

Dummy-Header File & Language Extensions

Use the dummy-header-file or the Language extensions to handle non-ANSI C keywords:

Language extensions can be supported by means of either the dummy-header file or the language extensions card of the Options menu:

Let us start with an example:

```
__callt TBool write_DK(int *pTab, void *buf, TBool Check)
{
...
}
```

a) Since ANSI C does not recognize `__callt` and `TBool`, you can add a declaration or `#define` for `__callt` and `TBool` in the Project Dummy Header file. Crystal REVS automatically `#includes` the Project Dummy Header file at the beginning of each file in the project.

You can open the Project Dummy Header file from the "Project" pull-down menu or the "Prj Files" pane of the Browse Window.

The Project Dummy Header file is like any other header file, you can edit it and save it. You can `#include` other files in it.

```
#ifndef dummy_header_is_included_once
#define dummy_header_is_included_once 1

# define __callt

typedef int TBool;

#endif
```

After you modify the dummy-header file, save it and Reparse All files.

b) Or you can use the Language Extensions card of the Options Menu. You can add a new scheme. Enter `TBool` in the "Types" and click the <Add> button. Similarly add `__callt` to the "Others" column.

(Adding to the "Others" column has the same effect as `#define __callt` as shown above.)

Both Project Dummy Header file and Options settings are saved on a per-project-basis.