

Overwriting levels

Each configuration entry has a so called "level". The level defines on which level the configuration value can be overridden.

All configuration values are predefined in the Zf_configuration table. The Zf_configuration table is also the storage for config changes on instance level, which are then customized pre-definitions for config values changeable on a lower level.

The following levels exist:

Level	Level Nr	Storage	Purpose
System	1	Zf_configuration	Config values of this level are not changeable via the GUI, so this is only possible in the DB direct or via installation.ini For example the OpenTM2 instance prefix is an example for such a configuration.
Instance	2	Zf_configuration	Config values of this level will be changeable in the GUI with admin rights. The changed values are stored in the Zf_configuration table and are default values then for the next levels.
Customer	4	to be implemented	Config values of this level will also be changeable in the GUI per each customer. This could be for example specific qm flags for tasks of one customer.
Task	8	to be implemented. first steps are done with the current task template functionality	Config values of this level will be set able per each task via task template. (perhaps over GUI will make sense too). This could be for example specific qm flags for dedicated tasks.
User	16	LEK_user_config table.	Config values of this level will be changeable per each user. For example the customizing of the task grid columns.

The level value in the Zf_configuration table defines to which level a config entry belongs to.

Also for each role in the Zf_acl_rules table must be defined which config level entries can be read and written by the role.



Read defines here reading from the GUI for manipulation, not the internal usage in the application. For example the DeepL Key must only be readable for admin users, but not for a Translator. Although the translator is able to use the DeepL feature in the application.

So an ordinary user has normally access to user level configs which the user may read and write, like customization of visible columns in the GUI.

A PM for example may read and edit the task and customer level entries, and the admin finally may edit all entries on instance level.

On saving config values the target level must be provided, so that backend knows in case of an admin if either the value in his LEK_user_config table (user level) should be saved or if the value in Zf_configuration (instance level) should be changed.