



Contents

- [1 Startup List Management API](#)
- [2 Notes](#)
- [3 Useful links](#)
- [4 Example Code](#)

Startup List Management API

Steps for auto-starting an EXE on boot time in S60 3rd Edition.

1. Create an .RSS file in your project's directories (e.g. in the <PRJ_DIRECTORY>\DATA directory, where the other RSS files are located too)

For example: 06000001.rss

Here a number is used for the name, which is the same as the Package UID (pUID) of the project. This might help later in identifying the resource and is a good reminder when adding the entry in the PKG file. The name is however irrelevant at this point.

Add the following code to the new .rss file:

```
#include <startupitem.rh>

RESOURCE STARTUP_ITEM_INFO startexe
{
    executable_name = "c:\\sys\\bin\\StartEXE.exe";
    recovery = EStartupItemExPolicyNone;
}
```

In the code above:

- **c:** represents the drive where the program to be started is expected to have been installed
- If the drive is not known (will be selected at install time) use the wild card encoding for drives, the **!:**
- **StartEXE.exe** is the name of the application to be started and this will of course depend on your choice of name for your project. Must be the same as specified for the **TARGET** statement in the MMP file.

2. Open your .MMP file.

Add the following lines to include the new resource in the build.

```
START RESOURCE ..\DATA\06000001.rss
END
```

- The path specified above is relative to the MMP file location, usually in the <PRJ_DIRECTORY>\GROUP. The path must point to the new RSS file created so make sure that the string is in sync with the file location decided at step **1.** above.

How_to_autostart_an_application_on_boot_up_in_3rd-Startup_List_Management_API

3. Add the following entry in your .PKG file:

```
"C:\Symbian\9.1\S60_3rd_MR\Epoc32\data\06000001.rsc"- "c:\private\101f875a\import\[06000001].rsc"
```

- Application can be installed on any drive (either phone memory or memory card) but resource file [uid].rsc must be in **c:\private\101f875a\import**.
- **C:\Symbian\9.1\S60_3rd_MR** represent the install path (aka EPOCROOT) of the SDK you use. If the PKG file is used exclusively under Carbide.c++ you can replace this string with the macro \$(EPOCROOT)
- The *.rsc file is the result of compiling the *.rss file created at step one. It will not exist before re-building the project. Also, the extension might be different if your project's supported languages set does not include "SC", e.g. the extension could be *.r01. See Internationalization.
- The resource file must be always copied on the drive **c:**, in a special data cage, so the destination path is always the same: **c:\private\101f875a\import**
- The name of the file must be changed when copied on the device. It will always be the Package UID (pUID) (as specified in the header of the PKG file) but without the **0x** prefix and enclosed by square brackets ([and]).

4. That's it. Now just build re-build the project making sure that the new resource file is built and if everything else is ok with the project you will obtain a sis files ready to install on the phone. On the next phone reboot your application will be started automatically.

Notes

- If the EXE exits within a few (about 5?) seconds of starting up, a message is displayed, reading: *Unable to start <name of EXE>. Application may need to be removed.*
- There are several Known Issues and Technical Solutions about using the Startup List logged in the Forum Nokia Technical Library. Search for "Startup List" to get an up-to-date overview.
- **Startup List Management API does not work with Self-Signed application**. It should be signed with a trusted certificate (Open Signed Online or Open Signed Offline during R&D stage and Symbian Signed when released) even if otherwise the capabilities required for the project do not justify it.
- It is observed that on some phones (tested on E61) this startup list manager API works for self-signed apps with capability WriteUserData (and unprotected-range UID) too.
- The exe to be started must be installed directly from the root of the package that contains the resource file (not accepted from embedded file)

Useful links

- Read this whole thread which contains all the possible cases working with Strtup List Management API <http://discussion.forum.nokia.com/forum/showthread.php?t=138079>

Example Code

Startup List Management Example
