHHMMSS.uuuuuu format depending on the context of the function. The SYSDATE function date/time value is the time when the actual statement executed. This is slightly different from the NOW function (described below) which is the time when the statement started executing. The distinction may seem imprecise and for most purposes the difference is likely to be slight if any.

The database function NOW() which returns the value of current date and time in ‘YYYY-MM-DD HH:MM:SS’ format or YYYYMMDDHHMMSS.uuuuuu format depending on the context (numeric or string) of the function. The database functions CURRENT_TIMESTAMP, CURRENT_TIMESTAMP(), LOCALTIME, LOCALTIME(), LOCALTIMESTAMP, LOCALTIMESTAMP() are all synonyms of NOW(). The NOW() returns the constant time when the statement began to work.

There are a few specific UTC date time functions available in MySQL. These are UTC_DATE, UTC_TIME and UTC_TIMESTAMP. The most appropriate of these is the UTC_TIMESTAMP function since we wish to have the date and the time available. As of release 1.6.2 the triggers now use the database function UTC_TIMESTAMP which returns the current UTC (Coordinated Universal Time) date as a value in 'YYYY-MM-DD HH:MM:SS' or 'YYYYMMDDHHMMSS.uuuuuu' format depending on the context of the function i.e. in a string or numeric context.

Part II. Appendices

Table of Contents

Appendix A. GNU General Public License version 3

Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program - to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright"; also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program"; refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a. The work must carry prominent notices stating that you modified it, and giving a relevant date.

b. The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c. You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d. If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a. Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b. Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c. Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d. Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e. Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a. Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b. Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c. Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d. Limiting the use for publicity purposes of names of licensors or authors of the material; or

e. Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f. Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the program's name and a brief idea of what it
 does.
Copyright (C) year name of author

This program is free software: you can redistribute it and/or
 modify
it under the terms of the GNU General Public License as published
 by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
```

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

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

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
program Copyright (C) year name of author
This program comes with ABSOLUTELY NO WARRANTY; for details type
 'show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type 'show c' for details.
```

The hypothetical commands '`show w`' and '`show c`' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see http://www.gnu.org/licenses/ [http://www.gnu.org/licenses/].

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read http://www.gnu.org/philosophy/why-not-lgpl.html [http://www.gnu.org/philosophy/why-not-lgpl.html].

Appendix B. GNU Free Documentation License

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/ [http://www.gnu.org/copyleft/] .

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (C) YEAR YOUR NAME.
>
> Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.