

## IAR EWARM 链接未使用的段

IAR ILink 在执行链接时，如果静态库或目标文件中的某些函数及变量从没有被引用，最终这些函数或者变量会被丢弃，不会被链接到可执行文件中。这样做可以减小生成的二进制文件的尺寸。但有的时候，用户出于某些原因想要保留未使用的函数或者变量。如果函数的定义是在 C/C++源码文件，可以使用“`__root`”关键字保留的函数或者变量。

例如，要保留程序中未调用的 `GenerateRandomNumber()` 函数。

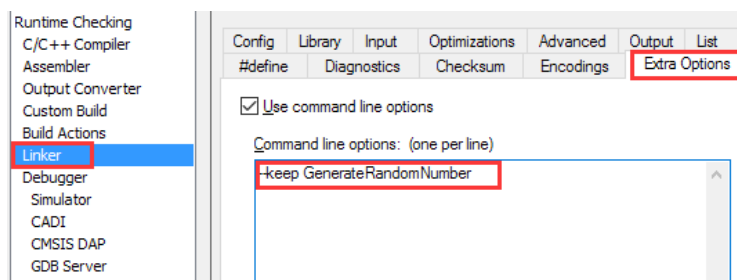
使用“`__root`”关键字：

```
__root void GenerateRandomNumber (uint16_t *ptr, uint16_t length, uint16_t factor)
```

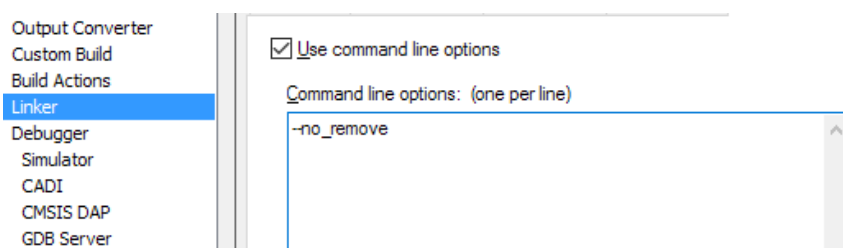
但如果要保留的是静态库中的函数，就没法用“`__root`”关键字了。这时可以使用 Linker 选项“`--keep symbol`”指令强制链接程序中未使用的符号。

`--keep` 的作用范围是整个工程，不论需要保留的符号是在源码文件中还是静态库中都适用。

在 Linker 选项中使用 `--keep` 加上函数名，保留 `GenerateRandomNumber()` 函数：



上面的方法适用在需要保留的符号较少的情况，假如有成百上千的符号要强制保留，使用前面的方法，似乎不太能接受。这时可以使用命令行 `--no_remove` 或 `--no_fragments`。



`--no_remove` 和 `--no_fragments` 有效范围是整个工程，工程中所有源文件或者静态库中的符号都将会保留。

此外，如果用户只想保留某几个库中的某几个段，其他的段不需要进行链接，以达到保留所需要的段的同时尽量减少最终的二进制文件的尺寸。要保留任意库或目标文件中的任意段，可以在 ILink 配置文件中使用 `keep` 命令。

icf 文件 `keep` 命令格式如下：

```
keep { section-selectors }[ except { section-selectors}];
```

section-selectors 是 section 的选择格式:

[section-attribute ] [ section section-name ][object {module|filename}]

section-attribute:

ro [ code|data ] | rw [ code|data ] | zi

只有指定了下列属性的 section 才会被选择:

ro|readonly, for read-only sections.

rw|readwrite, for read/write sections.

zi|zeroinit, for zero-initialized sections.

section section-name:

section 的名称, 只有匹配名称 section 才会选中, 有 2 个通配符可以使用: “?” 和 “\*”。

?: 匹配任何一个字符

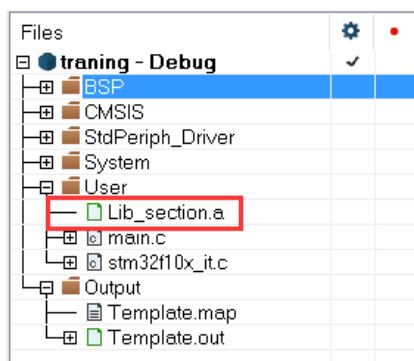
\*: 匹配 0 或者多个字符

object {module|filename}:

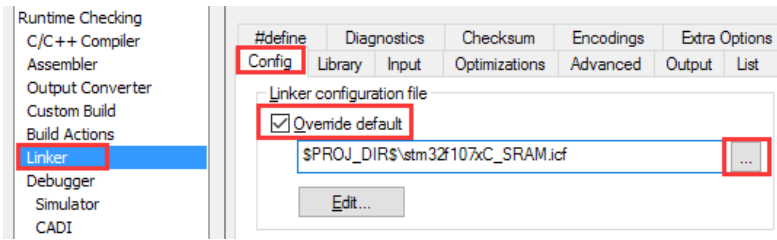
匹配静态库中的模块或者目标文件中的 section 才会选中。

except{}是可选项, 用于排除要不需保留的段。

示例: 保留 Lib\_section.a 库中除了.function2 以外属性为只读的段。



1. 从 IAR EWARM 的安装目录下将对应的链接器配置文件复制到工程目录, 避免修改原始的 icf 配置文件。然后进入工程的 Options 选项, 选择 Linker -> config, 勾选 “Override default”, 单击旁边的按钮, 选择之前复制到工程目录的 icf 文件。



2.用文本编辑器打开 icf 文件，可以在任何位置加上 keep 指令：

```

36 place at address mem: __ICFEDIT_intvec_start__ { readonly section .intvec };
37
38 place in ROM1_region { section .rodata,
39                       section .myconst }; /*constants*/
40 place in ROM2_region { section .text, section .mycode,
41                       section .data_init,
42                       section .iar.init_table,
43                       block FUNCBLOCK};
44
45 place in RAM1_region { section .mydata};
46 place in RAM2_region { readwrite,
47                       block CSTACK,
48                       block HEAP };
49
50 keep {readonly object Lib_section.a} except {section .function2 };
51

```

此后，构建程序时 Lib\_section.a 中的只读段，除了.function2 以外，不论程序中是否引用，都会保留。

版本	描述	作者	日期
1.0	创建	BMRTECH 付元斌	2017-12-15