Project migration between e^2 studio and CS+, Notes and Tips

Renesas Electronics Corporation
Software & Digitalization Group
Software Development Division
Tool Department 2
February 12, 2025

R20UT3239EJ0301

AGENDA

- Purpose of this document
- Renesas common project file
- Restrictions
- Conversion of Build variables

Purpose of this document

Existing projects can be migrated between e^2 studio and CS+ through the project import and export functions although there are some restrictions due to the differences in project management policies between the products.

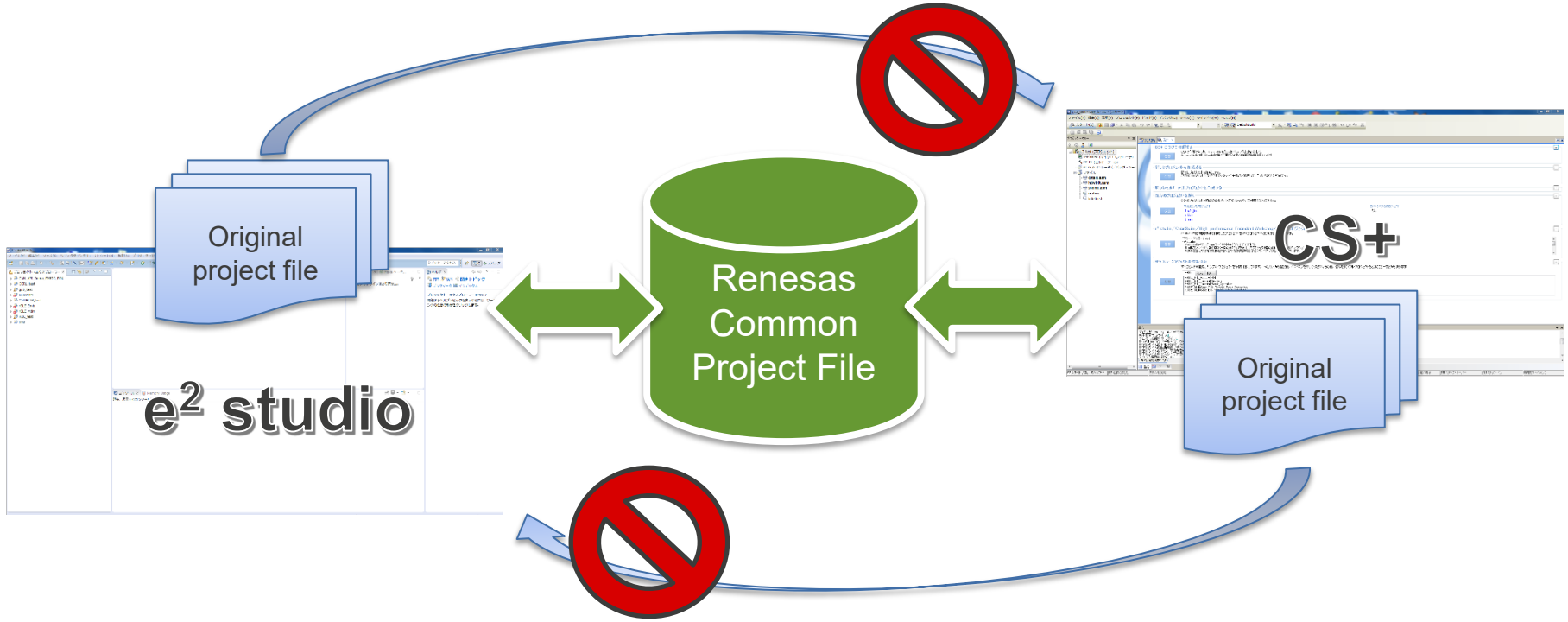
This document describes each restriction on functions that should be noted during project migration when using e^2 studio and CS+.

Note: The tool name "CubeSuite+" was changed to "CS+" from V3.00.00 released on Oct. 1, 2014.
The descriptions in this document apply to both CS+ and CubeSuite+ unless otherwise noted.

Renesas Common Project File

What is Renesas Common Project File?

The format for saving project files differs between e² studio and CS+. Therefore, a project created in either of these products cannot be used in the other product. To solve this problem, Renesas have defined Renesas Common Project File. Each product now supports a new function for importing and exporting Renesas Common Project File. Using Renesas Common Project File enables bidirectional project migration between each product.



Supported versions

The following are the versions of each product that support the import and export function of Renesas Common Project File.

Project	Export	Import
Project for RX family device using C/C++ Compiler Package for RX Family CC-RX V1.00.00 or later version	CubeSuite+ V2.00.00 - V2.02.01 CS+ V3.00 or later version	CubeSuite+ V2.00.00 - V2.02.01 CS+ V3.00 or later version
	e^2 studio V.3.0.0 or later version	e^2 studio V.3.0.0 or later version
Project for RL78 family device using C Compiler Package for RL78 Family CC-RL V1.00.00 or later version	CS+ V3.00 or later version	CubeSuite+ V2.00.00 - V2.02.01 CS+ V3.00 or later version
	e^2 studio V.4.1.0 or later version	e^2 studio V.4.0.0 or later version
Project for RH850 family device using C Compiler Package for RH850 Family CC-RH V2.04.00 or later version	CS+ V8.08 or later version	CS+ V8.08 or later version
	e^2 studio 2023-04 or later version	e^2 studio 2023-04 or later version

* Projects for other than above families do not support the import and export function for Renesas Common Project File.

Import Renesas CA78K0R project

e^2 studio has a function to import a CS+ (CS+ for CACX) project that uses the CA78K0R C compiler for the RL78 family and convert the toolchain to the CC-RL C compiler for the RL78 family.

Unlike the project migration using the Renesas Common Project File explained so far, this project migration uses the mtpj file from CS+ (CS+ for CACX). This project migration only supports one-way migration from CS+ (CS+ for CACX) to e^2 studio.

Output project	Import project
CS+ V3.00 or later version	e^2 studio V.5.1.0 or later version

Restrictions

- This section shows the code or settings that cause differences in operation from that in the original product as a result of project migration, and describes a workaround for the problem that may occur due to each product specification difference.
- The descriptions also include preventive actions that should be taken before migration; please be sure to read through this section before project migration.
- Under "Reason" on each page, the reason for the corresponding restriction and tips for avoiding problems are provided; please also read this information.

No.1
Restriction on output files of Compiler / Assembler

Applied/Not applied:
- Applied:
  This restriction is applied to the conversion in the direction of the arrow.
- Not applied:
  This restriction is not applied to the conversion in the direction of the arrow.

Restriction	e² studio > CS+	CS+ > e² studio
Build errors may occur when migrating a project in which the output folder name or output file name of the Compiler / Assembler object files have been changed.	Applied	Applied

- Workaround
  - Before importing a project, in the original product, revert the output folder and output file name of the Compiler / Assembler object files to default.
- Reason
  - The above restriction allows due to the difference in specifications between each product.
    - e² studio creates folders under the output folder while retaining the hierarchical structure of the source files and outputs the object files to the created folders.
      (Folder names and file names cannot be changed.)
    - CS+ outputs all object files to a single folder.
      (Folder names and file names can be changed.)

No.2
Restriction on output file of Linker

Restriction	e² studio > CS+	CS+ > e² studio
Build errors may occur when migrating a project in which the Linker output folder name or output file name has been changed.	Applied	Applied

- Workaround
  - Before importing a project, in the original project, revert the output folder and output file name of the Linker to default.
- Reason
  - The above restriction allows due to the difference in specifications between each product.
    - e² studio manages the Linker output folder and output file name as "Build Artifact".
      (The Linker file name can be changed, and the Linker folder name cannot be changed.)
    - CS+ manages the Linker output folder name and output file name as Linker options.
      (The Linker folder name and file name can be changed.)

No.3
Restriction on folder structure

Restriction	e² studio > CS+	CS+ > e² studio
The folder structure will be kept the same as the previous product structure, although the folder concept will change depending on the destination product specifications.	Applied	Applied

- Workaround
  - After importing a project, in the destination product, reconfigure the folder structure if necessary.
- Reason
  - The above restriction allows due to the difference in specifications between each product.
    - e² studio has three folder concepts in the project folder structure: physical folder, virtual folder, and linked folder.
      e² studio ports the category folder to physical folder when there is physical folder same location as category folder.
      If there is not physical folder same location as category folder, e² studio ports the category folder to virtual folder.
      e² studio shows all physical folders and files under the project folder to project tree.
      Files which is not used by Build are shown the file as "Exclude from build".
    - CS+ has one folder concept in the project folder structure: category folder (=virtual folder).
      CS+ ports the physical folder / virtual folder / linked folder of e² studio to the category folder (=virtual folder).
      And CS+ does not display the folders and files which are not used by Build.

No.3
Restriction on folder structure (Example 1)

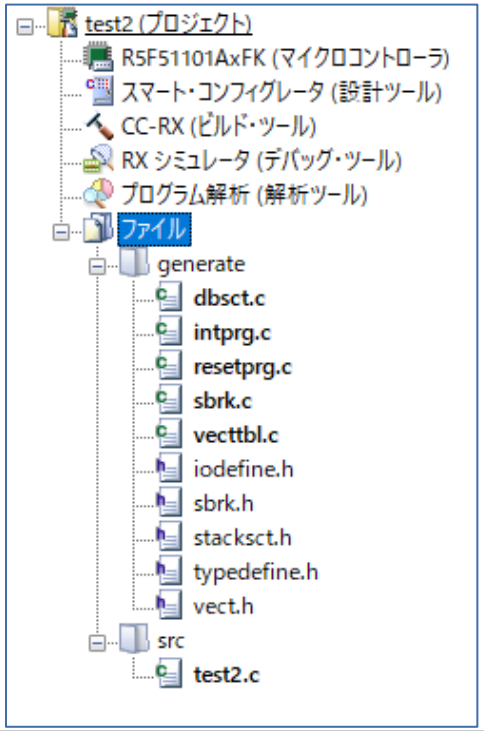
1. Actual folder / file structure

- ¥<Project folder>

  ¥dbsect.c ¥intprg.c ¥restprg.c
  ¥sbrk.c ¥vecttbl.c ¥iodefine.h
  ¥sbrk.h ¥stacksct.h ¥typedefine.h
  ¥vect.h ¥test2.c
  ¥hwsetup.c (Exclude from project)
  ¥lowlvl.src (Exclude from project)
  ¥lowsrc.c (Exclude from project)
  ¥lowsrc.h (Exclude from project)

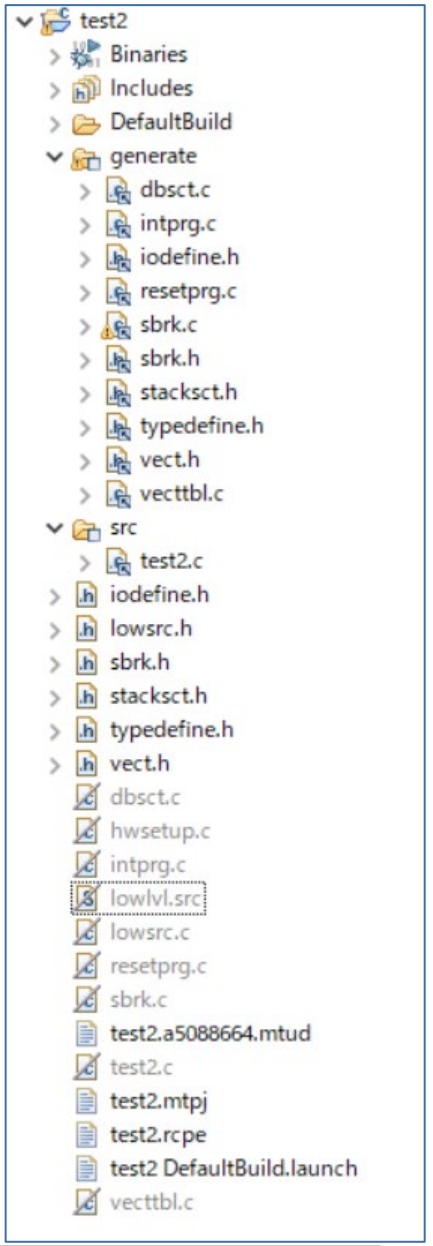
2. Create a project on CS+

Create “generate” category and “src” category as virtual folder. And register the source files on project folder to “generate” category and “src” category.



3. Import a project with e² studio

- Port “Generate” and “src” category to virtual folder.
- Port source files in each category to linked file.
- Port actual files in project folder to actual file as exclude from Build.



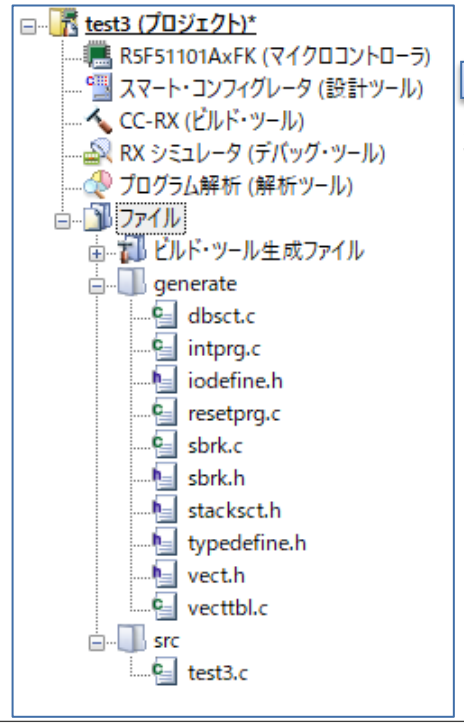
No.3
Restriction on folder structure (Example 2)

1. Actual folder / file structure

- ¥<Project folder>
  - ¥startup (folder)
    - ¥dbsect.c ¥intprg.c ¥restprg.c
    - ¥sbrk.c ¥vecttbl.c ¥iodefine.h
    - ¥sbrk.h ¥stacksct.h ¥typedefine.h
    - ¥vect.h
  - ¥src (folder)
    - ¥test3.c
    - ¥hwsetup.c (Exclude from project)
    - ¥lowlvl.src (Exclude from project)
    - ¥lowsrc.c (Exclude from project)
    - ¥lowsrc.h (Exclude from project)

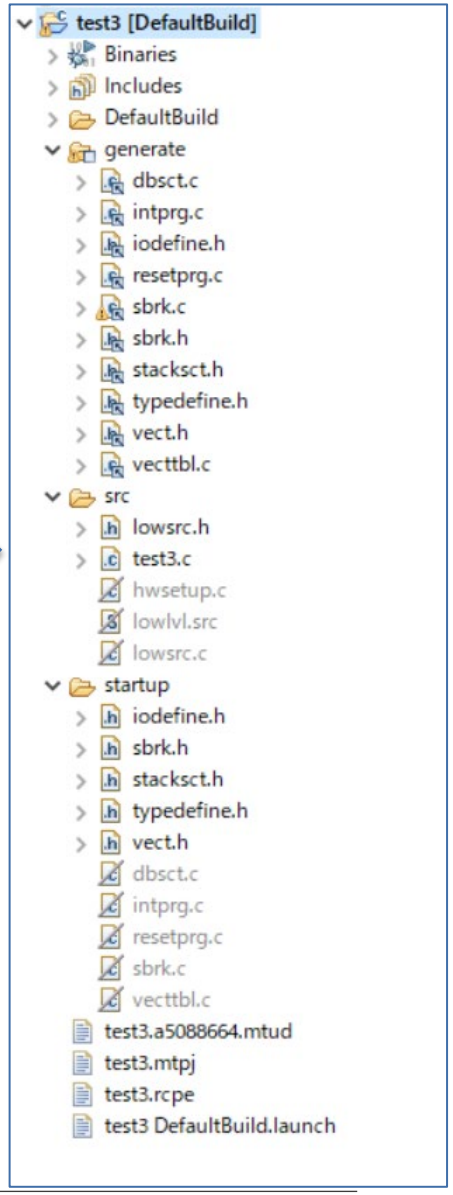
2. Create a project on CS+

Create “generate” category and “src” category as virtual folder. Then register source files on startup folder as actual folder to “generate” category and register source files on src folder as actual folder to “src” category.



3. Import a project with e² studio

- Port “generate” category to virtual folder.
- Port source files in “generate” category to linked file.
- Port “src” category to actual folder.
- Port source files in “src” category to actual file.
- Port startup folder to actual folder but source files are excluded from Build.



No.4
Restriction on same file names

Restriction	e² studio > CS+	CS+ > e² studio
Build errors may occur when a project has duplicate source file names although project migration can work fine.	Applied	Not applied

- Workaround
  - Before importing a project, in the original product, rename the source files so that there are no duplicate source file names.
- Reason
  - The above restriction allows due to the difference in specifications between each product.
    - In e² studio, duplicate source file names can be specified.
      e² studio outputs object files to folders in the same hierarchical structure of the source files.
      Therefore, object files are not conflicted.
    - In CS+, duplicate source file names cannot be specified.
      CS+ outputs object files to a single folder.
      Therefore, object files are conflicted when there are duplicate source file names.

No.5
Restriction on folder names and file names

Restriction	e² studio > CS+	CS+ > e² studio
When source file names or folder names/paths use a prohibited character (#, $, %, full-width characters), the project migration may not be able to work fine.	Applied	Applied

- Workaround
  - Before importing a project, in the original product, change source file names or folder names/paths to not use a prohibited character (#, $, %, full-width characters).
- Reason
  - The characters (#, $) indicating build variables in e² studio and the character (%) indicating placeholders in CS+ have special meaning and must not be used in source file names or folder names.
  - e² studio may not work correctly if the paths to the workspace folder and project folder, or the file names within the project, contain full-width characters.

No.6
Restriction on files (object file / library file) to be linked

Restriction	e² studio > CS+	CS+ > e² studio
Library files and object files registered in the project tree are not treated as files to be linked in the destination product.	Not applied	Applied

- Workaround
  - Before importing a project, in the original product, specify library files and object files in the option setting panel for Linker.
- Reason
  - The above restriction allows due to the difference in specifications between each product.
    - e² studio does not automatically link the library files and object files registered in the project tree. They should be explicitly specified in the option setting panel.
    - CS+ automatically links the library files and object files registered in the project tree. In addition, it also links the files specified in the option setting panel.

No.7
Restriction on linkage order of standard library

Restriction	e² studio > CS+	CS+ > e² studio
The standard libraries are linked in the order prescribed in the specifications of each product.	Applied	Applied

- Workaround
  - After importing a project, in the destination product, review the linkage order.
- Reason
  - The above restriction allows due to the difference in specifications between each product.
    - e² studio links the standard library at the end.
      The order of standard library linkage can be changed through the function for customizing the linkage order.
    - CS+ always links the standard library at the top.

No.8
Restriction on multiple projects migration

Restriction	e² studio > CS+	CS+ > e² studio
The multiple projects cannot be migrated at once.	Applied	Not applied

- Workaround
  - Before importing a project, in the original product, export the projects one by one.
  - After importing a project, in the destination product, import the project one by one.
- Reason
  - The above restriction allows due to the difference in specifications between each product.
    - e² studio can export only one project at a time, although e² studio can import multiple projects at a time.
    - CS+ can export and import multiple projects at once (one main project and multiple sub-projects).

No.9

Restriction on additional command for before/after each phase

Restriction	e² studio > CS+	CS+ > e² studio
Command to be executed before or after each Build processing phase (such as the compiler phase or assembler phase) will be removed, although project migration can work fine.	Not applied	Applied

- Workaround
  - After importing a project, in the destination product, add your commands according to each product specifications.
- Reason
  - The above restriction allows due to the difference in specifications between each product.
    - e² studio allows commands to be added before or after the build process but does not allow commands before and after each phase in the build process.
    - CS+ allows commands to be added before or after the build process, and also before or after each phase in the build process.

No.10
Restriction on additional command for before/after Build

Restriction	e² studio > CS+	CS+ > e² studio
Multiple commands are written in the command before or after Build, 2nd or later commands will be removed, although project migration can work fine.	Not applied	Applied

- Workaround
  - Before importing a project, in the original product, when commands are written in multiple lines, write them in a single BAT file and add the file to the command for before or after build process.
- Reason
  - The above restriction allows due to the difference in specifications between each product.
    - In e² studio, you need to specify '&' as separate character when adding multiple command lines.
    - CS+ allows commands in multiple lines.

No.11
Restriction on additional python console command for Build

Restriction	e² studio > CS+	CS+ > e² studio
Build errors may occur when a project has the commands that can be executed on the Python console, although project migration can work fine.	Applied	Applied

- Workaround
  - After importing a project, in the destination product, review the commands.
- Reason
  - The above restriction allows due to the difference in specifications (Supported commands, command parameters, etc.) between each product.

No.12
Restriction on debugging configuration

Restriction	e² studio > CS+	CS+ > e² studio
The project migration does not include the debugging configuration.	Applied	Applied

- Workaround
  - After importing a project, in the destination product, add the debugging configuration.
- Reason
  - e² studio and CS+ do not support exporting the debugging configuration to the Renesas Common Project File.

No.13
Restriction on RTOS project

Restriction	e² studio > CS+	CS+ > e² studio
The RTOS project information will be removed in the destination product, although project migration can work fine.	Applied	Applied

- Workaround
  - After importing a project, in the destination product, create a new project with RTOS.
- Reason
  - e² studio and CS+ do not support importing the RTOS information in the Renesas Common Project File.

Conversion of Build variables

Conversion of Build variables 1

When migrating a project from CS+ to e2 studio

Original product (CS+)	Meaning	Destination product (e² studio)
%FilePath%	Absolute path of a file	Convert it to "${selected_resource_loc}". This value changes depending on the node selected in the [Project Explorer] view. Please modify it to the appropriate folder in e² studio.
%FileRelativePath%	Relative path of a file from the project folder	Convert it to "${selected_resource_loc}". This value changes depending on the node selected in the [Project Explorer] view. Please modify it to the appropriate folder in e² studio.
%FileDir%	Absolute path of a folder where a file exists	Convert it to "${selected_resource_loc}". This value changes depending on the node selected in the [Project Explorer] view. Please modify it to the appropriate folder in e² studio.
%FileName%	File name	Convert it to "${selected_resource_name}". This value changes depending on the node selected in the [Project Explorer] view. Please modify it to the appropriate folder in e² studio.
%FileLeaf%	File name without extension	Convert it to "$(basename $(<F))".
%FileExt%	Extension of a file	Convert it to "$(suffix $(<F))".

Continued on next page

Conversion of Build variables 1

When migrating a project from CS+ to e² studio (Continue)

Original product (CS+)	Meaning	Destination product (e² studio)
%MainProjectDir%	Absolute path of the main project folder	<e² studio 2023-07 or earlier> Convert it to ${ProjDirPath}. <e² studio 2023-10 or later> Convert it to ${workspace_loc:/${ProjName}}.
%MainProjectName%	Main project name	Convert to ${ProjName}.
%ProjectDir%	Absolute path of the project folder	<e² studio 2023-07 or earlier> Convert it to ${ProjDirPath}. <e² studio 2023-10 or later> Convert it to ${workspace_loc:/${ProjName}}.
%ProjectName%	Project name	Convert it to ${ProjName}.
%BuildModeName%	Build mode name	Convert it to ${ConfigName}.
%ConfigDir%	Absolute path of the configuration folder	Convert it to ${workspace_loc:/${ProjName}/${ConfigName}}.
%MicomToolPath%	Absolute path of the CS+ install folder	Do not convert it. Please modify it to the appropriate folder in e² studio.
%TempDir%	Absolute path of the temporary folder	Convert it to ${TEMP}.
%WinDir%	Absolute path of the Windows system folder	Convert it to ${windir}.
%ActiveProjectDir%	Absolute path of the active project folder	<e2 studio 2023-07 or earlier> Convert to ${ProjDirPath}. <e2 studio 2023-10 or later> Convert to ${workspace_loc:/${ProjName}}.
%ActiveProjectName%	Active project name	Convert it to ${ProjName}.

Continued on next page

Conversion of Build variables 1

When migrating a project from CS+ to e^2 studio (Continue)

Original product (CS+)	Meaning	Destination product (e^2 studio)
%MainProjectMicomName%	Main project microcontroller name	Do not convert it. Please modify it to the appropriate name in e^2 studio.
%ProjectMicomName%	Project microcontroller name	Do not convert it. Please modify it to the appropriate name in e^2 studio.
%ActiveProjectMicomName%	Active project microcontroller name	Do not convert it. Please modify it to the appropriate name in e^2 studio.

Conversion of Build variables 2

When migrating a project from e² studio to CS+

Original product (e² studio)	Meaning	Destination product (CS+)
${workspace_loc}	Absolute path of the workspace folder	Convert it to "%ProjectFolder%¥..¥".
${workspace_loc:/<path>}	Absolute path of the workspace folder/<folder name>	Convert it to "%ProjectFolder%¥..¥<path>".
${workspace_loc:/${ProjName}/<path>}	Absolute path of the workspace folder/<project folder>/<folder name>	[Project which is created on e² studio 2023-07 or earlier] Convert it to "%ProjectFolder%¥..¥%ProjectName%¥<path>". [Project which is created on e² studio 2023-10 or later] Convert it to "%ProjectFolder%¥<path>".
${WorkspaceDirPath}	Absolute path of the workspace folder	Convert it to "%ProjectFolder%¥..¥".
${ProjDirPath}	Absolute path of the project folder	Convert it to "%ProjectFolder%".
${ProjName}	Project name	Convert it to "%ProjectName%".
${CONFIGDIR}	Absolute path of the configuration folder	Convert it to "%CondifDir%".
${ConfigName}	Configuration name	Convert it to "%BuildModeName%".

Continued on next page

Conversion of Build variables 2

When migrating a project from e² studio to CS+ (Continue)

Original product (e² studio)	Meaning	Destination product (CS+)
${selected_resource_loc}	Absolute path of the selection item in the project tree	Convert it to "%FullFile%".
${selected_resource_name}	Name of the selection item in the project tree	Convert it to "%FileName%".
${selected_resource_path}	Absolute path of the selection item in the project tree	Do not convert it. Please modify it to the appropriate folder in CS+.
${eclipse_home}	Absolute path of the e2 studio install folder	Do not convert it. Please modify it to the appropriate folder in CS+.
${TEMP}	Absolute path of the temporary folder	Convert it to "%TempDir%".
${TMP}	Absolute path of the temporary folder	Convert it to "%TempDir%".
${windir}	Absolute path of the Windows system folder	Convert it to "%WinDir%".
${TCINSTALL}	Absolute path of the toolchain folder	Remove it. Please Add the appropriate folder in CS+.

Renesas.com