

Mobile Broadband Configuration Assistant for
NetworkManager

Closing Report
Kesäkoodi 2008

Antti Kaijanmäki
<antti@kaijanmaki.net>

September 8, 2008

Contents

1	Abstract	2
2	Realization	3
2.1	Schedule	3
2.2	Deliverables	3
2.2.1	Assistant Program - libmbca	3
2.2.2	Service Provider Database - mobile-broadband-provider-info	4
2.2.3	Documentation	4
2.2.4	Translations	4
2.2.5	Packaging	4
2.3	Hardware Support	4
2.4	Test Results	4
3	Problems and Shortcomings	5
3.1	NetworkManager	5
3.2	Toolset	5
3.3	Libraries and Components	5
3.4	Communication and Community	5
4	Word from Summercoder	7
5	Word from Mentor	8
6	Future Enchantments	9
6.1	libmbca	9
6.2	mobile-broadband-provider-info	9
6.3	NetworkManager	9
	APPENDIX	10
A	libmbca API	11
B	mobile-broadband-provider-info DTD	14
C	Thanks to	15

1 Abstract

This document contains the closing report for Mobile Broadband Configuration Assistant for NetworkManager – a project for Kesäkoodi 2008. Development was executed during the period of 3 months which started on May 26th and ended on August 22nd. Development was executed according to project plan¹ approved in March 2008.

¹project plan is downloadable from <http://www.kaijanmaki.net/kesakoodi/projectplan.pdf>

2 Realization

This section describes the realization of the project. See project blog archives² to find more details on implementation and to see how the project proceeded during the summer.

2.1 Schedule

The project was executed in 13 weeks according to schedule that was set in project plan. The schedule was accurate and did not slip. Planned usability testing did not take place because of illness of a family member.

Here's the planned schedule from the project plan:

wk22	project kickoff set up the development environment
wk23	design the UI
wk24	specify the XML schema format for the service provider information develop the parser for the service provider information integrate the XML parser to the UI
wk25	implement the UI navigation implement the general UI functions implement the serial page
wk26	implement the USB page implement the Bluetooth page
wk27	testing localization time for unfinished tasks
wk28	usability testing
wk29	reacting on usability testing feedback
wk30 wk31 wk32 wk33	time for unfinished tasks integration with nm-applet implement NetworkManager's missing features
wk34	writing the documentation packaging project ending

2.2 Deliverables

The project plan listed 5 different deliverables for the project.

2.2.1 Assistant Program - libmbca

The assistant was realized as a library called *libmbca* that provides a GObject based API. Source code is available from *svn.gnome.org*³. See appendix A for reference.

²<http://www.kaijanmaki.net/blog/category/kesakoodi>

³\$ svn co <http://svn.gnome.org/svn/libmbca>

The assistant meets the requirements stated in the project plan. Some minor changes were made to improve the overall usability.

2.2.2 Service Provider Database - mobile-broadband-provider-info

The database was realized as a single XML file containing service provider information for multiple countries. The database is available from *svn.gnome.org*⁴. See appendix B for Document Type Definition that describes the database format.

The database was based on an existing one from *GPRS Easy Connect*⁵ project which has been closed and is lacking maintenance and development; mobile-broadband-provider-info did receive almost 400 different entries as a legacy, but many of them are outdated. There has already been some effort to get the new database up to date and some new service providers have also added already.

2.2.3 Documentation

Documentation is provided with the source package. There's also a test program that also acts as an example of how to use the library.

2.2.4 Translations

libmbca is hosted at *svn.gnome.org* repository and thus translations are handled by *GNOME Translation Project*⁶. Currently there are translations for 24 languages available from the SVN repository.

2.2.5 Packaging

Database and assistant have been separated to different packages. Packages for Ubuntu 8.04 and 8.10 are available from *Personal Package Archive* of Antti Kaijanmäki⁷.

2.3 Hardware Support

Elisa provided hardware for testing and all the devices should be supported out of the box in Ubuntu 8.10. The devices included Huawei E169, Option iCON255, Vodafone VMC HSDPA 3G and Nokia E81.

2.4 Test Results

Usability testing was canceled due family member's illness, but all user feedback gathered otherwise during the summer has been mainly positive.

⁴`svn co http://svn.gnome.org/svn/mobile-broadband-provider-info`

⁵<http://easyconnect.linuxuser.hu>

⁶<http://www.gnome.org/i18n>

⁷<https://launchpad.net/~kaijanmaki/+archive>

3 Problems and Shortcomings

Although all the deliverables were provided there was some problems along the way. All the problems could not be solved without compromises or shortcomings.

3.1 NetworkManager

Although *libmbca* supports all the device types that were listed in the project plan, NetworkManager does not. There was not enough time to implement the missing device types in NetworkManager and proper implementations most likely require some heavy changes in NM. NM currently supports only USB, PCMCIA, PC Card, ExpressCard and built-in devices and there's no support for Bluetooth and old fashioned serial ports.

3.2 Toolset

There was some problems anticipated with toolset and this risk was realized. The most troublesome component in the toolset was Glade3 Interface Designer. The GUI parts of *libmbca* were indeed designed with Glade3, but it had to be customarily patched and some changes were performed by hand to the resulting UI Definition file.

Glade3 is part of GNOME development toolset and thus it's highly probable that the problems encountered and worked around will be addressed in the near future.

At the moment Glade3 can not and must not be used to modify the UI.

3.3 Libraries and Components

There were some bugs encountered in the GTK library. Summercoder personally tracked down, reported and provided fixes for multiple bugs during the summer.⁸

Some required functionality was already implemented by other projects, but no public API was available. As there was little time and resources to spend with external projects a decision to include a private copy of some components from external projects was made. It's taken into account that having external code included is going to add maintenance burden, but in this situation it's acceptable as long as proper libraries of the components are not released.

External components are clearly separated in the source tree.

3.4 Communication and Community

Communicating with the mentor was not very fluent. The main reasons for this were the time difference between the Summercoder and the mentor and the mentor was very busy with his own work on NM. Having a time difference of over six hours did make instant messaging very hard to arrange and email conversation was also very slow. This was not fatal though as the project was highly autonomous to the point of integration with NetworkManager.

Community was invited to participate in the discussion of database format and some other parts of the design. There was very little input from the

⁸<http://bugzilla.gnome.org>, bugs [#540723](#), [#543545](#), [#545982](#), [#546378](#)

community, but all the decisions were done after all the options were carefully evaluated.

4 Word from Summercoder

View and opinions of the Summercoder about the project:

Kesäkoodi was great as it allowed me to finally realize a project that I had planned for a couple of years. I clearly saw a need for the project and the benefits that come out from it. Now after the summer I'm generally really pleased of the outcome and the results, but as always there are some things that did not go so swell.

The most problematic thing of the summer was the process of getting me an account to `svn.gnome.org`. The account was needed in order for me to set up public repositories of my work. I had a plan to release early and often, but getting the account took 8 weeks. At that point the library was already finished.

My son was quite sick a total of couple of weeks during the summer. This naturally prevented me from working and lead to postponing of usability testing.

One thing I really was rather disappointed to was the lack of community interest towards my project. I truly believed the project would have caused more people to lift their eyebrow and participate to the process. Of course there were a couple of people who did, but not that many. This lead to lack of faith time to time and I thought that my project was not an important one or there's no true need for it. Hopefully I have managed to get my point through and it's now clear to everyone that there indeed is a need.

Now the last goal is to get distributions to adapt the project. Work towards Ubuntu inclusion has already started and as of today it seems that Ubuntu 8.10 will be released in October and it will ship the assistant by default. I just hope there will be enough time for me to work on any defects and bugs should any emerge.

5 Word from Mentor

Mentor Dan Williams told the following when asked about his view and opinions about the project:

In the big picture there's definitely a need for the library and database you've developed. Efforts were made in the past, but none of them were very comprehensive and they eventually just died because they didn't get widely used and the author apparently gave up. The other major PC and mobile phone OSes (including the Nokia Internet Tablets) have this functionality and its well integrated into their network configuration interfaces. Having this functionality on Linux is a benefit.

The split between the library and the database is also smart because the data itself is useful to more than one project. Politics is unfortunately a part of open source projects, and walking the fine line between different technologies (and providing libraries built on more than one of those technologies) is important if you want people to use your code. I think you've been able to do that so far, though you'd have to do a QT-based libmbca that knetworkmanager could use to fully complete that goal.

It's good to get a fresh start, build that momentum, get user submissions, integrate with the desktop, and push the ease of configuration forward. You seem to have attracted good contributions so far, and you've been able to build up some momentum via blogging and getting the library accepted for at least one Linux distribution. That shows that you are willing to do "the last 20%", which is to get distros and people to actually care about your project after you've written some code.

The code itself looks fairly well-structured and demonstrates a clear understanding of how to use a number of different libraries in the GNOME stack (gtk, libhal, glib, gobject, libxml). You appear to have a good handle on API nuances like allocation behavior. One suggestion for improvement is more pervasive use of GError for richer error handling and reporting; while not necessary it's an indication of a mature and/or well-designed API. Obviously libmbca is quite new so lack of this isn't a mark against you.

I believe your communication was sufficient. Any real deficiencies were probably my fault.

Dan

6 Future Enchantments

6.1 libmbca

Current NetworkManager integration utilizes only service provider selection part of libmbca; device selection related parts are not used. If there's not going to be any other projects that utilize libmbca it might make sense to remove device selection altogether. Thus libmbca would only provide an frontend to mobile-broadband-provider-info. Functionality that no one uses only increases maintenance burden.

6.2 mobile-broadband-provider-info

Most obvious goal for the future is to update and increase the number of provider entries in the database.

The database was designed to allow other projects to utilize it as well. Wader project⁹ has shown interest to adopt and share the database if a *netid*¹⁰ element is added to the database format.

6.3 NetworkManager

Hopefully NetworkManager can utilize the device selection bits of libmbca in the future. There's a lot of work to be done in order to support Bluetooth devices and the support for arbitrary serial ports might never get developed as they are against the current design principles¹¹ of NM.

⁹<http://public.warp.es/wader>

¹⁰<http://public.warp.es/wader/browser/trunk/resources/extra/networks.py>

¹¹NM is designed to only support devices that can be automatically detected. User telling NM there's a device at some location is against the principles.

APPENDIX

A libmbca API

Essential parts of the public API from files *mbca_assistant.h* and *mbca_serviceprovider.h*

```
/* This file is part of Mobile Broadband Configuration Assistant.
 *
 * Mobile Broadband Configuration Assistant is free software:
 * you can redistribute it and/or modify it under the terms
 * of the GNU General Public License as published
 * by the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * Mobile Broadband Configuration Assistant is distributed in the hope that it
 * will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty
 * of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with Mobile Broadband Configuration Assistant.
 * If not, see <http://www.gnu.org/licenses/>.
 */

/**
 * @brief cellular network type
 *
 * Cellular network a provider is using. Network type helps to choose the right
 * dial sequence as data modems are standardised.
 */
enum MBCANetwork
{
    MBCA_NETWORK_GSM,      /**< GSM network, means GPRS, UMTS, etc */
    MBCA_NETWORK_CDMA     /**< CDMA network */
};

/**
 * @brief GSM information
 *
 * GSM network specific information.
 */
struct _mbcaGSMInfo
{
    gchar* apn;           /**< Access Point Name */
};
/** standard typedef */
typedef struct _mbcaGSMInfo mbcaGSMInfo;

/**
 * @brief service provider name
 *
 * Mobile Broadband Service Provider Database supports multiple names in
 * different languages for a service provider and this struct contains name of
 * the service provider in some language.
 */
struct _MBCAServiceProviderName
{
    gchar* lang;         /**< language, contains xml:lang attribute as defined in
 * database */
    gchar* name;        /**< name of the service provider */
};
/** standard typedef */
typedef struct _MBCAServiceProviderName MBCAServiceProviderName;

/**
 * @brief service provider data
 *
 * Contains the data of one service provider. In general NULL means
 * "not needed" or "undefined". Fields correspond to Mobile Broadband Service
 * Provider Database format.
 *
 * @see http://live.gnome.org/NetworkManager/MobileBroadband/ServiceProviders
 */
struct _MBCAServiceProvider
{
    GSList* names;      /**< List of _MBCAServiceProviderName structs.
 * Contains names as defined in the database. */
    enum MBCANetwork type; /**< cellular network the provider is using */
    union
    {
        mbcaGSMInfo gsm; /**< contains GSM network specific settings if
 * _MBCAServiceProvider::type is MBCA_NETWORK_GSM */
    };
    /**< union containing network specific settings.
 * _MBCAServiceProvider::type indicates valid
 * member */

    gchar* username;    /**< if NULL, no authentication required */
    gchar* password;    /**< if NULL, no password required. */

    gchar* dns1;        /**< primary DNS, NULL if automatically acquired */
    gchar* dns2;        /**< secondary DNS, NULL if none. */
    gchar* gateway;     /**< gateway to use, NULL if none */
};
/** standard typedef */
typedef struct _MBCAServiceProvider MBCAServiceProvider;

/**
```

```

* @brief type of device
*
* Type of the device (and connection).
*/
typedef enum /*< skip >*/
{
    MBCA_DEVICE_NONE, /*< no device selected, for initialization */
    MBCA_DEVICE_BLUETOOTH, /*< Bluetooth device */
    MBCA_DEVICE_HAL, /*< devices discovered using HAL */
    MBCA_DEVICE_SERIAL, /*< Serial device */
    MBCA_DEVICE_PSEUDO /*< pseudo device, not shown on summary page */
} MBCADeviceType;

/**
 * @brief connection information
 *
 * Contains all the information needed for establishing a connection.
 */
struct _MBCAConfiguration
{
    gchar* name; /*< name of the connection */
    MBCADeviceType type; /*< type of the connection */
    gchar* device; /*< device to use, format depends on type
 * @see mbca_assistant_run_for_device() */
    gchar* baud; /*< baud rate to use with the device */

    MBCAServiceProvider* provider; /*< service provider specific settings */
};
/** standard typedef */
typedef struct _MBCAConfiguration MBCAConfiguration;

/**
 * @brief free configuration
 *
 * deallocates previously allocated MBCAConfiguration.
 *
 * @note this function should be used only with configurations that are build
 * in mbca_assistant_run() or mbca_assistant_run_for_device().
 *
 * @param configuration configuration to be freed
 */
void
mbca_free_configuration (MBCAConfiguration* configuration);

/**
 * @brief states of MBCAAssistant
 */
typedef enum /*< prefix=MBCA, underscore_name=mbca_assistant_state >*/
{
    MBCA_STATE_READY, /*< Assistant is ready to be run */
    MBCA_STATE_RUNNING, /*< Assistant is running */
    MBCA_STATE_ABORTED, /*< Assistant was aborted */
    MBCA_STATE_DONE /*< Assistant has finished */
} MBCAAssistantState;

/**
 * @brief create new instance
 *
 * Creates new instance of MBCAAssistant. The instance must be unreferenced
 * when not anymore needed with g_object_unref()
 *
 * @note gtk_init() and g_thread_init() must be called before this function
 *
 * @returns a newly allocated instance.
 */
MBCAAssistant*
mbca_assistant_new ();

/**
 * @brief run fully
 *
 * Runs the assistant fully. The assistant contains 5 pages:
 * 1. introduction
 * 2. connection method
 * 3. device page based according to method
 * 4. service provider
 * 5. summary
 *
 * actually calls:
 * <code>
 * mbca_assistant_run_for_device (assistant,
 * MBCA_DEVICE_NONE,
 * NULL, NULL);
 * </code>
 *
 * @see mbca_assistant_run_for_device()
 *
 * @note assistant must be in MBCA_STATE_READY state
 *
 * @param assistant assistant to run
 *
 * @returns '0' on success
 * @returns '-1' on failure
 */
gint
mbca_assistant_run (MBCAAssistant* assistant);

```

```

/**
 * @brief run partially
 *
 * Runs the assistant. If device type is MBCA_DEVICE_NONE assistant is
 * run fully. Otherwise assistant skips device selection and service provider
 * page follows directly after introduction page. If device type is
 * MBCA_DEVICE_PSEUDO device field on summary page is also hidden.
 *
 * After assistant is once run it can't be run again. Unref the old and create
 * a new one if necessary.
 *
 * format for device depends from type:
 *
 * MBCA_DEVICE_BLUETOOTH
 * - device is Bluetooth device address
 * - e.g. "01:23:45:67:89:AB"
 *
 * MBCA_DEVICE_HAL
 * - device is HAL UDI of a device that belongs in info.category "serial"
 * - e.g. "/org/freedesktop/Hal/devices/usb_device_421_453_noserial_if0_3_
 *   serial_unknown_0"
 *
 * MBCA_DEVICE_SERIAL
 * - device is path to character device representing serial port. For serial
 *   devices nice_device_name means baud rate!
 * - e.g. "/dev/ttyS0", "9600"
 *
 * MBCA_DEVICE_PSEUDO
 * - device and nice_device_name are ignored
 *
 * @see mbca_assistant_run()
 *
 * @note assistant must be in MBCA_STATE_READY state
 *
 * @param assistant assistant to run
 * @param type device type to preset
 * @param device device to preset
 * @param nice_device_name name for the device that is displayd to the user
 *
 * @returns '0' on success
 * @returns '-1' on failure
 */
gint
mbca_assistant_run_for_device (MBCAAssistant* assistant,
                              MBCADeviceType type,
                              const gchar* device,
                              const gchar* nice_device_name);

/**
 * @brief get state
 *
 * @param assistant assistant to get state from
 *
 * @returns assistants state
 */
MBCAAssistantState
mbca_assistant_get_state (MBCAAssistant* assistant);

/**
 * @brief get configuration
 *
 * @note assistant must be in state MBCA_STATE_DONE
 * @note callee is responcible of calling mbca_free_configuration() when
 *   returned configuration is no longer needed.
 *
 * @param assistant assistant which configuration to retrieve
 *
 * @returns configuration of user selection
 */
MBCAConfiguration*
mbca_assistant_get_configuration (MBCAAssistant* assistant);

/**
 * @brief abort assistant
 *
 * @param assistant assistant to abort
 *
 * aborts assistant if in RUNNING state and sets state to MBCA_STATE_ABORTED
 */
void
mbca_assistant_abort (MBCAAssistant* assistant);

/**
 * @brief presents assistant to user
 *
 * Presents an already running assistant to the user.
 *
 * @param assistant assistant to present
 */
void
mbca_assistant_present (MBCAAssistant* assistant);

```

B mobile-broadband-provider-info DTD

Here's the Document Type Definition for the database:

```
<!ELEMENT serviceproviders (country*)>
<!ATTLIST serviceproviders format CDATA #REQUIRED>

<!ELEMENT country (provider*)>
<!ATTLIST country code CDATA #REQUIRED>

<!ELEMENT provider ( name+,
                    (gsm|cdma),
                    username?,
                    password?,
                    dns*,
                    gateway?)>

<!ELEMENT name (#PCDATA)>
<!ATTLIST name xml:lang CDATA #IMPLIED>

<!ELEMENT gsm (apn)>
<!ELEMENT apn (#PCDATA)>

<!ELEMENT cdma EMPTY>

<!ELEMENT username (#PCDATA)>
<!ELEMENT password (#PCDATA)>
<!ELEMENT dns (#PCDATA)>
<!ELEMENT gateway (#PCDATA)>
```

.....
An example of how an actual database could look like:

```
<?xml version="1.0"?>
<!DOCTYPE serviceproviders SYSTEM "serviceproviders.dtd">
<country code="fi"> <!-- Finland -->
<provider>
<name>Saunalahti</name>
<gsm>
<apn>internet.saunalahti</apn>
</gsm>
</provider>
</country>
<country code="tw"> <!-- Taiwan -->
<provider>
<name>Asia Pacific Telecom (APBW)</name>
<cdma />
</provider>
</country>
</serviceproviders>
```

C Thanks to

my wife, Dan Williams, Alexander Sack, Ruben Romero, Simon Péter, Ross Barkman, Kai Keinänen, Manu Setälä, Bastien Nocera, Niklas Laxström, Siebrand Mazeland, Teemu Merikumpu, Juha Hytönen, Paul Tötterman, Jari Huilla ...

... and all the other people who showed interest towards my work and provided insight during the summer.