

# Device Specific Instructions for your Application's Linker Script File

## IAR - STM32L496x

First thing. Copy the folder "SBLp" from the example application's project folder supplied with the bootloader into your application project folder. You will need to include the .s, .c and .h files it contains into your application build. Add this folder to your tool's include paths for access to the .h file.

If you are not creating and using your own memory regions and sections via the linker script and you have not made edits to your existing project linker script you may use the script file (\*.icf) found in the folder (SBLp) in place of the one created by the IAR tool. **This is required.** To do this use the top tool strip of IAR and go to **Project>Options**. In the Category pane (left most pane) select **Linker**. In the right pane select the **"Config"** tab. Check the box labeled **"Override default"** then fill in the path text box with "\$PROJ\_DIR\$/../SBLp/SBL\_App\_STM32L496x.icf". Click the OK button. Now the application build will use the linker file supplied by Driven 2 Design and not the tool generated linker script. This **MUST** be done to locate the application above the bootloader in the MCU's

While in the settings also set the tool to output a raw binary image. In the left pane under Category select **"Output Converter"**. In the right pane select the **"Output"** tab. Check the box labeled **"Generate additional output"**. From the drop down box labeled **"Output format"** select **"Raw binary"**. This causes the tool to output a raw binary image of the build as well as the elf file. It is the raw binary image (.bin file) that Flash File Guardian uses to create the secure (.ffg) file.

That's it! Continue to develop your application as normal. It will compile and link as before but located to a higher address (above the bootloader) and include a blank ID Data Block in FLASH memory.

If you are adding the bootloader to an existing project with a linker script that you have made edits to then you may choose 1 of the following 2 options. Edit your existing linker script to include the required memory regions and sections or edit the Driven to Design provided linker script to include your additions. It's up to you.

Here are the required memory regions and code section edits.

```
/* Memory Region Addresses */
define symbol __ICFEDIT_region_RAM_start__      = 0x20000000;
define symbol __ICFEDIT_region_RAM_end__        = 0x2000BFFF;
define symbol __ICFEDIT_region_SRAM2_start__    = 0x10000000;
define symbol __ICFEDIT_region_SRAM2_end__      = 0x10003FFF;

/* Replaced ROM memory Region Addresses */
/* The new start address places your application above the bootloader residing at 0x08000000 */
define symbol __SBLp_region_ROM_start__         = 0x08006800;
define symbol __SBLp_region_ROM_end__           = 0x0803FFFF;

/* place the interrupt vector table and the ID Data Block sections */
place at address mem: __int_vec_table__ { readonly section .intvec };
place at address mem: __id_data_block__ { readonly section .id_data };

/* now place the code and the bss sections */
place in ROM_region { readonly };
place in RAM_region { readwrite, block CSTACK, block HEAP };
```

The rest of the file should be left as it was. Now your application may be built, secured with Flash File Guardian and uploaded to your board via MCU Flasher.