Maemo: Hildonizing application UI

This article is not meant as a complete porting guide to Maemo platform, but rather a collection of things that need to be addressed in order for a GTK+ application to be compatible with *Hildon*, the graphical user interface used in Maemo. Although most GTK+ widgets and methods work as-is under maemo GUI environment, some source code changes are required in order to maintain a consistent application look-and-feel and operation.

HildonProgram

A HildonProgram widget, inherited from GObject, is the base of Hildon applications. HildonProgram keeps track of application foreground status, informing the Task Navigator whenever the application can be hibernated. It also provides tools to get and set menu and toolbar resources common for all application windows.

Check the dependencies to all required libraries.

hildon-1 and **libosso** libraries need to be added to the list in PKG_CHECK_MODULES macro in configure.ac.

```
# part of configure.ac
PKG_CHECK_MODULES(MOA, gtk+-2.0 hildon-1 libosso)
# ...
```

Integrating to Application Framework

To have the application to appear in the menu and be controllable by Task Navigator, two files - desktop and service files - must be created.

Desktop file

Below is the contents of a minimal desktop.in file.

```
[Desktop Entry]
Encoding=UTF-8
Version=1.0
Type=Application
Name=MyOwnApplication
Exec=/usr/bin/moa
X-Osso-Service=org.maemo.moa
Icon=qgn_list_gene_default_app
```

The desktop file defines the the icon and caption text, executable name and path, and D-Bus service name (*X-Osso-Service*). Applications appearing in the Menu are started via the D-Bus activation mechanism, which use the service name to identify each application.

The desktop file must be copied to */usr/share/applications/hildon* directory. The name must be unique and reflect the name of the application, e.g. *moa.desktop* in this case.

Service file

In order to D-Bus to activate our application on demand and make sure that only one instance of the application is running, a service file is required.

Contents of org.maemo.moa.service.in file of the example application:

```
[D-BUS Service]
Name=org.maemo.moa
Exec=/usr/bin/moa
```

Service name defined in the service file must match with the one defined in the desktop file. The file must be placed in */usr/share/dbus-1/services*.

Registering to D-Bus

The application must be registered as a D-Bus service so that it will not be killed automatically by the system. This can be done with `osso_initialize()`. The first parameter needs to match the service name defined in the service file. Also remember to call `osso_deinitialize()` before exiting the application.

```
#include <stdlib.h>
#include <hildon/hildon-program.h>
#include <libosso.h>

#define PACKAGE_DBUS_NAME "org.maemo." PACKAGE_NAME

int main(int argc, char** argv) {

  HildonProgram* program;
  HildonWindow* window;

  osso_context_t* context = NULL;
```


(C) Copyright Nokia 2013. All rights reserved.

```
  /*Initialize GTK+*/
  gtk_init(&argc, &argv);

  program = HILDON_PROGRAM(hildon_program_get_instance());
  g_set_application_name("My Own Application");
  window = HILDON_WINDOW(hildon_window_new());
  hildon_program_add_window(program, HILDON_WINDOW(window));

  context = osso_initialize(PACKAGE_DBUS_NAME, PACKAGE_VERSION, TRUE, NULL);

  if(context == NULL) {
    g_print("LibOSSO init failed!");
    return EXIT_FAILURE;
  }

  ...

  gtk_main();

  osso_deinitialize(context);

  return EXIT_SUCCESS;
}
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.