KiCad installation

The parts of KiCad

KiCad consists of 3 packages:

kicad - KiCad programs and core files.

kicad-doc - Documentation and interactive help (optional package).

kicad-library - KiCad schematic, pcb & 3D-model libraries (optional package).

Installation from binary packages

KiCad binary packages exist for Linux and Windows (XP, 2000).

Data files (schematic, boards, libraries) are compatible with all platforms.

*.zip - KiCad packages for Windows.

*.tbz2 - KiCad for Linux.

Installation from binary packages for Windows

KiCad can be installed in 'C:\kicad', 'D:\kicad', 'C:\Program files\kicad', 'D:\Program files\kicad'.

For example, for an installation in the folder 'C:\kicad', unzip KiCad packages:

```
kicad-{version}.zip
kicad-doc-{version}.zip
kicad-library-{version}.zip
```

to the folder 'C:\kicad'.

The main program is the project manager (kicad.exe) and from it you can run the other programs (schematic editor - eeschema, pcb editor - pcbnew, utilities: cvpcb and gerbview).

You can create a shortcut to 'C:\kicad\bin\kicad.exe'.

Installation from binary packages for Linux

KiCad can be installed in '/usr' or '/usr/local'.

You must have "root" access for installation.

```
cd /
tar -xjf kicad-{version}.tbz2
tar -xjf kicad-doc-{version}.tbz2
tar -xjf kicad-library-{version}.tbz2
```

The main program is '/usr/bin/kicad'.

Windows KiCad tree
----

kicad/bin	- Binaries (executable files).
kicad/doc	- Various documentation.
kicad/doc/help	- Interactive help.
kicad/share/demos	- Sample schematics and printed boards.
kicad/share/internat	- Interface localization files.
kicad/share/library	- Libraries for schematic.
kicad/share/modules	- Module libraries for printed boards.
kicad/share/modules/packages3d	- 3D component models (.wrl and .wings format).

Files '*.mod' are libraries, and files '*.brd' are printed boards you can view with pcbnew.
Files *.brd show the existing modules (and 3D shapes) in libraries.

Linux KiCad tree

KiCad can use the Windows tree or the below tree.
When Kicad is coming from Linux distribution, the tree is:

/usr/bin	- Binaries (executable files).
/usr/share/doc/kicad/	- Various documentation.
/usr/share/doc/kicad/help	- Interactive help.
/usr/share/kicad/demos	- Sample schematics and printed boards.
/usr/share/kicad/internat	- Dictionaries for interface localization.
/usr/share/kicad/library	- Interface localization files.
/usr/share/kicad/modules	- Module libraries for printed boards.
/usr/share/kicad/modules/packages3d	- 3D component models (.wrl and .wings format).

if KiCad does not found its files in usr/share, the search is made in /usr/local/kicad
and kicad uses the same tree as the Windows KiCad tree above

*** When KiCad comes from .tgz archive from http://iut-tice.ujf-grenoble.fr/cao/,
it must me installed in /usr/local and uses the Windows tree ***.

Files '*.mod' are the libraries, and files '*.brd' are printed boards you can
view with pcbnew.
Files *.brd show the existing modules (and 3D shapes) in libraries.

Warning:
Do not change the KiCad tree, or the location of binary files,
else KiCad will not be able to find its required files (configuration, libraries, etc.).

Mac OS X KiCad tree

System wide files

/Library/Application Support/kicad/demos

```
/Library/Application Support/kicad/internat
/Library/Application Support/kicad/library
/Library/Application Support/kicad/modules
/Library/Application Support/kicad/modules/packages3d
```

User files can be the same as the system wide files but only inside the users home directory.

```
$HOME/Library/Application Support/kicad
```

Warning:
These paths are hardcoded into KiCad, if you put them somewhere else KiCad will not find them when a new project is created.

Installation from source code

Some dependencies must be satisfied for the correct installation of KiCad:

```
under Linux:
wxWidgets           >= 2.8.11        http://www.wxwidgets.org/
under Windows:MacOSX
wxWidgets           >= 2.9.3        http://www.wxwidgets.org/
CMake               >= 2.6.4         http://www.cmake.org/
Boost C++ Libraries  (files used by kicad are provided in kicad sources)
http://www.boost.org/
OpenGL
  Linux:   Mesa 3D Graphics Library  http://www.mesa3d.org/
  Windows: built-in
Zlib Compression Library             http://www.zlib.net/
```

In source-tree-build are mostly unwanted, so make a subdir called "build" and change to it.

Call cmake with the path to KiCad. E.g., when your build-folder is "build" within source-tree, type "cmake ../".

Now your system get checked if it is able compiling KiCad and cmake generates the Makefiles.

After calling cmake just type "make" and build begins.

It is easy to build only a specific binary such as pcbnew alone:

```
 make pcbnew
```

After "make" type "make install" and install begins.

You may install to a temporary-root with

```
 make install DESTDIR=<temproot>
```

If you want to uninstall KiCad again type "make uninstall" from within the build directory.

Important parameters to cmake

-DCMAKE_BUILD_TYPE=<buildtype>
<buildtype> may current one of "Debug" and "Release".

-DCMAKE_INSTALL_PREFIX=<prefix>
Default to "/usr/local".

-DwxWidgets_ROOT_DIR=<wxInstallDir>
Required for Windows platform.

-DwxWidgets_USE_DEBUG=ON
Can be used only with -DCMAKE_BUILD_TYPE=Debug

-DwxWidgets_USE_STATIC=ON
For building statically linked executables. Can be used only if wxWidgets configured and builded with "--enable-monolithic --disable-shared" parameters.

-DwxWidgets_USE_STATIC=OFF
For building dinamically linked executables. Can be used only if wxWidgets configured and builded with "--disable-monolithic --enable-shared" parameters.

-DwxUSE_UNICODE=ON
Require on locale utf8 for build the KiCad with cyrillic fonts support.

-DKICAD_GOST=ON
Build the KiCad with russian GOST support.

-DKICAD_KEEPCASE=ON
Build the KiCad with no component name conversion to uppercase (if you want your
ADuC.../Si.../bq... components named as just so).

-DCMAKE_CXX_FLAGS=<some extra flags>
Extra flags for the c++ compiler for your system required.

-DCMAKE_VERBOSE_MAKEFILE=ON
When more output is wanted use this cmake parameter or call "make VERBOSE=1".

Extra CFLAGS and linker flags

If you require extra flags for compiler and linker you may give them via environment variables

```
  "CXXFLAGS" (c++ compiler)
  "LDFLAGS"  (for linker)
  "CFLAGS"   (for c-compiler, not needed in kdesvn build)
```

eg., it may usefull on 64bit systems "-m64" to CXXFLAGS and LDFLAGS.