

Disabling autostart by default

Currently, on S60 3rd Edition, if an application wants to be autostarted, you can use the Startup list management API. Detailed information can be found in the S60 3rd Edition SDK.

In order to pass the Symbian Signed criteria, you need to provide a function in your application to disable the autostart or autoboot function. For the time being, there is no public API to address this problem. There are two workarounds:

1) Design a launcher application, which is used to launch a REAL application. The launcher application needs to be autostarted whenever the phone is booted. It checks the setting file (.ini). If the boot flag in the setting file is ON, then it launches the REAL application; if it is OFF, then it just terminates itself without launching the REAL application.

Please note that the launcher should not terminate itself after it successfully launches the REAL application. Otherwise, the REAL application cannot be successfully launched because the launcher application has terminated.

2) You can also make the REAL application autostart every time the phone boots up. In the early phase of the application, it checks the setting file. If the flag in the setting file is ON, it continues to run itself. If the flag is OFF, then it can call the `CAknAppUi::Exit();` function to terminate itself in the early phase of launching the application.

Compared to the first method, this method does not need to face the problem that two applications need to have the same secure ID. However, it may take more CPU time to load the application if the application is big and the user chooses not to boot the application every time.

There are also a couple of problems related to the second method:

If the application can be launched both automatically and manually from the menu, a mechanism for detecting the startup type is needed.

This can be done as follows:

When the application is launched from the startup list, all launch parameters specified in the application registration file are omitted.

Define 'opaque_data' parameter in registration file, APP_REGISTRATION_INFO resource:

```
#include <appinfo.rh>

#include <uikon.rh>

RESOURCE APP_REGISTRATION_INFO

{
    ...

    opaque_data = r_startup_type;
}

RESOURCE NUMBER_INT8 r_startup_type

{
    value = 1;
}
```

Then, override `ProcessCommandParametersL()` function in AppUI implementation:

```
TBool CMyAppUi::ProcessCommandParametersL( CApaCommandLine &aCommandLine )
```

```
{  
  
    if(aCommandLine.OpaqueData().Length() > 0)  
  
    {  
  
        // Opaque data exists, app. has been manually started from the menu  
  
    }  
  
    else  
  
    {  
  
        // App. has been auto-started -> exit if auto-start in settings is OFF  
  
    }  
  
    return CEikAppUi::ProcessCommandParametersL( aCommandLine );  
  
}
```

Another problem is that the startup service monitors all launched applications, and displays a warning message to the user if the application is terminated within 5 seconds after launching it. In order to avoid the unwarranted warning, the application must wait at least five seconds before attempting to exit. A timer can be set up in `ProcessCommandParametersL()` for this task. The application should wait in the background for the exit timer to complete. This can be done with

```
iEikonEnv->RootWin().SetOrdinalPosition(-1);
```

According to the Nokia test criteria, the second method is better when compared to the first one. However, its implementation can be rather tricky, as seen above.

Related Links

- [How to autostart an application on boot up in 3rd- Startup List Management API](#)
- [TSS000371 - Implementing user-selectable autostart feature](#)