Command-Line Arguments of KLayout

KLayout's command line basically looks like this:

```
klayout [<options>] [<file>] ..
```

Options start with a hyphen ("-") and can be mixed with file names. Files given on the command line without an option are treated as layout files (GDS, OASIS, ...). Each option must be specified separately, i.e. "-ne" is *not* option "n" and "e". Option arguments must be separated by a space from the option itself. For example:

```
klayout -s file1.gds file2.gds -l layers.lyp
```

This command will open "file1.gds" and "file2.gds" in the same view (option "-s") and use the layer properties file "layers.lyp".

The following is a description of KLayout's command-line options.

General Options

`-b`	Batch mode (same as `-zz -nc -rx`).
`-c` <config file>	Use the specified configuration file (reading it on start and writing it on exit) instead of the default configuration file. This option allows to switch between different configurations.
`-nc`	Don't use a configuration file (implies `-t`).
`-d` <debug level>	Controls the verbosity of the log output. Values are: 0 (silent), 10 (basic info), 11 (basic info plus basic timing), 20 (detailed info), 21 (detailed info plus detailed timing) to 40 or 41 for noisy log output and timing respectively.
`-e`	Enter edit mode even if non-edit mode was specified in the configuration as default mode.
`-ne`	Enter viewer mode even if edit mode was specified in the configuration as default mode. If neither `-e` or `-ne` is specified, the default mode from the configuration will be used.
`-i`	Disable undo buffering (less memory requirements).
`-ni`	Enable undo buffering. This is the default. This option overrides previously set "`-i`" options.
`-j` <path>	Add the given path to the macro project paths.
`-l` <lyp file>	Use the specified layer properties file instead of the default layer properties.
`-lx`	Used with "`-l`": add other layers to the layer properties even if they are not defined in the properties file.
`-lf`	Used with "`-l`": use the ".lyp" file as it is (no expansion to multiple layouts).
`-m` <database file>	Load the given report database together with the previously defined layout. This option must *follow* a layout file argument.
`-n` <technology>	Technology to use for *next* layout(s) on command line.
`-nn` <tech file>	Technology file (".lyt") to use for next layout(s) on command line.
`-p` <plugin>	Load the plugin (a shared object). This option can be used multiple times.
`-r` <script>	Run the given script in interpreter mode. The script file can be either a raw Ruby or Python script, a generic ".lym" script or a domain specific language script such as a DRC script (.drc). In "-r" mode, KLayout will exit after the script is executed. To start the KLayout application, the script must contain a "Application.exec" method call. The script is executed after all other requisites from the command line have been loaded (files, plugins etc.) This option can be combined with "`-z`" (no GUI) or "`-zz`" (no GUI at all). That way, KLayout is converted into a Ruby or Python interpreter.
`-rm` <script>	Run the given script before KLayout starts the user interface. In contrast to "`-r`", KLayout continues normal execution after the script is executed successfully. This is the preferred way to install user interface add-ons ("Modules"). In addition to the modules specified by "`-rm`", KLayout collects script files from various places, i.e. the place specified with `$KLAYOUTPATH` or the installation folder.
`-rd` <name>=<value>	Define the variable in the Ruby or Python context with the given string value. In Ruby, the variable will be accessible as "`$name`".

`-rx`	Ignore all implicit macros ("*.rbm", "rbainit", "*.lym"). This is a "safe" or "fast" mode since errors in those script cannot spoil the application and starting will be faster.
`-s`	Load files into same view.
`-t`	Don't update the configuration file on exit (amnesia mode).
`-nt`	Update the configuration file on exit (default, overrides previous `-t` option).
`-u` <file name>	Restore the session from the given session file.
`-v`	Print program version and exit.
`-wd` <name>=<value>	Define a variable for use within expressions.
`-x`	Synchronous drawing mode (non-threaded). This mode can be useful if scripts are run which produce screen snapshots. By using this option is made sure that all drawing operations have finished before the snapshot method returns.
`-z`	Non-GUI mode. KLayout will not bring up the user interface. See the "`-r`" option for useful applications of this option. Please note that on Linux, this mode still requires a display connection.
`-zz`	Non-GUI mode. In that mode, the GUI is disabled completely. On Linux, this mode allows to use KLayout as a script interpreter for Ruby scripts without a display connection. However, some Ruby objects are not available (i.e. MainWindow).

Special Options

`-gr` <file name>	Record GUI actions in the given file for test purposes.
`-gp` <file name>	Replay the GUI actions from the given file for test purposes.
`-gb` <line number>	Stop replaying GUI actions at the given line for test purposes.
`-gx` <milliseconds>	Replay rate for GUI test file for test purposes.
`-gi`	Incremental logs on the GUI record file (crash safe logging).
`-y` <package name>	Package installation: install package(s) and exit - can be used more than once ('package' is a name, an URL and optionally a version in round brackets)
`-yd`	With -y: include dependencies.

Environment variables

KLayout reads the following environment variables:

- `$KLAYOUT_PATH`: contains a series of search locations separated by a ":" on Linux or ";" on Windows. These are the locations searched for macros, libraries, plugins, ruby modules etc. The first entry of that path will be the one where the configuration is stored when KLayout exits. The installation site (where the KLayout executable resides) is always part of the search path. It cannot be disabled.
- `$KLAYOUT_HOME`: This is the place where KLayout keeps the configuration files, local macros and technology files. By default, this place is "~/.klayout" on Linux and "%HOME%/KLayout" on Windows.
- `$KLAYOUT_SALT_MINE`: Contains the URL the package manager ("Salt Mine") will pull the package list from. By default, the global package index is used. This environment variable can be used to provide a local package index.
- `$KLAYOUT_PYTHONPATH`: Used by the Python interpreter instead of "PYTHONHOME". This path can be used to establish a Python search path.