



## Adding A New GCT Build Target

- [Adding A New Build Target](#)
- [Adding A Typical New Non-RESOURCE Build Target](#)
- [Adding A Typical New RESOURCE Build Target](#)

### Adding A New Build Target

These instructions assume that the standard GCT package has previously been installed and operates correctly for the standard build target. See [Installing GCT](#) for information on installing GCT.

This is a summary of the tasks required to add a new build target to GCT:

- "DEFAULT.ODB" - Add new build target to list of available build targets
- "DEFAULT.ODB" - Add target specific compiler options (if required)
- "TARGET.ODB" - Add new build target compiler options
- "INCLUDE" directory - create new build target "INCLUDE" directory
- "LIB" directory - create build target "LIB" directory

Note that the target names must be DOS filename compatible (eight or less characters). This is to ensure that 16-bit DOS compilers are able to access the target directories. Non-DOS filename compatible target names result in filename mangling.

#### "DEFAULT.ODB"

The "DEFAULT.ODB" file is located in the "OPTION" subdirectory under your GCT root directory.

Add your unique target name to the "TargetsSupported" list.

You may wish to add specific target compiler options here by using the syntax  
"[target name]:CompilerOptions:=[your options]"

#### "TARGET.ODB"

The "TARGET.ODB" file is located in the "OPTION" subdirectory under your GCT root directory.

The "TARGET.ODB" lists all the compiler specific options used to build for your specific target.

The syntax is as follows:

- [target name]:ObjectExtension:=[Compiled Object filename extension without the preceding "."]
- [target name]:CompilerName:=[The name of the compiler. Current supported compiler names are: "Gnu", "Sds", "Ppc", "Microsoft", and "Borland". Defaults to Microsoft]
- [target name]:CCompiler:=[Command for invoking C Compiler]
- [target name]:CplusplusCompiler:=[Command for invoking C++ Compiler if different from the C Compiler]
- [target name]:COptionsFileCommand:=[If set, specifies the compiler options/response file invoking command, eg "-f", otherwise invokes compiler with full command-line options. For compilers requiring a space between the command and the filename, underscores are translated into space, eg "-f\_" would invoke "-f compile.cfg"]
- [target name]:CFlags:=[C Compiler flags for compiling standard ST blocks]
- [target name]:NonSTBodyCFlags:=[C Compiler flags for compiling blocks whose body is specified in C using source.c files and "NON\_ST\_BODY" command in the ST. If not specified, uses CFLAGS]
- [target name]:NonSTBodyCplusplusFlags:=[C++ Compiler flags for compiling blocks whose body is specified in C++ using source.cxx/source.cpp/source.C files and "NON\_ST\_BODY" command in the ST. If not specified, uses NonSTBodyCFLAGS]
- [target name]:IncludeSTFlags:=[Includes ST compiler defines from default.odb for the C compilation]
- [target name]:FilesToCompile:=[Optional list of files to compile. Note that the list of files to compile MUST have unique first letters, as the first letter is used to generate a unique filename for each function block. By

default will compile exec.c, template.c, instance.c and resource.c (resources only). Otherwise only compiles files explicitly listed]

- [target name]:CompileFilesSeparately:=[If set to "YES", then compiles files separately, otherwise will invoke compiler once for each block with a list of files]
- [target name]:RenameObjects:=[Explicitly name object when compiling (only valid if CompileFilesSeparately is "YES") using the specified command, eg "-o". For compilers requiring a space between the command and the filename, underscores are translated into spaces, eg "-o \" would invoke "-o filename.ObjectExtension"]
- [target name]:AddIncludeSeparator:=[if set "YES", adds additional directory separator after include path. Default is no]
- [target name]:Librarian:=[Command for invoking Librarian. If none provided, will not generate a library file from the compiled objects]
- [target name]:LibrarianOptionsCommand:=[If set, specifies the librarian options/response file invoking command, eg "@", otherwise invokes librarian with full command-line options. For librarians requiring a space between the command and the filename, underscores are translated into space, eg "@ \" would invoke "@ libopts.cfg"]
- [target name]:Linker:=[Command for invoking Linker. If none provided, will not attempt to link objects]
- [target name]:LinkerOptionsCommand:=[Specifies the linker options/response file invoking command, eg "@". For linker requiring a space between the command and the filename, underscores are translated into space, eg "@ \" would invoke "@ linkopts.lnk"]
- [target name]:LinkerFlags:=[Linker flags for generating resource]
- [target name]:Libraries:=[List of standard libraries to link into resource]
- [target name]:DLLLibraryImporter:=[Command for invoking library importing utility. Note that this is for Windows only, where the resource is a DLL and requires importing the symbols into a library. This option must be specified for Microsoft and Borland Windows compilers]
- [target name]:Function.PostBuildCommand :=[List of commands (line separated) to be executed following the build of a function]
- [target name]:Function\_Block.PostBuildCommand :=[List of commands (line separated) to be executed following the build of a function block]
- [target name]:Program.PostBuildCommand :=[List of commands (line separated) to be executed following the build of a program]
- [target name]:Resource.PostBuildCommand :=[List of commands (line separated) to be executed following the build of a resource]
- [target name]:PostBuildCommand :=[List of commands (line separated) to be executed following the build of all functions, function blocks, programs, and resources]
- [target name]:Cleanup:=[If "YES/yes/TRUE/true", will delete all temporary build files after each build. Otherwise does not delete compiler and linker options/response files or exec.c files. Default is false]

Note that because each compiler has its unique set of commands, each compiler has to be handled differently, making it difficult to produce a common setup. If the required combination of commands cannot be achieved with the currently available list of options, additional target specific support may need to be added the the GCT build system. If this is the case, contact your local GCT support representative.

### "INCLUDE" Directory

You will need to create an include directory for your new target under "[target name]\include" containing all the target specific and common include files.

See [Adding A Typical New Non-RESOURCE Build Target](#) and [Adding A Typical New RESOURCE Build Target](#) for information about what files are typically required.

### "LIB" Directory

You will need to create a lib directory for your new target under "[target name]\lib" containing all the target specific library files (only for RESOURCE build targets).

See [Adding A Typical New Non-RESOURCE Build Target](#) and [Adding A Typical New RESOURCE Build Target](#) for information about what files are typically required.

## Adding A Typical New Non-RESOURCE Build Target

These instructions assume that the standard GCT package has previously been installed and operates correctly for the standard build target.

A Non-RESOURCE build target does not require the compiling of the "INSTANCE.C", "TEMPLATE.C" or "RESOURCE.C" files, or the linking of the RESOURCE block. Consequently the RESOURCE specific header files

are not required.

This is a summary of the tasks required to add a new build target to GCT:

- "DEFAULT.ODB" - Add new build target to list of available build targets
- "DEFAULT.ODB" - Add target specific compiler options (if required)
- "TARGET.ODB" - Add new build target compiler options
- "INCLUDE" directory - create new build target "INCLUDE" directory

Note that the target names must be DOS filename compatible (8.3). This is to ensure that 16-bit DOS compilers are able to access the target directories. Non-DOS filename compatible target names result in filename mangling.

### "DEFAULT.ODB"

The "DEFAULT.ODB" file is located in the "OPTION" subdirectory under your GCT root directory.

Add your unique target name to the "TargetsSupported" list.

You may wish to add specific target compiler options here by using the syntax

"[target name]:CompilerOptions=[your options]"

### "TARGET.ODB"

The "TARGET.ODB" file is located in the "OPTION" subdirectory under your GCT root directory.

The "TARGET.ODB" lists all the compiler specific options used to build for your specific target.

Below is a list of typical options required. Options not required do not need to be specified.

- [target name]:ObjectExtension:=o
- [target name]:CompilerName:=
- [target name]:CCompiler:=ccompiler.exe
- [target name]:CCOptionsFileCommand:=-f\_
- [target name]:CFlags:=-flags -moreflags
- [target name]:NonSTBodyCFlags:=-cflags -morecflags
- [target name]:NonSTBodyCPPFlags:=-cppflags -morecppflags
- [target name]:IncludeSTFlags:=no
- [target name]:FilesToCompile:=exec
- [target name]:CompileFilesSeparately:=yes
- [target name]:RenameObjects:=-o\_
- [target name]:AddIncludeSeparator:=yes
- [target name]:Cleanup:=no

Note that this typical list will vary from target to target.

### "INCLUDE" Directory

You will need to create an include directory for your new target under

"[target name]\include" containing all the target specific include files.

Below is a minimum list of files required, which can be copied from the standard Windows target include directory.

- "MACHINE.H" - Defines specific configuration of the target
- "UI.H" - User Includes - this includes all necessary header files required

In addition, the above "UI.H" should include the following header files to obtain standard definitions required by the files generated during the GCT building process.

- "STDYPES.H" - Eurotherm Controls Standard Data Types
- "DEF\_DEFS.H" - Basic CDL/IEC/RESOURCE Data Types
- "DEF\_FBTP.H" - Type Definitions for ST and C
- "DEF\_VALS.H" - Value structure data type

Individual targets may wish to provide their own implementation of the following files - alternatively, they can use the standard implementation by including the following files:

- "FUN\_FUNC.H" - IEC Functions prototypes/macro definitions

## "LIB" Directory

No "LIB" directory is required for Non-RESOURCE build targets.

## Adding A Typical New RESOURCE Build Target

These instructions assume that the standard GCT package has previously been installed and operates correctly for the standard build target.

This is a summary of the tasks required to add a new build target to GCT:

- "DEFAULT.ODB" - Add new build target to list of available build targets
- "DEFAULT.ODB" - Add target specific compiler options (if required)
- "TARGET.ODB" - Add new build target compiler options
- "INCLUDE" directory - create new build target "INCLUDE" directory
- "LIB" directory - create build target "LIB" directory

Note that the target names must be DOS filename compatible (8.3). This is to ensure that 16-bit DOS compilers are able to access the target directories. Non-DOS filename compatible target names result in filename mangling.

## "DEFAULT.ODB"

The "DEFAULT.ODB" file is located in the "OPTION" subdirectory under your GCT root directory.

Add your unique target name to the "TargetsSupported" list.

You may wish to add specific target compiler options here by using the syntax "[target name]:CompilerOptions:=[your options]"

## "TARGET.ODB"

The "TARGET.ODB" file is located in the "OPTION" subdirectory under your GCT root directory.

Below is a list of typical options required. Options not required do not need to be specified.

- [target name]:ObjectExtension:=o
- [target name]:CompilerName:=Gnu
- [target name]:CCompiler:=gcc.exe
- [target name]:COptionsFileCommand:=@
- [target name]:CFlags:=-Uunix -mc68000 -Dm68k -Dcpu32 -O -Dxec\_cif -c
- [target name]:NonSTBodyCFlags:=-Uunix -mc68000 -Dm68k -Dcpu32 -O -Dxec\_cif -c
- [target name]:Librarian:=ar.exe
- [target name]:Linker:=ld.exe
- [target name]:LinkerOptionsCommand:=@
- [target name]:LinkerFlags:=-e TheResourceC -T link68k.map
- [target name]:Libraries:=libgcc.a gnu\_lib.o
- [target name]:Resource.PostBuildCommand :=objcopy -Osres resource.out resource.m00
- [target name]:Cleanup:=yes

Note that this typical list for using the GNU compiler will vary from target to target.

## "INCLUDE" Directory

You will need to create an include directory for your new target under "[target name]\include" containing all the target specific include files.

It is recommended that all files listed in [GCT Common Include Files](#) should be copied unmodified to the target include directory.

Below is a minimum list of files required, which can be copied from the standard Windows target include directory.

- "MACHINE.H" - Defines specific configuration of the target
- "UL.H" - User Includes - this includes all necessary header files required
- "INSTANCE.H" - RESOURCE Instance Data header - Common File
- "TEMPLATE.H" - RESOURCE Template Data header - Common File

In addition, the above "UL.H" should include the following header files to obtain standard definitions required by the files generated during the GCT building process.

- "STDYPES.H" - Eurotherm Controls Standard Data Types
- "DEF\_DEFS.H" - Basic CDL/IEC/RESOURCE Data Types
- "DEF\_FBTP.H" - Type Definitions for ST and C
- "DEF\_VALS.H" - Value structure data type
- "SEH\_UW.H" - Table defs for creating downloadable code for the sequence prog
- "STC\_DEFS.H" - Prototypes for ST compiler plants
- "CIF\_VREF.H" - VarRef 'C' functions for ST compiler
- "CIF\_RSRC.H" - The header file for the 'C' interface to the RESOURCE
- "CIF\_TFL.H" - TASK list manager
- "RUT\_TIME.H" - RESOURCE time utilities

Individual targets may wish to provide their own implementation of the following files - alternatively, they can use the standard implementation by including the following files:

- "FUN\_FUNC.H" - IEC Functions prototypes/macro definitions

---

## "LIB" Directory

You will need to create a lib directory for your new target under "[target name]lib" containing all the target specific library files.

You will need to contact the GCT project team or Mike Fox to determine the library files required.

---



Last Modified: Monday, 25-Nov-96 15:57:04 GMT

---

*Site Maintained By Adrian Oliver*

---