

EXT: Developer Log

Extension Key: devlog

Language: en

Keywords: forAdmins, forDevelopers, forIntermediates

Copyright 2000-2008, François Suter, <typo3@cobweb.ch>

This document is published under the Open Content License
available from <http://www.opencontent.org/opl.shtml>

The content of this document is related to TYPO3
- a GNU/GPL CMS/Framework available from www.typo3.org

Table of Contents

EXT: Developer Log.....	1	Developer manual.....	6
Introduction.....	3	Using the logging method.....	6
Credits.....	3	Extra information.....	7
Screenshot.....	3	User manual.....	8
Questions and support.....	3	Viewing a single entry.....	8
Keeping the developer happy.....	3	The BE module.....	9
Participating.....	4	Viewing the log entries.....	10
Installation.....	5	Cleaning up.....	12
Switching from cc_devlog.....	5	To-Do list.....	13
Configuration.....	5	Changelog.....	14

Introduction

“Developer log; This should be implemented around the source code, both frontend and backend, logging everything from the flow through an application, messages, results from comparisons to fatal errors. The result is meant to make sense to developers during development or debugging of a site.” (quoted by René Fritz, source unknown)

TYPO3 provides a functionality for logging information for the purposes of development and debugging via the static method `t3lib_div::devLog()`. However this method does not do anything by itself. It simply provides a hook for registering whatever logging method you wish. The Developer Log provides such a method.

Information is stored in a database table and can be browsed using a backend module.

This extension provides the following features:

- Log entries are logged in log sessions – one log session per script run. That makes it easy to separate log runs.
- Frontend logging includes the page id
- Backend module can be opened in a separate window and set to refresh automatically
- Show log runs, latest entries or all entries with filtering, searching and paginating options
- Clear up logs selectively

Credits

Obviously credits are due to René Fritz who developed the original `cc_devlog` extension. I dearly wanted to just contribute or take over the development but given the apparent impossibility of getting ownership of the extension key I finally decided to fork it.

Many people contributed some code and ideas to this new version (see Changelog).

Some of the icons used in this extension come from the Silk Icons library (<http://www.famfamfam.com/lab/icon/silk/>), a beautiful set of icons developed under the Creative Commons license by Mark James, of Birmingham, UK.

Screenshot

Here is a view of what you get in the backend module.

Developer Log

Select Log: ☐ Auto refresh ☐ Expand all extra data

LOG ENTRIES:

Log period: 13-02-09 10:25 (16 days) - 22-02-09 22:26 (7 days)

The following keys are currently not logged: t3lib_userAuth,realuri

Search log data:

Entries: 0-30 30-60 60-90 90-120 120-150 150-180 180-210 210-240 240-270 270-300 300-330 330-360 360-390 390-420 420-450 450-480 480-510 510-540 540-570 570-600 600-630 630-660 660-690 690-720 720-750 750-780 780-810 810-840 840-870 870-900 900-930

Date ?	Severity ?	Extension ?	Message ?	Called from ?	Page ?	User ?	Extra data ?
22-02-09 22:26:45	✓	sv	User setup	class.tx_sv_authbase.php, line 161			
22-02-09 22:20:35	✓	sv	User setup	class.tx_sv_authbase.php, line 161			
22-02-09 22:20:25	✓	sv	User setup	class.tx_sv_authbase.php, line 161			
22-02-09 22:19:49	✓	sv	User setup	class.tx_sv_authbase.php, line 161			
22-02-09 22:18:08	✓	sv	User setup	class.tx_sv_authbase.php, line 161			
22-02-09 22:18:08	✗	fimldap	(init) LDAP servers no responded	class.tx_fimldap_sv1.php, line 96			
22-02-09 22:18:08	✗	fimldap	(LDAPconnectionOpen) LDAP socket connection failed.	class.tx_fimldap_sv1.php, line 579			
22-02-09 22:18:08	ⓘ	fimldap	(init) LDAP configuration found.	class.tx_fimldap_sv1.php, line 78			
22-02-09 22:18:08	ⓘ	fimldap	(init) PHP LDAP extension installed.	class.tx_fimldap_sv1.php, line 64			

Questions and support

If you have any questions about this extension, please ask them in the TYPO3 English mailing list (typo3.english), so that others can benefit from the answers. Please use the bug tracker on forge.typo3.org to report problem or suggest features (<http://forge.typo3.org/projects/extension-devlog/issues>).

Keeping the developer happy

If you like this extension, do not hesitate to rate it. Go the Extension Repository, search for this extension, click on its title to go to the details view, then click on the “Ratings” tab and vote (you need to be logged in). Every new vote keeps the developer ticking. So just do it!

Participating

The devlog is an interesting tool for all TYPO3 developers and as such I am very open to collaborating with others to continue improving this extension. If you are interested go to forge.typo3.org and apply to become a project member. I'll get in touch with you.

Installation

If you run TYPO3 4.1 or more, just download the extension from the TER, install it and get rolling.

If you run TYPO3 4.0 or less, you will need to place the Prototype library yourself somewhere on the server and configure the path to it in the extension's configuration. You have to give the full path starting from your site root. Note that this is necessary only if you want to use the feature for viewing the extra data (see User Manual below), but it's a useful one...

Read the paragraph below if you are already using cc_devlog.

Switching from cc_devlog

If you are already using cc_devlog and want to switch to devlog without losing your existing logs, you can simply go to your favorite database administration tool and manually rename table tx_ccdevlog to tx_devlog **BEFORE** installing devlog. Obviously you should then remove cc_devlog.

Configuration

A number of parameters can be configured for this extension in the Extension Manager. They are described below:

- **Maximum number of log runs:** the backend module always displays a list of log runs to select from. As this can get quite long over time, this parameter makes it possible to cap the number of log runs displayed. If set to 0, all log runs are displayed.
- **Number of entries per page:** when viewing all log entries, a pagination system is activated to avoid displaying all entries on a single screen. This parameter defines the number of log entries to display per page.
- **Maximum number of rows:** this parameter will put an absolute limit on the number of log entries that are stored in the database. When that limit is reached, 10% of the oldest entries are deleted to make room for new ones. If you don't want any limit, set this parameter to zero.
- **Optimize devlog table:** the database table may see a lot of operations, both inserts and deletes. After a time some overhead accumulates. By selecting this option, an optimization is run every time entries are deleted, as per the "maximum number of rows" option above. Note however that this operation uses the "OPTIMIZE" SQL statement which is proprietary of MySQL and is not compatible with DBAL. If you use another database than MySQL you should **not** activate this option.
- **Maximum size of extra data:** as is described in "Developer manual" below, any amount of additional information can be stored in a log entry as a PHP array passed to the variable `$dataVar`. This so-called "extra data" could become quite large in some situations. Writing a very large amount of data to the database may actually crash it. To prevent this from happening, this setting will limit the size of the extra data (once serialised). The default value is 1 million (characters). Any data above this limit will be stripped and replaced by an informational message (see screenshot in "Developer manual" below).
- **Minimum level for logging:** log entries come with a severity that can have the following values: -1 for ok, 0 for an information, 1 for a notice, 2 for a warning and 3 for an error. With this setting you can define from which level you want to start logging entries. All entries with a severity below that setting will be ignored.
- **Excluding keys:** every log entry (normally) comes with a key. This is generally an extension key or some other key in the case of Core scripts, often the name of the calling class. This parameter allows you to define a list (comma-separated) of keys that you don't want to log. This will help keep your log table smaller. A typical example could be: `"realurl,t3lib_userAuth"` as both scripts output a lot of log messages.
- **Highlight style:** when a search is performed, the search for string is highlighted in the BE module. You can customise this highlighting style with this configuration option. Make sure to use only valid CSS properties, as the value of this option is included as is inside a `style=""` attribute.
- **Autorefresh frequency:** it is possible to tell the BE module to refresh itself automatically. This setting makes it possible to define at which frequency (in seconds).
- **Path to Prototype:** for TYPO3 4.0 or less, the path to the Prototype script, as described above.
- **Automatic clean up:** if this is activated, the backend module will automatically delete any log entries that are in excess of the maximum number of log runs defined above. This is convenient to make sure that your log table does not get out of hand. No clean up takes place if the maximum number of log runs is 0. *Note: the new parameter "Maximum number of rows" is much more convenient for limiting the size of the database table as it happens at the time of logging and not when the BE module is accessed. You should use this new parameter. The automatic clean up feature will probably be dropped in the future.*

Developer manual

Using the logging method

To log something simply call `t3lib_div::devLog()`. This method takes the following parameters:

Parameter	Information
\$msg	The message that you want to write to the log
\$extKey	The key of the extension writing to the log
\$severity	Indication of the severity of the message. The following values are expected: <ul style="list-style-type: none"> -1 for ok status 0 for a purely informational message 1 for a notice 2 for a warning 3 for a (fatal) error <p>This parameter is optional and defaults to 0.</p>
\$dataVar	This is an array that can contain any data you wish and that you find useful for information or debugging purposes. It is serialised before being stored in the database. In the backend module it is unserialised and displayed as with <code>t3lib_div::debug()</code> . <p>This parameter is optional and defaults to false.</p> <p>If something else than an array is passed to this parameter, it will be stripped and replaced by an error message (see screenshot below).</p>

A word of caution: if you store a lot of stuff in the `$dataVar` and also if you call the `devlog` very frequently, you may end up with a very large **tx_devlog** table in your database. Check it out regularly and don't hesitate to use the clean up features described below as well as the configuration options outlined above.

Furthermore if the extra data is too large (see "Configuration" above) it will be stripped. This is done to avoid crashing the database when actually writing the log entry to it. It is replaced by a message as in the following screenshot:

Developer Log

Select Log: ☐ Auto refresh
 ☐ Expand all extra data

LOG ENTRIES:
Log period: 26-03-09 15:24 (0 min)

The following keys are currently not logged: t3lib_userAuth,realurl,template,cms

Search log data:

Entries: 0-30

Date ?	Severity ?	Extension ?	Message ?	Called from ?	Page ?	User ?	Extra data ?
26-03-09 15:24:46	0	test	No extra data	index.php, line 200		cobweb	
26-03-09 15:24:46	0	test	Object dump	index.php, line 199		cobweb	The extra data was not an array. It was not stored.
26-03-09 15:24:46	0	test	Big array dump	index.php, line 198		cobweb	The extra data exceeded the maximum size. It was not stored.

More about severity levels

It may not always be easy to choose a severity level. The descriptions below go into a bit more detail and will – hopefully – make the choice easier.

- OK (-1):** These events indicate that everything went fine, no error occurred (at least up to that point where the event was created). No action needs to be taken.
- Info (0):** These events are purely informational. They are normally used for debugging purposes only and require no special action.
- Notice (1):** Abnormal condition, but not blocking. Notices are meant to raise attention. Processes have been completed, but things are not running as smoothly as they could and the condition should be investigated.
- Warning (2):** These events are used to notify significant problems. Processes have been completed, but parts of

them may be missing, wrong or corrupted. Warnings should not be ignored and action should definitely be taken.

- **Error (3):** These events signal that something went fatally wrong. Processes were not completed and action is definitely needed. Alternately this level may be used to point to a failed event, but in a process where failure can be expected, e.g. a login attempt with the wrong password.

Extra information

The devlog extension stores information beyond the parameters passed to the `t3lib_div::devLog()` method. Some of it is automatically retrieved, like the id of the currently logged in BE user, if any.

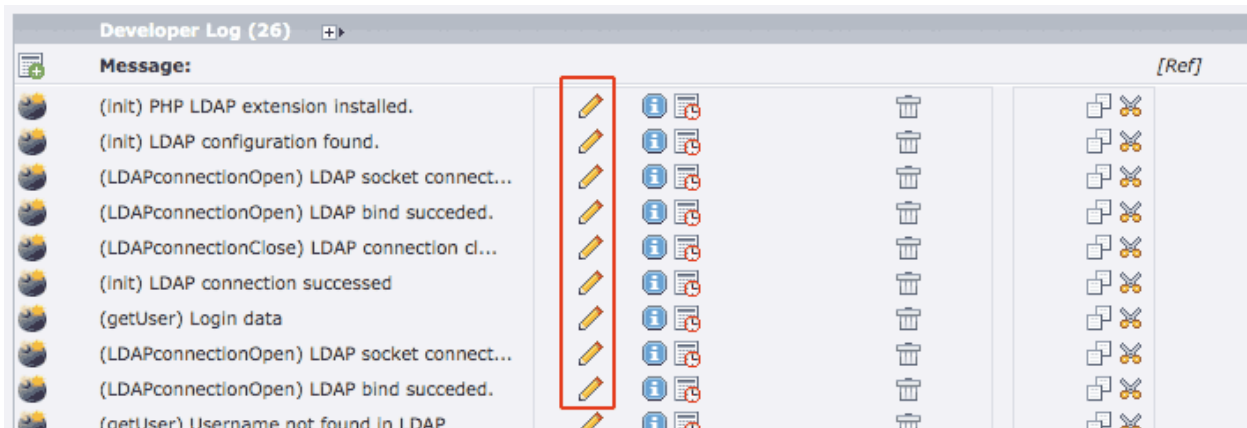
If the devlog call is made in the FE context, the page id will also automatically be retrieved from `$TSFE->id`. This variable is not defined in other contexts, in particular the BE. There's a way to pass a page id to the devlog, but you need to set it yourself in the global variable `$TYP03_CONF_VARS['SC_OPTIONS']['GLOBAL']['debugData']['pid']`. If defined, this variable will be taken into account by the devlog (since version 2.5.0).

User manual

Log records can be viewed individually as described below. While this can sometimes be useful, it is generally more convenient to use the BE module provided by the extension.

Viewing a single entry

Log records are stored in the page where the call happened or at the root of the TYPO3 installation if the call didn't happen within the context of a page. Those records can be seen in the Web > List view. Records are ordered with the most recent on top.



Clicking on the "Edit" icon will open a read-only view of the record displaying all relevant information.

Developer Log [956998] - User setup

Run date and time:
22:26 22-02-2009

Severity:
✓ **Ok**

Extension:
sv

Message:
User setup

Called from:
class.tx_sv_authbase.php
Calling line:
161

Additional data:

table	be_users
userid_column	uid
username_column	username
userid_column	password
usergroup_column	usergroup
enable_clause	AND be_users.pid=0 AND be_users.disable=0 AND be_users.deleted=0 AND (be_users.starttime<=1235338005) AND (be_users.endtime=0 OR be_users.endtime>1235338005)
checkPidList	0
check_pid_clause	AND pid IN (0)

The BE module

The backend module provides a way to visualise all the entries generated by calls to `t3lib_div::devLog()` in a central place, regardless of which page the log entries are related to. Furthermore it provides a tool to clean up those entries. The function menu **(1)** gives access to those two main views.



Viewing the log entries

This screen can be used to filter and browse through the log entries. Many functions are available. First of all, you can choose which entries you want to display **(1)**. This can be either the latest log run or any of the older ones, but also all entries or some of the latest (25, 50 or 100). Checking the "Expand all extra data" box **(3)** will force the module to automatically display all the extra data, rather than having to expand them one by one.

The screen to automatically refresh itself **(2)**, which it then does at an interval of 2 seconds. In conjunction with this autorefresh feature, the screen can be opened in a new window **(4)**.

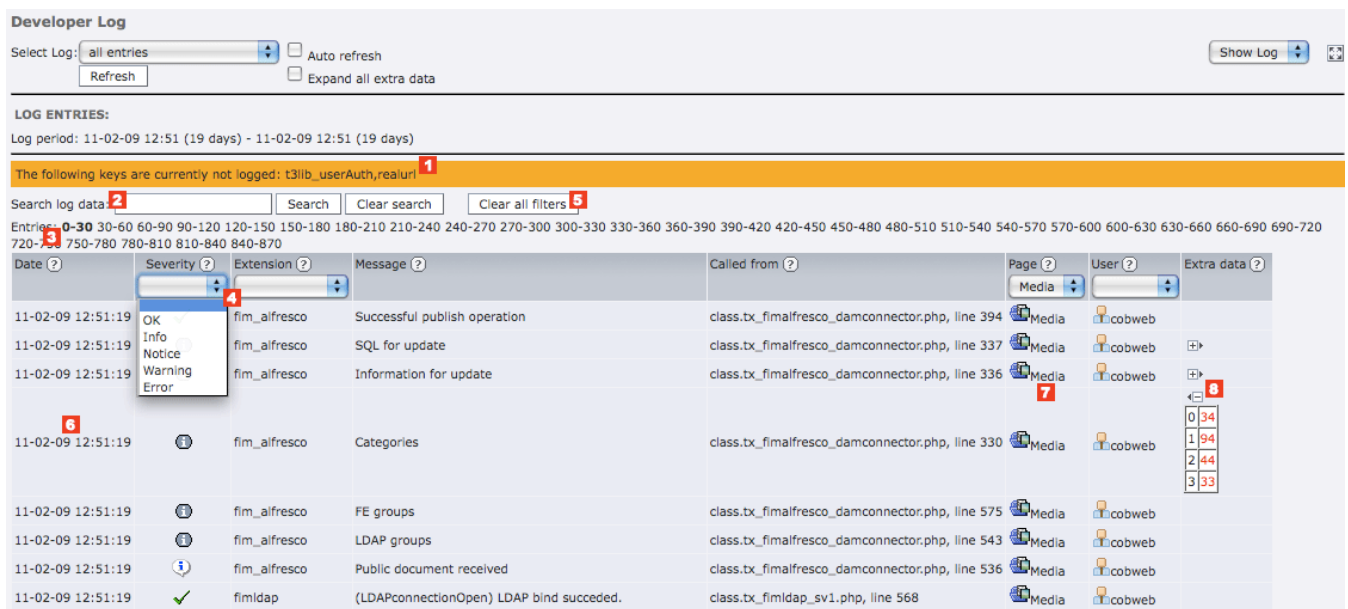


The log view (below) itself offers several more options. If some extension keys are filtered automatically (see Configuration), a reminder is displayed **(1)**.

When viewing all entries you get a lot of options. First of all you can search the message and the extra data **(2)**. Search words are always used with SQL wildcards (%) at the beginning and at the end. When you're done with the search, click on the "Clear search" button. Results are paginated (25 entries a page by default, see Configuration) and you can navigate through them using the page browser **(3)**. What's more various filters **(4)** are available to select entries based on severity or extension key. Filters and search get combined. You can also clear all filters **(5)**.

By clicking on the date of a log entry **(6)**, you can set a filter to display only the log run this entry belongs to. You can also click on the (+) icon to get a view of the deserialised extra data **(8)**. This is displayed inline using JavaScript.

When the log entry is related to a page, you can click on that page **(7)** and you will be taken to the corresponding Web > Page view.



In the results of a search, the keywords are highlighted making them easier to spot.

Developer Log

Select Log: ☐ Auto refresh ☐ Expand all extra data

LOG ENTRIES:
Log period: 11-02-09 12:51 (19 days) - 22-02-09 22:26 (7 days)

The following keys are currently not logged: t3lib_userAuth,realurl

Search log data:

Entries: 0-30 30-60 60-90 90-120

Date ?	Severity ?	Extension ?	Message ?	Called from ?	Page ?	User ?	Extra data ?
22-02-09 22:26:45	✓	sv	User setup	class.tx_sv_authbase.php, line 161			<div> <div>table</div> <div>be_users</div> <div>userid_column</div> <div>uid</div> <div>username_column</div> <div>username</div> <div>userid_column</div> <div>password</div> <div>usergroup_column</div> <div>usergroup</div> <div>enable_clause</div> <div>AND be_users.pid=0 AND be_users.disable=0 AND be_users.deleted=0 AND (be_users.starttime<=1235338005) AND (be_users.endtime=0 OR be_users.endtime>1235338005)</div> <div>checkPidList</div> <div>0</div> <div>check_pid_clause</div> <div>AND pid IN (0)</div> </div>
22-02-09 22:20:35	✓	sv	User setup	class.tx_sv_authbase.php, line 161			

Additionally when viewing a single log run, you get a log run browser (1) to move to the latest, next, previous or oldest log run available.

Developer Log

Select Log: ☐ Expand all extra data

LOG ENTRIES:
Log period: 13-02-09 20:55 (17 days)

The following keys are currently not logged: t3lib_userAuth,realurl



<< oldest < previous 1 next > latest >>

Date ?	Severity ?	Extension ?	Message ?	Called from ?	Page ?	User ?	Extra data ?
13-02-09 20:55:57	i	fimldap	(init) PHP LDAP extension installed.	class.tx_fimldap_sv1.php, line 64			
13-02-09 20:55:57	i	fimldap	(init) LDAP configuration found.	class.tx_fimldap_sv1.php, line 78			
13-02-09 20:55:57	x	fimldap	(LDAPconnectionOpen) LDAP socket connection failed.	class.tx_fimldap_sv1.php, line 579			
13-02-09 20:55:57	x	fimldap	(init) LDAP servers no responded	class.tx_fimldap_sv1.php, line 96			

<< oldest < previous next > latest >>

Cleaning up

This screen allows to delete log entries which can quickly accumulate. You can choose to clear entries for a given extension or if they are older than a given period of time. Alternately you can simply decide to trash them all... When the clean up is done, you get informed on how many entries were deleted.

Clean up  

CLEAR LOG:

Use any of the clean up options below to clear your developer log.

20 log entries were deleted.

There are currently 363 log entries in the Developer Log.

Delete all log entries for extension:

▼

Clear Log

Delete all log entries that are older than:

1 hour ▼

Clear Log

Pressing the button below will actually delete ALL log entries. Make sure that this is really what you want to do!

Clear All Entries!

To-Do list

Here are some of the ideas I have in mind. Feel free to suggest more. Or even better, contact me to participate (see Participating above).

- Change module initialisation to use the framework introduced in TYPO3 4.1
- Add a TYPO3 4.2-style doc header.
- Use PHP 5 feature in microtime() to get more reliable values into the crmsec field

Changelog

DevLog was released at version 2.0.0 to show that it was taking over from cc_devlog. Since no changelog was kept by René Fritz, it starts with the first release of devlog.

Version	Changes:
2.9.2	Added security fix against potential XSS attacks
2.9.0	Added option for optimizing devlog table Switched default database engine of devlog table to InnoDB
2.8.0	Added option for limiting the size of the extra data Added options categorisation (visible only with TYPO3 4.3+) Caught condition of unitialised DB object Caught error when extra data is not an array
2.7.0	Improved view of single records in Web > List view (thanks to Fabien Udriot) Added button for clearing all filters Added warning in BE module about filtered keys Added CSH to single records view and BE module (thanks to Steffen Müller) Changed BE module table layout Changed order of records in Web > List view to have most recent on top (thanks to Steffen Müller)
2.6.0	Added option for limiting the size of the log table Added option for minimum level of severity for logging Added option for filtering (extension) keys Added option for configuring autorefresh frequency Optimized hook usage by making it a (pseudo-)singleton
2.5.0	Added page and user filter Restored link to page feature (was broken) Corrected behaviour of log run browser Added handling of \$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['GLOBAL']['debugData']['pid'] Partial refactoring
2.4.0	Added log search feature
2.3.3	Simplified handling of extra data
2.3.2	Caught error when extension is not configured
2.3.1	Corrected HTML markup error in BE module
2.3.0	Automatically retrieve and store devLog call location (thanks to Kasper Ligaard)
2.2.0	Added various selection options for cleaning up log entries Removed htmlspecialchars() call on log message (suggested by Ingo Renner)
2.1.0	Added a log run browser in single log run view (thanks to Rupert Germann) Added a checkbox to automatically expand all extra data (thanks to Rupert Germann) Added extension configuration options

Version	Changes:
2.0.0	<p>Increased size of database field for extra data</p> <p>Added the possibility of viewing the deserialised extra data</p> <p>Added the possibility of viewing all entries and paginating through them</p> <p>Added filters on status and extension keys</p> <p>Added display of the crdate column (thanks to Stefano Cecere)</p> <p>Cleaned up part of the code</p> <p>Totally revamped manual</p>
1.1.0	Last version of cc_devlog by René Fritz