



Workflow Designer

Expressions

This Page describe a feature, which require PHP development skills. If you use this feature in a wrong way, you could crash Workflows!

Often the default functions of the Workflow Designer don't fulfill all requirements of individual companies.

For that reason, you can use expressions inside almost all actions of Workflow Designer and extend them with custom function directly within the configuration

You recognize the option to use this functions, if you have an green border around a textfield or a blue icon next to the textbox. Almost all such fields have special hints about this function. (for example, what value you can to return)

These scripts are default PHP Scripts with limited functions, which are executed within a secure sandbox.

All whitelisted PHP functions have the same arguments, like the original one.

The most important rule: **Each custom function have to return a value with the "return" statement.**

You can use the following PHP functions inside Expressions:

- json_encode, nl2br, date_default_timezone_get, date_default_timezone_set, round, strtotime, md5, rand, count, implode, substr, floor, ceil, explode, microtime, date, trim, time, sha1, hash, intval, floatval, ucfirst, number_format, is_numeric, sqrt, pow, mt_rand, json_encode, json_decode, urlencode, urldecode, password_hash, password_verify, in_array, preg_match, preg_replace, base64_encode, base64_decode
- all functions, started with "str" (strpos, str_pad, str_replace, ...)

To enable the full power of PHP and only blacklist some bad functions, like eval, system, passthrough you can set the variable \$disableFunctionlist to 1. Then the whitelisting switches to a blacklisting and you can access really all PHP functions.

```
$disableFunctionlist = 1;
```

Debug / Log

When the scripts is getting more complex, you would like to have a method to debug variable values.

Or you need to have a option to keep some information within the logs of Workflow Designer

This can be done with the **wf_log(\$variable)** function. So when you use the following:

```
wf_log($accountname);
```

You can check the value of variable *\$accountname* within your workflow statistics.

You can read the value with default Workflow debugging methods: - Enter the workflow analytics / statistics - Click on path execution number - Select a single execution in sidebar - Double click on block to open the debug log from a single execution on this single block

