Implementing user-selectable autostart feature in Qt for Symbian

Overview

This article describes how to enable a Qt application to be started during a device bootup process, with the ability to enable/disable the autostart feature dynamically.

Detailed description

This article extends Archived:Implementing user-selectable autostart feature using Symbian C++ specifically for Qt applications. In native Symbian code, it's possible to detect whether the application was started during bootup or from the menu by checking for the existence of additional data defined in the registration resource file. If `opaque_data` is missing, we know the application was auto-started as these registration files are omitted by the startup list manager.

For Qt applications however, similar implementation is difficult as it requires re-implementing methods from the S60 application framework. As an alternative, it is possible to create a small, native Symbian C++ executable that will be added to the startup list; this EXE will check if the main application has the autostart setting turned on and launches it if it does.

S60AutoStart

S60AutoStart is a piece of code that can be included in an existing Qt project to add the autostart capability in S60 devices.

Preconditions

- Install Qt SDK

Usage

- #Download and extract the ZIP into your Qt project folder (that contains your **.pro** file).
- Define the following variables in your **.pro** file:

- `TARGET.UID3`
  - Specifies a unique ID to use for the target Symbian application
  - See qmake variable reference for more information.

- `s60launcher.UID3`
  - Specifies a unique ID to use for the native launcher application
  - Same format as `TARGET.UID3`, value must be different.

- Add the following line to the end of your **.pro** file:

```
include(s60autostart/s60autostart.pri)
```

- Run **qmake** to recreate makefile, then build normally.

Notes:

After a successful build, the SIS file needs to be signed against a trusted certificate, as required by the S60 startup list manager. Autostart application cannot be installed using a self-signed SIS package.

Implementation details

(C) Copyright Nokia 2013. All rights reserved.

S60AutoStart uses a native Symbian executable, included in the main application SIS file and added to the startup list. This executable checks for the presence of a `c:\system\data\<AppName>.ini` file, and launches the main application if it exists.

It's easy to provide a setup UI for turning autostart on/off in the main application using the provided helper functions:

```
#ifdef Q_OS_SYMBIAN

#include "s60autostart.h"

bool autostart = S60AutoStart::IsAutoStartEnabled();

// toggle autostart
S60AutoStart::SetAutoStart( !autostart );

#endif // Q_OS_SYMBIAN
```

The launcher passes a command line parameter *autostart* to the application. This makes it possible to determine whether the application was launched automatically or by the user. Parse the command line arguments in `main()` for *autostart* string, or simply check if argc > 1.

Notes about backup & restore

S60AutoStart currently uses the `c:\system\data` folder to store the application startup settings file. This location allows both processes (main application and the native launcher) to access the file without any special platform security capabilities. However, a minor issue is that `c:\system\data` is not included in the backup process, so after running a backup and restore the application will lose the current startup setting - it defaults to `off` and must be turned on again in the main application.

If this is a problem, S60AutoStart can be modified to use some other (unprotected) folder instead, for example `C:\data`. Backup and restore will work correctly for files in this folder, but they will also be visible in the File Manager application.

Download

S60AutoStart

- File:Qt for symbian s60autostart.zip

Simple example Qt project that has a UI to turn autostart on/off

- File:Qt AutoStartApp.zip

See also

- Archived:Implementing user-selectable autostart feature using Symbian C++
- TSS000348 - Backup & Restore of public files in the C:\System folder is not supported

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.