

Archived:Compress your Symbian C++ executables



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

On all devices based on Symbian OS v7.0s (S60 2nd Edition and Series 80 2nd Edition) it is possible to save a considerable amount of disk space by compressing an application's executable files. Executable files include GUI application files (.APP), dynamic link library files (.DLL), and server files (.EXE). This step typically cuts the executable file size by 30 to 40 percent.

From Symbian OS v9 onwards (S60 3rd Edition), compression of the executable files is enabled by default for both GCCE and ARMV5 targets.

Solution

To use compression on S60 2nd Edition, simply add the COMPRESSTARGET statement to the corresponding project definition file (.MMP) and recompile the project. The directive line can be anywhere within the project definition file.

On S60 3rd Edition and onwards, adding the COMPRESSTARGET statement to the .MMP file has no effect on output files as compression is enabled by default. Disabling compression is possible by using the NOCOMPRESSTARGET statement.

The start-up time of the compressed application can vary from the uncompressed version, although the difference is typically not significant. The benefits of using compression generally outweigh possible disadvantages; therefore, it is strongly recommended to use compression.

Also, executable file compression does not significantly change the Symbian OS installation (.SIS) file size, since by default the installation file is also compressed.

Note that NOCOMPRESSTARGET has no effect for GCCE targets on S60 3rd Edition SDK, Maintenance Release. As a workaround, the following lines

```
if (&main::CompressTarget) {
    &main::Output(' --uncompressed ');
}
```

can be added to the `epoc32\tools\c1_bpapi.pm` file (line 977). This issue has been fixed in S60 3rd Edition, Feature Pack 1 and later SDKs.

