

# Archived:Implementing user-selectable autostart feature using Symbian C++

 Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template `{{ReviewForRemovalFromArchive|user=~~~~|write your reason here}}`.

## Description

If an application needs to be launched automatically at device startup, the [Startup List Management API](#) can be used. In order to pass the Symbian Signed test criteria, an application needs to provide a possibility for the user to disable the autostart feature. For the time being, there is no public API to address this problem.

## Solution

There are two workarounds:

- Design a launcher application, which is used to launch the actual application. The launcher application needs to be autostarted whenever the phone is booted. It checks the settings file (.ini). If the boot flag in the settings file is ON, then it launches the actual application; if it is OFF, it just terminates itself without further action.  
Note that the location of the .ini file in the file system should be such that both processes can access it. For Qt applications, this approach is also easier as the second one requires re-implementing methods from the S60 application framework.
- Make the actual application autostart every time the phone boots up. In the early phase of the application, check the settings file. If the flag for autostart is ON, it continues to run itself. If the flag is OFF, then call the `CAknAppUi::Exit()` function to terminate the application.

Note that there are 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:

Define an `opaque_data` parameter in the registration file, in the `APP_REGISTRATION_INFO` resource. When the application is launched from the startup list, all launch parameters specified in the application registration file are omitted; this feature can be used to resolve the startup type.

```
#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 the `ProcessCommandParametersL()` function in AppUI implementation:

```
TBool CMyAppUi::ProcessCommandParametersL( CApaCommandLine &aCommandLine )
{
    if(aCommandLine.OpaqueData().Length() > 0)
    {
        // Opaque data exists, app. was manually started from the menu
    }
}
```

```
else
{
// App. was auto-started ->exit if auto-start flag 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. 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 than the first one. However, its implementation can be rather tricky, as seen above.

## See also

---

[How to conform with Symbian Signed criteria - UNI-11: Autostart at device boot-up](#)

[Implementing user-selectable autostart feature in Qt](#)