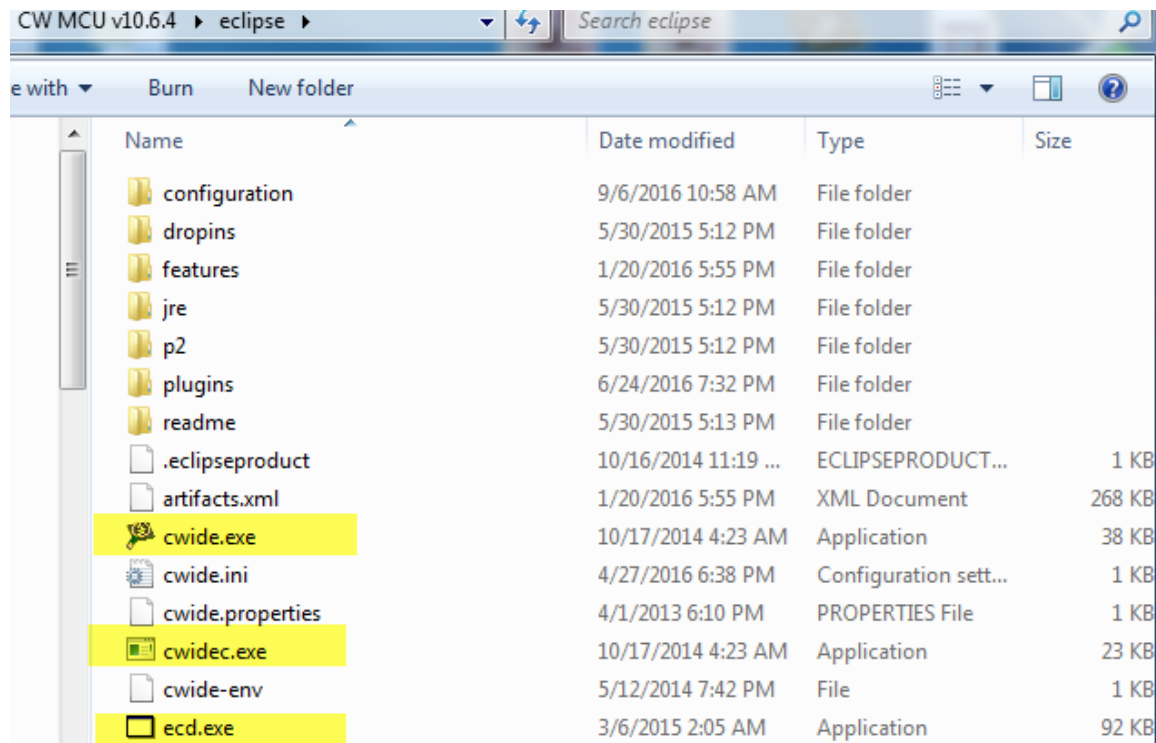


# CodeWarrior 10 Command Line Interface – usage and examples

By Jennie Zhang

Inside {CW10 install folder}\eclipse, there are three executables command line version: **ecd.exe**, **cwide.exe**, **cwidedec.exe**



1. **ecd.exe**: this is used to run build projects from the command line.
2. **cwide.exe and cwidedec.exe**: The cwide.exe and cwidedec.exe can be started to execute a debugger shell script (see article <https://mcuoneclipse.com/2012/08/03/codewarrior-flash-programming-from-a-dos-shell/>)

This article will focus on **ecd.exe** as discussion.

## ECD command

The ecd commands are listed below: the first 3 commands are more widely used, so I will focus on them in this article with examples.

**build**

**generateMakefiles**

**references**

**getOptions**

**setOptions**

## updateWorkspace

# 1. Build

### Syntax

```
ecd.exe -build [-verbose] [-cleanAll] [ -project path [ - config name | -allConfigs] -cleanBuild]
```

### Parameters

-cleanBuild

The -cleanBuild command applies to the preceding -project only.

-cleanAll

The -cleanAll command applies to all -project flags.

-config

The build configuration name. If the -config flag isn't specified, the default build configuration is used.

### Examples:

- Create two project "project1", "project2" in CW10 IDE in workspace D:\workspace
- Ecd.exe execution examples.

#### 1.1 Example 1:

Build the project in commandline with below command. it's necessary to use -data thus ecd can know the current workspace. If the -config flag isn't specified, the default build configuration is used.

Command:

***ecd.exe build -data d:\workspace -project D:\workspace\project1***

```
C:\Freescall\CW MCU v10.6.4\eclipse>ecd.exe build -data d:\workspace -project D:\workspace\project1
building project project1

**** Build of configura ct1 ****

"C:\Freescall\CW MCU v10.6.4\gnu\bin\mingw32-make" -j8 all
mingw32-make: Nothing to be done for `all`.
Build succeeded for project project1
```

#### 1.2 Example 2:

use -config to define the build configuration. Use -cleanBuild to clean build, it removes any previously-built binaries, obj,etc.

Command:

***ecd.exe build -data d:\workspace -project D:\workspace\project1 -config RAM -cleanBuild***

```

Administrator: Command Prompt
C:\Freescale\CW MCU v10.6.4\eclipse>ecd.exe build -data d:\workspace -project D:\workspace\project1 -config RAM -cleanBuild
Building project project1 configuration RAM
**** define build configuration ****
**** clean build preceding -project ****
**** Build of configuration RAM for project project1 ****
"C:\Freescale\CW MCU v10.6.4\gnu\bin\mingw32-make" -j8 all
'Building file: ../Sources/main.c'
'Building file: ../Project_Settings/Startup_Code/__arm_end.c'
'Building file: ../Project_Settings/Startup_Code/__arm_start.c'

```

### 1.3 Example 3:

Clean build and build two project with defined configurations

***ecd.exe build -data d:\workspace -project D:\workspace\project1 -config FLASH -project D:\workspace\project2 -config FLASH -cleanBuild***

```

Administrator: Command Prompt
C:\Freescale\CW MCU v10.6.4\eclipse>ecd.exe build -data d:\workspace -project D:\workspace\project1 -config FLASH -project D:\workspace\project2 -config FLASH -cleanBuild
!AutoBuildApp.Wa: gUnknownArgument!
**** build project1 with FLASH configuration ****
**** build project2 with FLASH configuration ****
**** Clean-only build of configuration FLASH for project project1 ****
Build complete for project project1
**** Build of configuration FLASH for project project1 ****
"C:\Freescale\CW MCU v10.6.4\gnu\bin\mingw32-make" -j8 all
'Building file: ../Sources/main.c'

```

## 2. generateMakefiles

Syntax

```
ecd.exe -generateMakefiles [-verbose] [ -project path [ - config name ] [-allConfigs] ] [-data workspace-path]
```

Parameters

**-config**

The build configuration name. If the `-config` flag isn't specified, the default build configuration is used.

**-data workspace-path**

The `-data workspace-path` flag can be used to specify a custom workspace.

Examples:

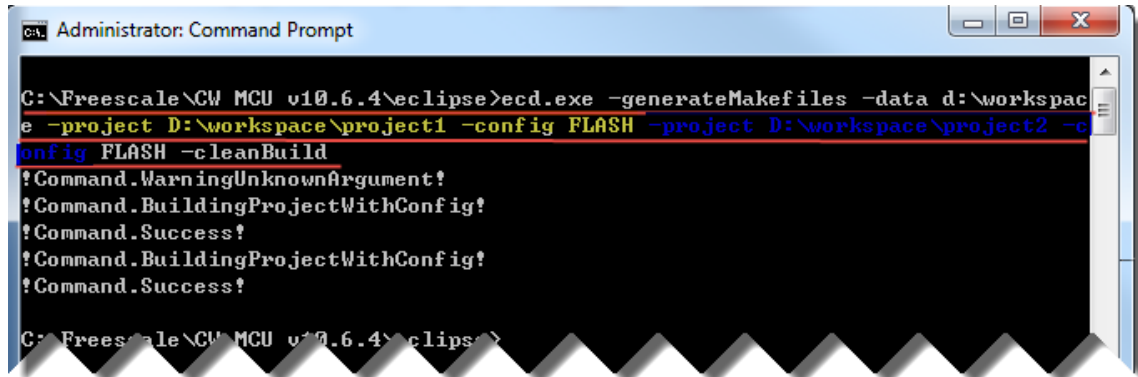
- Create two project "project1", "project2" in CW10 IDE in workspace D:\workspace
- Ecd.exe execution examples.

## 2.1 Example 1.

Generate makefiles for each of the two projects.

Command:

```
ecd.exe -generateMakefiles -data d:\workspace -project D:\workspace\project1 -config FLASH -project D:\workspace\project2 -config FLASH -cleanBuild
```



```
Administrator: Command Prompt
C:\Freescall\CW MCU v10.6.4\ eclipse>ecd.exe -generateMakefiles -data d:\workspace -project D:\workspace\project1 -config FLASH -project D:\workspace\project2 -config FLASH -cleanBuild
!Command.WarningUnknownArgument!
!Command.BuildingProjectWithConfig!
!Command.Success!
!Command.BuildingProjectWithConfig!
!Command.Success!
C:\Freescall\CW MCU v10.6.4\ eclipse>
```

After executing the command, two makefile related files will be generated under *D:\workspace\project1\FLASH* folder and *D:\workspace\project2\FLASH* separately for both of the projects.

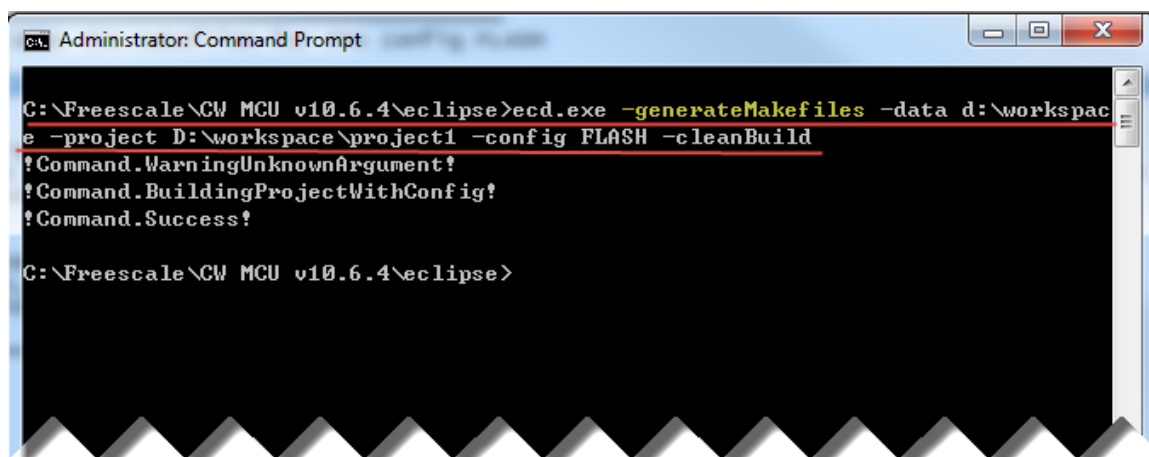
## 2.2 Example 2.

Generates the makefiles for the config FLASH. Then use *make.exe + makefile* to build a project

Step1: Generates the makefiles for the config FLASH

Command:

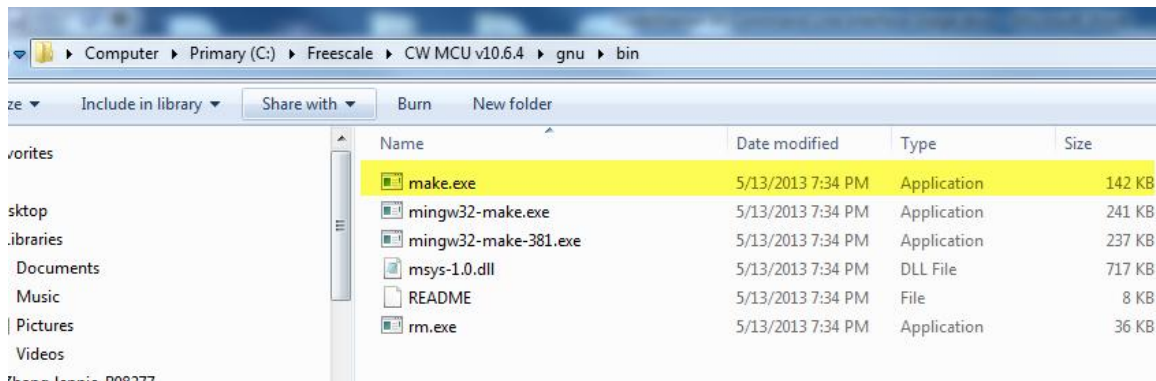
```
ecd.exe -generateMakefiles -data d:\workspace -project D:\workspace\project1 -config FLASH -cleanBuild
```



```
Administrator: Command Prompt
C:\Freescall\CW MCU v10.6.4\ eclipse>ecd.exe -generateMakefiles -data d:\workspace -project D:\workspace\project1 -config FLASH -cleanBuild
!Command.WarningUnknownArgument!
!Command.BuildingProjectWithConfig!
!Command.Success!
C:\Freescall\CW MCU v10.6.4\ eclipse>
```

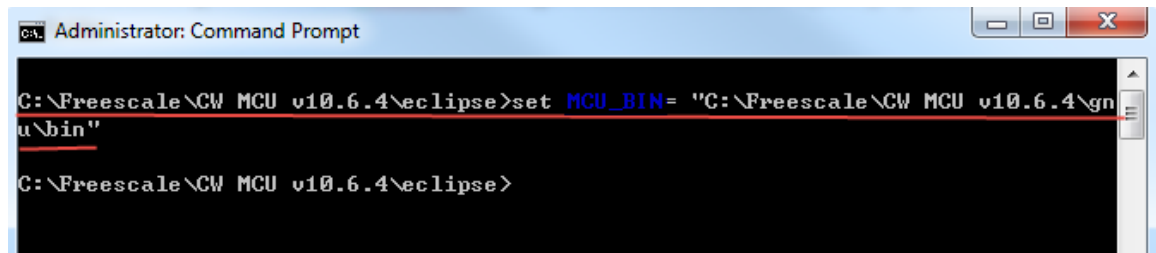
After executing the command, makefile will be generated under D:\workspace\project1\FLASH folder.

Step 2. Gnu make.exe is under C:\Freescale\CW MCU v10.6.4\gnu\bin. Set environment variable for *make.exe*



Command:

```
set MCU_BIN="C:\Freescale\CW MCU v10.6.4\gnu\bin"
```



Step 3. Go to makefile located folder then run make.exe to build the project

```
Administrator: Command Prompt
D:\workspace\project1\FLASH>%MCU_BIN%\make
'Building file: ../Sources/main.c'
'Executing target #1 ../Sources/main.c'
'Invoking: ARM Ltd Windows GCC C Compiler'
"C:/Freescale/CW MCU v10.6.4/Cross_Tools/arm-none-eabi-gcc-4_7_3/bin/arm-none-eabi-gcc" "../Sources/main.c" @"Sources/main.args" -MMD -MP -MF"Sources/main.d" -o "Sources/main.o"
'Finished building: ../Sources/main.c'
', '
'Building file: ../Project_Settings/Startup_Code/__arm_end.c'
'Executing target #2 ../Project_Settings/Startup_Code/__arm_end.c'
'Invoking: ARM Ltd Windows GCC C Compiler'
"C:/Freescale/CW MCU v10.6.4/Cross_Tools/arm-none-eabi-gcc-4_7_3/bin/arm-none-eabi-gcc" "../Project_Settings/Startup_Code/__arm_end.c" @"Project_Settings/Startup_Code/__arm_end.args" -MMD -MP -MF"Project_Settings/Startup_Code/__arm_end.d" -o "Project_Settings/Startup_Code/__arm_end.o"
'Finished building: ../Project_Settings/Startup_Code/__arm_end.c'
', '
'Building file: ../Project_Settings/Startup_Code/__arm_start.c'
'Executing target #3 ../Project_Settings/Startup_Code/__arm_start.c'
'Invoking: ARM Ltd Windows GCC C Compiler'
"C:/Freescale/CW MCU v10.6.4/Cross_Tools/arm-none-eabi-gcc-4_7_3/bin/arm-none-eabi-gcc" "../Project_Settings/Startup_Code/__arm_start.c" @"Project_Settings/Startup_Code/__arm_start.args" -MMD -MP -MF"Project_Settings/Startup_Code/__arm_start.d" -o "Project_Settings/Startup_Code/__arm_start.o"
'Finished building: ../Project_Settings/Startup_Code/__arm_start.c'
', '
'Building file: ../Project_Settings/Startup_Code/kinetis_sysinit.c'
'Executing target #4 ../Project_Settings/Startup_Code/kinetis_sysinit.c'
'Invoking: ARM Ltd Windows GCC C Compiler'
"C:/Freescale/CW MCU v10.6.4/Cross_Tools/arm-none-eabi-gcc-4_7_3/bin/arm-none-eabi-gcc" "../Project_Settings/Startup_Code/kinetis_sysinit.c" @"Project_Settings/Startup_Code/kinetis_sysinit.args" -MMD -MP -MF"Project_Settings/Startup_Code/kinetis_sysinit.d" -o "Project_Settings/Startup_Code/kinetis_sysinit.o"
'Finished building: ../Project_Settings/Startup_Code/kinetis_sysinit.c'
', '
'Building target: project1.elf'
'Executing target #5 project1.elf'
'Invoking: ARM Ltd Windows GCC C Linker'
"C:/Freescale/CW MCU v10.6.4/Cross_Tools/arm-none-eabi-gcc-4_7_3/bin/arm-none-eabi-gcc" @"project1.args" -o"project1.elf"
'Finished building target: project1.elf'
', '
D:\workspace\project1\FLASH>
```

After executing the command, project1.elf and project1.map will be generated under D:\workspace\project1\FLASH folder.

### 3. References

#### Syntax

```
ecd.exe -references -project path [-config name | -allConfigs] ( - list | -add | -remove)
referencedProjectLocation [buildConfigurationName]
```

## Parameters

```
-config name
```

The name of the build configuration to edit or list referenced project. If the -config flag is omitted, the active build configuration will be used.

```
-allConfigs
```

Specifies that all build configurations will be edited or listed.

```
-list
```

List all the referenced projects and build configurations

```
-add referencedProjectLocation [buildConfigurationName]
```

Adds a new referenced project, specified by the 'referencedProjectLocation', which can be either an absolute path, or a variable relative path (relative to the path variables defined in the project specified by the -project flag). If the buildConfigurationName is specified, a specific build configuration rather than the active build configuration will be referenced.

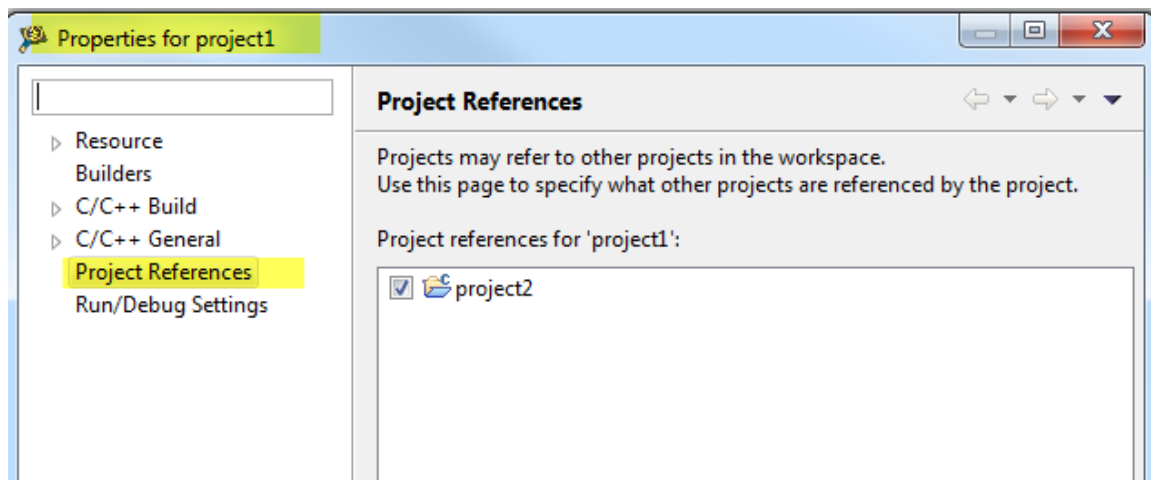
```
-remove referencedProjectLocation [buildConfigurationName]
```

Removes an existing referenced project, specified by the 'referencedProjectLocation', which can be either an absolute path, or a variable relative path (relative to the path variables defined in the project specified by the -project flag). If the buildConfigurationName is specified, only the specific referenced build configuration will be removed, otherwise all references to the specified project will be removed.

**Understanding `--references` : "`--references`" is for *\*Referenced Projects\**. A source file folder can not be as a reference.**

Referenced projects allow you to create build dependencies between projects. If project A is set up as a referenced project for project B, then project A will be built before each project B build. Referenced projects are automatically imported and opened when a project is imported in the workspace, so referenced projects can be used to automatically populate the workspace with a set of projects.

For example, see below setting of project1 in CW 10 IDE, project2 is referenced by project1 in workspace,

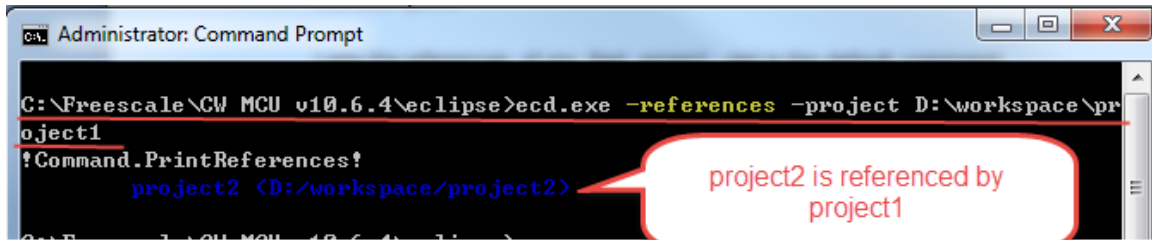


### 3.1 Example 1

Lists the references of project1; -list is the default command.

Command:

***ecd.exe -references -project D:\workspace\project1***



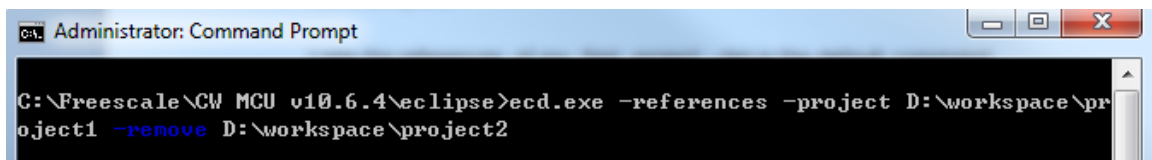
```
Administrator: Command Prompt
C:\Freescall\CW MCU v10.6.4\eclipse>ecd.exe -references -project D:\workspace\project1
!Command.PrintReferences!
project2 <D:/workspace/project2>
```

### 3.2 Example 2

Remove all the references of project2 from all configurations of project1.

Command:

***ecd.exe -references -project D:\workspace\project1 -remove D:\workspace\project2***



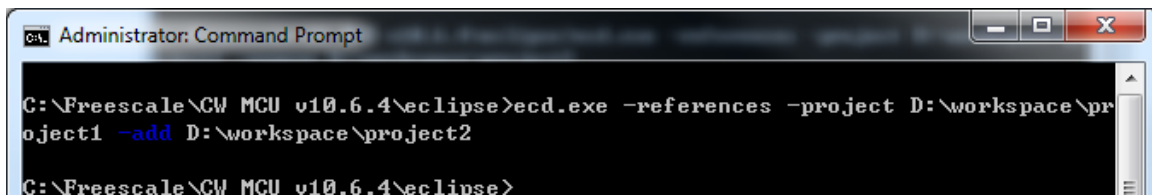
```
Administrator: Command Prompt
C:\Freescall\CW MCU v10.6.4\eclipse>ecd.exe -references -project D:\workspace\project1 -remove D:\workspace\project2
```

### 3.3 Example 3

Adds a reference of the active configuration of project2 to the active configuration of project1

Command:

***ecd.exe -references -project D:\workspace\project1 -add D:\workspace\project2***



```
Administrator: Command Prompt
C:\Freescall\CW MCU v10.6.4\eclipse>ecd.exe -references -project D:\workspace\project1 -add D:\workspace\project2
C:\Freescall\CW MCU v10.6.4\eclipse>
```



## 4. *getOptions, setOptions, updateWorkspace*

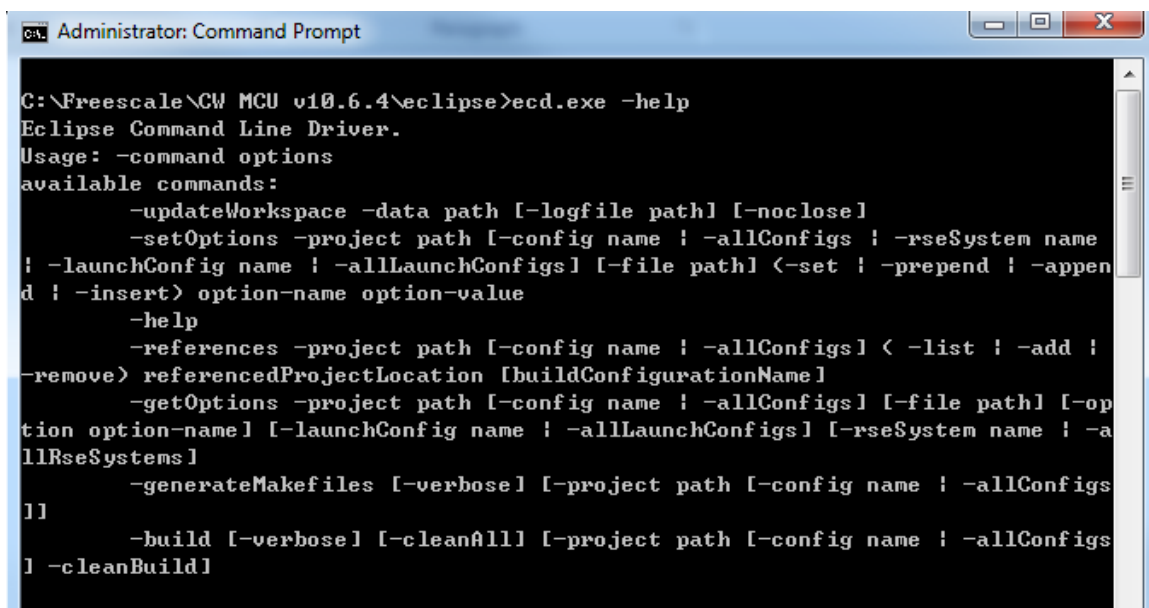
`getOptions` : Prints to the standard output C/C++ managed build, launch configuration or RSE system settings.

`setOptions`: Modifies C/C++ managed build, launch configuration or RSE system settings.

`updateWorkspace`: Updates a workspace .metadata by including any project already located in the workspace file system directory.

the usage of these ecd commands are very simple but not frequently used. If user needs know more information about these command, see *CodeWarrior Common Features Guide.pdf* under CW10 install folder, Or run below command:

`ecd.exe -help`



```
Administrator: Command Prompt
C:\Freescall\CW MCU v10.6.4\eclipse>ecd.exe -help
Eclipse Command Line Driver.
Usage: -command options
available commands:
    -updateWorkspace -data path [-logfile path] [-noclose]
    -setOptions -project path [-config name | -allConfigs | -rseSystem name
| -launchConfig name | -allLaunchConfigs] [-file path] <-set | -prepend | -append | -insert> option-name option-value
    -help
    -references -project path [-config name | -allConfigs] < -list | -add |
-remove> referencedProjectLocation [buildConfigurationName]
    -getOptions -project path [-config name | -allConfigs] [-file path] [-op
tion option-name] [-launchConfig name | -allLaunchConfigs] [-rseSystem name | -a
llRseSystems]
    -generateMakefiles [-verbose] [-project path [-config name | -allConfigs
]]
    -build [-verbose] [-cleanAll] [-project path [-config name | -allConfigs
]] -cleanBuild
```

## 5. Reference

- CodeWarrior Common Features Guide.pdf
- <https://mcuoneclipse.com/2013/10/26/eclipse-command-line-code-generation-with-processor-expert/>
- CodeWarrior help manual.
- <https://mcuoneclipse.com/>