

## Reducing the size of your GCCE executables

You may find that executables compiled with GCCE for S60 3rd Edition are significantly larger when compiled with a GCCE and an S60 3rd Edition SDK than the corresponding S60 2nd Edition version, even though S60 3rd Edition binaries are compressed by default.

According to the [release notes for the S60 3rd Edition SDK Maintenance Release](#) ~~the recommended~~ possible optimizations for reducing the size of a 3rd Edition executable are `-O2 -fno-unit-at-a-time`. However, this is not the default after installation of the SDK. The easiest way to enable these for a particular project is to add the following line to the [MMP file](#):

```
OPTION GCCE -O2 -fno-unit-at-a-time
```

This is probably no longer necessary if you are using the S60 3rd Edition FP1 SDK, because it already includes the line

```
REL_OPTIMISATION=-O2 -fno-unit-at-a-time
```

in the `gcce.mk` file, making the above settings the default.

### **"Danger, Will Robinson!"**

This optimization is NOT recommended by Nokia but merely recognized as an option. Read the release notes for information about the side-effects of using this optimization and then decided whether to use it or not.

## Why this is needed

It seems that there is a single culprit that is alone responsible for the majority of the size increase: the `AknsConstants.h` file declares a host of "static const" structures for skin-related IDs that make it into the final binary, even though they are never referenced. These constants are included once for each `.cpp` file that uses "popular" header files such as `<aknutils.h>` or `<aknmessagequerydialog.h>`, leading to about 20k of excess data per file.

With the optimization options set as above, GCCE recognizes these as being "dead data", and removes them from the final executable.