

Archived:Static library usage on Symbian

 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}}`.

Overview

Static library usage in Symbian OS

Description

Symbian build tools provide a possibility to create statically linked libraries. When linking against a static library, links to routines and variables are resolved at compile time, and code from the static library is merged with the target application to form a single executable. Therefore, static linking increases the size of the executable when compared to dynamic linking.

The use of static libraries should be seen as an option in cases where linking to a DLL is not required or desired, but modularization and low-level version control still needs to be accomplished. When static libraries are used initially in a project, it is later easy to convert the application to use static interface DLLs instead.

In the MMP project definition file, 'TARGETTYPE LIB' is used for building a static library. As with other target types (for example, APP and DLL), this will specify the UID1 for the target. No other UID definitions are needed for static libraries.

Because a static library is just a collection of objects, it does not require to be linked against any import libraries. Hence, in the MMP file of the static library, LIBRARY and SYSTEMLIBRARY statements should not be used. Instead, the user of the static library is responsible for linking against all the required import libraries.

Using global and writable static data (WSD) in static libraries is possible but not recommended as it increases memory consumption and has limited support in the emulator. Furthermore, WSD may cause problems if the library is later converted into a static interface DLL.