kitlist

Building

kitlist - a program to maintain a simple list of items and assign items to one or more categories.

Required Packages for Desktop Build

On a Debian 5.0 (Lenny) system, the following packages need to be installed to build the application:

- intltool
- libgconfmm–2.6-dev
- libgtkmm–2.4-dev
- libpqxx-dev (pgsql only)
- libtool
- libxml++2.6-dev
- libglademm–2.4-dev

Required Packages for Maemo Build

On the Maemo platform, the following packages are required:

- intltool
- libgconfmm–2.6-dev
- libgtkmm–2.4-dev
- libhildonmm-dev
- libhildon-fmmm-dev
- libtool
- libxml++2.6-dev
- libglademm–2.4-dev

Version 0.6.2 has been confirmed to compile on the following platforms:

- Debian 4.0 (Etch)
- Debian 5.0 (Lenny)
- Ubuntu 8.04 (Hardy Heron)

C++ Compiler Flags

Set the environment variable CXXFLAGS when running ./configure. E.g.

```
$ CXXFLAGS="-g -O0 -fno-inline" ./configure
```

to include debug messages, define KITLIST_DEBUG

```
$ CXXFLAGS="-g -O0 -fno-inline -DKITLIST_DEBUG" ./configure
```

Obtaining the required libraries in the Maemo scratchbox environment

1. Add the following repositories to the scratchbox configuration:

   ```
   cat >> /etc/apt/sources.list.d/maemo-extras.list <<EOF
   deb http://repository.maemo.org/extras/ diablo free non-free
   deb http://repository.maemo.org/extras-devel/ diablo free non-free
   deb-src http://repository.maemo.org/extras/ diablo free non-free
   deb-src http://repository.maemo.org/extras-devel/ diablo free non-free
   EOF
   ```

2. Install the required packages:

   ```
   $ fakeroot apt-get update
   $ fakeroot apt-get install maemo-explicit maemo-cplusplus-env
   $ fakeroot apt-get install libglademm-2.4-dev libgtkmm-2.4-doc intltool
   ```

3. To re-generate the GNU configure files, run the following command:
   To build for deployment to a Maemo platform (E.g. Nokia N810 Tablet)

   ```
   $ ./autogen.sh
   ```

   For the default Linux desktop build:

   ```
   $ ./autogen.sh
   ```

Note: The warnings generated by aclocal are usually just warning of deprecated behavior.

4. Then you can build the application with:

```
$ ./configure
$ make
```

Note: Building the docs requires doxygen and pandoc. See the Documentation section.

Running in Maemo Scratchbox Environment

1. Xephyr is used to provide an X server based emulation window. Install the xserver-xephyr package in Debian 4.0 (Etch).
2. Run the Xephyr server from a normal user terminal:

   $ Xephyr :2 -host-cursor -screen 800x480x16 -dpi 96 -ac -extension Composite

3. Login into scratchbox environment, build and run with:

```
$ /scratchbox/login
$ export DISPLAY=:2
$ af-sb-init.sh start
$ cd ~/kitlist
$ ./autogen.sh
$ ./configure
$ make
$ run-standalone.sh ./src/kitlist
```

4. In the scratchbox environment, build a release with:

```
$ dpkg-buildpackage -rfakeroot -b -tc
```

Note: See the Debian Package Builds section for more information.

5. and install it with:

```
$ fakeroot dpkg --install ../kitlist_X.X.X_ARCH.deb
```

6. uninstall with:

```
$ fakeroot dpkg --remove kitlist
```

7. purge with:

```
$ fakeroot dpkg --purge kitlist
```

See `http://wiki.maemo.org/Scratchbox_C++`

Maemo Runtime Library Dependencies

- libgtkmm–2.4–1c2a
- libxml++2.6c2a
- libglademm–2.4
- libgconfmm–2.6–1c2
- libhildon-fmmm
- libhildonmm

Debian Package Builds

Two configurations are provided for Debian packaging in folders named 'debian.debian' and 'debian.maemo'. During 'make' the appropriate folder is copied to a folder named 'debian', only if the folder does not already exist. It is not removed during 'make distclean', because dpkg-buildpackage calls 'make distclean' at the start of the build process, otherwise removing the directory if we included it's deletion in 'make distclean'. The folder can be removed with 'make debclean'.

This situation is less than ideal, producing the possibility of inconsistent builds, but I can't see a better solution that does not result in maintaining separate sources.

Building for Windows

The application will build on a Windows platform using MinGW. It's not currently documented and not entirely straight-forward, but can be done, although I have not as yet been able to build the application's alternative language files. However, this fails after the build is otherwise complete, so interrupting the looping make file with Ctrl-C does the trick. In any event you need to:

1. Install the MinGW development environment
2. Install various MinGW packages (TODO:: document which packages) to support the build
3. Install version 2.16 of gtkmm for Windows
4. compile with './configure

Cross Compilation on Linux

The application can be cross-compiled on Linux for a Windows target. These notes are based on instructions for Cross-compiling GTK+ apps for Windows

Setup the tool chain following the instructions on the MinGW Wiki.

The following settings in x86-mingw32-build.sh.conf worked for me:

```
assume GCC_VERSION                  3.4.5-20060117-2
assume BINUTILS_VERSION             2.19.1
assume RUNTIME_VERSION              3.14
assume W32API_VERSION               3.13-mingw32
```

Execute the mingw32 build script with an appropriate target. E.g.:

```
$ sh x86-mingw32-build.sh i686-pc-mingw32
```

Download the gtkmm developer bundle and install it in a new folder using Wine (or Windows), then copy the contents to /opt/mingw/i686-pc-mingw32.

Fix the package config files to have the correct prefix and rename the DLLs.

```
cd /opt/mingw/i686-pc-mingw32
sed -i 's|^prefix=.*$|prefix=/opt/mingw/i686-pc-mingw32|g' lib/pkgconfig/*.pc
cd ./lib
for f in *.lib; do mv $f lib${f%%lib}a; done
```

Finally, build the kitlist application as follows:

```
$ export PATH=/opt/mingw/bin:$PATH PKG_CONFIG_PATH=/opt/mingw/i686-pc-mingw32/lib/pkgcon
$ ./configure --prefix=/opt/mingw/i686-pc-mingw32/ --host=i686-pc-mingw32 --build=i686-p
$ make
$ makensis kitlist.nsi
```

Note: The application does not run under Wine.

Useful Links

- http://www.gtk.org/download-windows.html
- http://live.gnome.org/gtkmm/MSWindows/BuildingGtkmm

Environment Variables

The application can optionally be compiled to use a PostgreSQL database instead of XML documents, using `./configure --disable-xml-dao`. In this case there are a number of environment variables that can be used to specify various connection parameters to the PostgreSQL database. These are listed in the PostgreSQL Documentation. Some of them are mentioned briefly below:

Example Environment Variables

- PGHOST - The database server name
- PGPORT - The port to use
- PGDATABASE - The database name
- PGUSER - The database user name
- PGPASSWORD - The connection password

Internationalisation

1. Translatable strings contained in the program have been written in American English. To create a translation for another language, go to the po subdirectory and run the following command to update the default language file ./po/kitlist.pot:

   ```
   $ intltool-update --pot
   ```

2. Copy this file to languagecode.po, e.g. fr.po. This file contains pairs of strings, one in the default language, the other the translated version, initially blank. Also add the language to the list in the ./po/LINGUAS file, and the ALL_LINGUAS entry in ./configur.ac.

3. To merge code changs into a translated po file, e.g. French:

   ```
   $ intltool-update fr
   ```

4. Re-build and re-install the program. To specify the language in a shell, specify the LANG environment entry, e.g.:

   ```
   $ export LANG=fr_FR.UTF-8
   ```

Note: the kitlist program must be installed before the language files are picked up at runtime.

More information in Programming with gtkmm

Documentation

The documentation for the code is maintained using Doxygen. The documentation is regenerated from the source code as part of the build. You can disable building the docs by passing `--disable-build-docs` to configure. The documentation is not built by default for the Maemo platform.

The generated Doxygen documentation can be viewed under ../doc/doxygen/

License

The source code and documentation are licensed under the GPL. See the COPYING and AUTHORS files distributed with the source code for information and contact details.