e² studioとCS+のプロジェクト移行に関する機能制限

ルネサス エレクトロニクス株式会社
IoT・インフラ事業本部
ソフトウェア開発統括部
2022/2/21

R20UT3239JJ0300

アジェンダ

本書の目的

e² studio と CS+では、プロジェクトのインポート・エクスポート機能を介して
作成済みのプロジェクトを変換することが可能です。
ただし、プロジェクト管理方法の違いなどによりいくつかの制限があります。

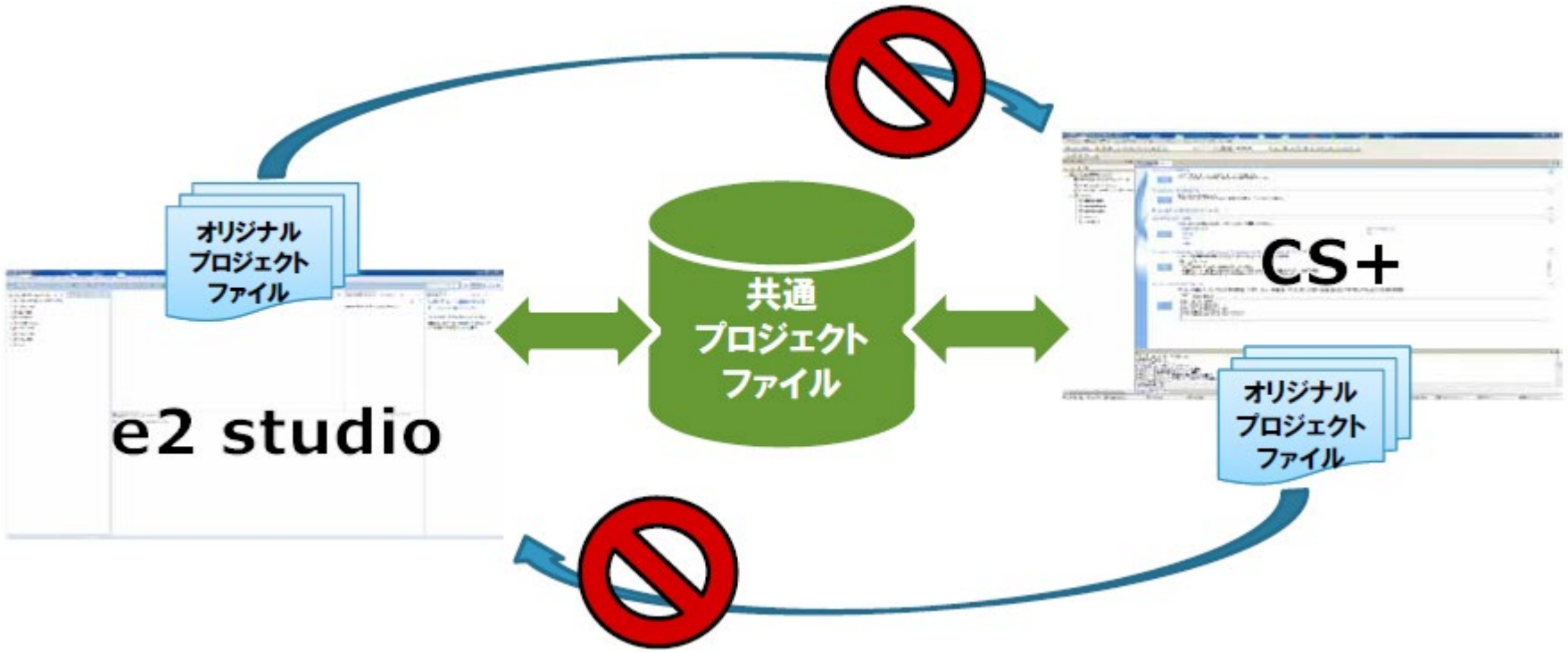
本書では、e² studio 2022-01およびCS+ V8.07.00を使用した場合の
プロジェクトを変換する際に起こり得る機能制限とその背景について解説します。

注：CubeSuite+は、2014年10月1日リリースのV3.00.00より製品名をCS+に変更いたしました。
以降の記載事項は特にことわりのない限りCS+およびCubeSuite+の両製品について適用される内容です。

プロジェクト変換方法

プロジェクト変換とは

e2 studioとCS+は、保存するプロジェクト・ファイルの形式が異なります。
そのため、片方の製品で作成したプロジェクトをもう片方の製品で使用することが出来ません。
これを解決するために、Renesas共通のプロジェクト・ファイルを出力する機能を各製品でサポートしました。
Renesas共通プロジェク・ファイルを使用して双方向のプロジェクトの変換を可能にしています。



プロジェクト変換方法

以下の手順によりCS+とe² studioの間でプロジェクト変換を行うことができます。

- プロジェクトのインポート（CS+ => e² studio）
  CS+は自動的に「Renesas共通プロジェクト・ファイル（*.rcpe）」を生成します。
  このファイルを使用してプロジェクトを変換します。

  1. e² studioのメニュー［ファイル］>［インポート］から［インポート］ダイアログを表示し、「一般 > Renesas CC-RX/CC-RL（CS+）プロジェクト」を選択して「Renesas共通プロジェクトファイル（*.rcpe）」を指定します。
  2. インポートしたプロジェクトがe² studioで使用可能になります。

- プロジェクトのエクスポート（e² studio => CS+）
  e² studioはビルド時に「Renesas共通プロジェクト・ファイル（*.rcpc）」を生成します。
  このファイルを使用してプロジェクトを変換します。

  1. CS+のメニュー「プロジェクト」→「プロジェクトを開く」からe² studioが出力した「Renesas共通プロジェクトファイル（*.rcpc）」を指定します。
  2. インポートしたプロジェクトがCS+で使用可能になります。

プロジェクト変換の組み合わせ
同一ツールチェーンへ変換

- RXファミリ
  RXファミリ用C/C++コンパイラ CC-RX V1.00.00およびそれ以降のバージョンに対応

Renesas共通プロジェクトの出力	Renesas共通プロジェクトの読み込み
CubeSuite+ V2.00.00～V2.02.01 CS+ V3.00およびそれ以降のバージョン	e² studio V.3.0.0およびそれ以降のバージョン
e² studio V.3.0.0およびそれ以降のバージョン	CubeSuite+ V2.00.00～V2.02.01 CS+ V3.00およびそれ以降のバージョン

- RL78ファミリ
  RL78ファミリ用Cコンパイラ CC-RL V1.00.00およびそれ以降のバージョンに対応

Renesas共通プロジェクトの出力	Renesas共通プロジェクトの読み込み
CS+ V3.00およびそれ以降のバージョン	e² studio V.4.0.0およびそれ以降のバージョン
e² studio V.4.1.0およびそれ以降のバージョン	CubeSuite+ V2.00.00～V2.02.01 CS+ V3.00およびそれ以降のバージョン

- その他のファミリ
  未対応

プロジェクト変換の組み合わせ
異なるツールチェーンへ変換

- RXファミリ
  RXファミリ用C/C++コンパイラ以外のツールチェーンを使用したプロジェクトをRXファミリ用C/C++コンパイラを使用したプロジェクトへ変換
  - 未対応
- RL78ファミリ
  RL78ファミリ用CコンパイラCA78K0R（RL78デバイスのみ）からRL78ファミリ用CコンパイラCC-RLへ変換
  - e^2 studioは、 RL78ファミリ用CコンパイラCA78K0Rをサポートしていません。

プロジェクトの出力	プロジェクトの読み込み
CS+ V3.00およびそれ以降のバージョン	e^2 studio V.5.1.0およびそれ以降のバージョン

※ このプロジェクト変換のみCS+のプロジェクト・ファイル（*.mtpj）を使用します。

※ ソースファイルの変換は実施しません。ソースファイルを変換する方法は、下記ページをご参照ください。
RL78ファミリ用Cコンパイラへの移行支援 | Renesas

- その他のファミリ
  未対応

制限事項

- この章では、プロジェクト変換した結果、元のビルド環境とは異なる動作になる表記や設定項目を挙げ、それぞれについて起こり得る問題を回避する具体的な対策を示します。
- 変換前に行っておくべき予防策も含まれますのでプロジェクト変換を行う前に是非ご一読ください。
- 「背景」には、各制限事項がなぜ起こるかの理由や、問題を回避するためのヒントになる情報を記載しておりますので併せてお読みください。

コンパイラ、アセンブラが出力するファイルに関する制限

制限内容	e² studio -> CS+	CS+ -> e² studio
コンパイラ、アセンブラが出力するオブジェクト・ファイル（.obj）のフォルダ名またはファイル名を変更したプロジェクトを変換すると、ビルド・エラーが発生する場合があります。	該当	該当

該当/非該当について
- 該当
  矢印方向の変換について
  本制限が当てはまります
- 非該当
  矢印方向の変換について
  本制限は当てはまりません

- 回避方法
  - プロジェクトを変換する前に、オブジェクト・ファイルの出力フォルダ名および出力ファイル名をデフォルト設定に戻してください。
- 背景
  - コンパイラおよびアセンブラが出力するオブジェクト・ファイルのフォルダ名およびファイル名は、各IDEが自身の仕様に合わせて変換します。
    - CS+は、すべてのオブジェクト・ファイルを１つのフォルダに出力するように出力フォルダ設定を変更します。
    - e² studioは、ソースファイルの階層構造を維持したフォルダを作成してオブジェクト・ファイルを出力するように出力フォルダ設定を変更します。
  - 各製品の仕様が異なるためです。

リンカが出力するファイルに関する制限

制限内容	e² studio -> CS+	CS+ -> e² studio
リンカが出力するファイル（.abs等）のフォルダ名またはファイル名を変更したプロジェクトを変換すると、ビルド・エラーが発生する場合があります。	該当	該当

- 回避方法
  - プロジェクトを変換する前に、出力フォルダ名および出力ファイル名をデフォルト設定に戻してください。
- 背景
  - リンカが出力するファイルのフォルダ名およびファイル名は、各IDEが自身の仕様に合わせて変換します。
    - CS+は、フォルダ名およびファイル名をリンカのオプションとして管理します。
    - e² studioは、リンカのオプションとは別の設定項目（ビルド成果物）で管理します。
  - 各製品の仕様が異なるためです。

フォルダ構成に関する制限

制限内容	e² studio -> CS+	CS+ -> e² studio
フォルダ構成は維持しますが、フォルダの取り扱いをIDEの仕様に従って変更します。	該当	該当

- 回避方法
  - プロジェクトを変換した後で、必要に応じてフォルダ構成を変更してください。
- 背景
  - フォルダ構成は維持しますが、各製品仕様の違いによりフォルダ構成の表示方法が異なります。
    次ページの表示例を参照ください。
    - CS+は、e² studioの物理フォルダ／仮想フォルダ／リンク・フォルダをカテゴリとして引き継ぎます。ビルド対象ではないフォルダやファイルは表示しません。
    - e² studioは、CS+のカテゴリを物理フォルダ（カテゴリと同名の物理フォルダがある場合）、および仮想フォルダ（カテゴリと同名の物理フォルダがない場合）として引き継ぎます。プロジェクト・フォルダ以下のすべての物理的なフォルダおよびファイルをプロジェクト・ツリーに表示します。ビルドで使用しないファイルは、ビルド対象外として扱います。

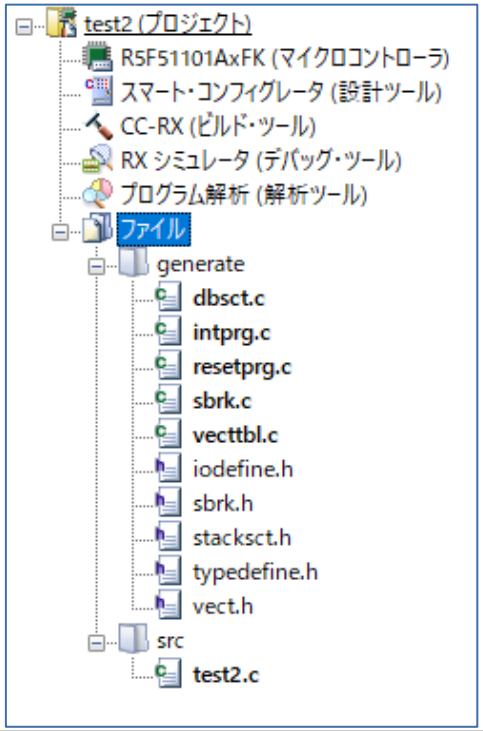
フォルダ構成に関する制限（例１）

① 実際のフォルダ/ファイル構成

- ¥<プロジェクト・フォルダ>
  ¥dbsect.c　¥intprg.c　¥restprg.c
  ¥sbrk.c　¥vecttbl.c　¥iodefine.h
  ¥sbrk.h　¥stacksct.h　¥typedefine.h
  ¥vect.h　¥test2.c
  ¥hwsetup.c（プロジェクトから除外）
  ¥lowlvl.src （プロジェクトから除外）
  ¥lowsrc.c （プロジェクトから除外）
  ¥lowsrc.h （プロジェクトから除外）

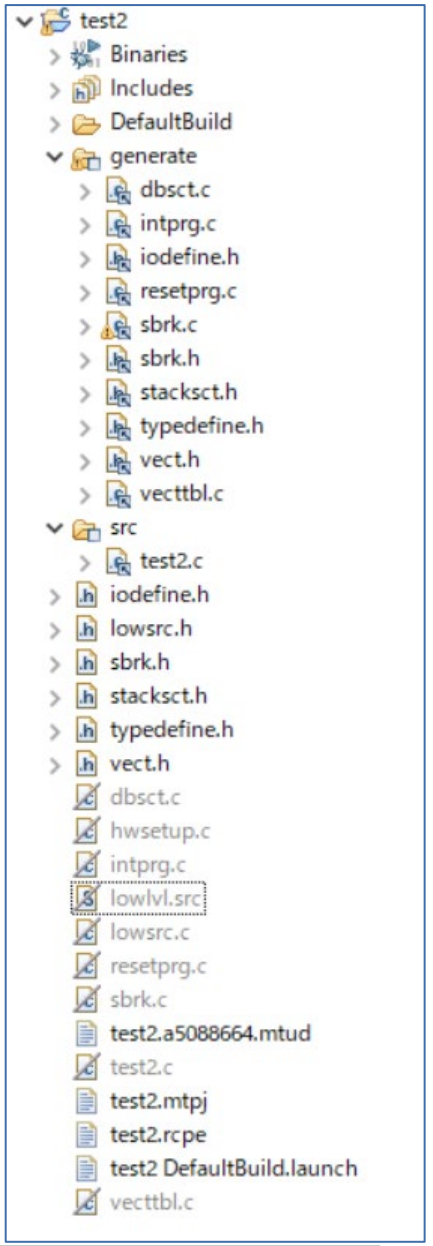
② CS+でプロジェクトを作成

仮想フォルダ「generate」および「src」を作成して、プロジェクト・フォルダにあるソース・ファイルを作成した仮想フォルダに、登録します。



③ e² studioのプロジェクトに変換

Generate、srcは仮想フォルダとして、ソース・ファイルはリンク・ファイルとして扱われます。実体のファイルはビルド対象外としてプロジェクト・ツリーに表示されます。



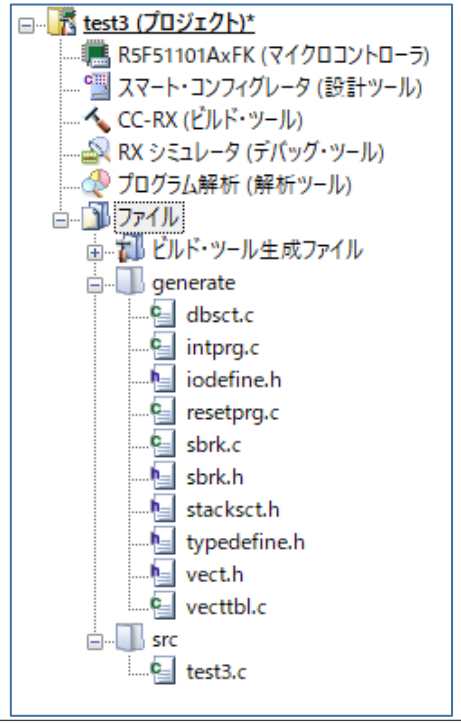
フォルダ構成に関する制限（例２）

① 実際のフォルダ/ファイル構成

- ¥<プロジェクト・フォルダ>
  ¥startupフォルダ
  ¥dbsect.c　¥intprg.c　¥restprg.c
  ¥sbrk.c　¥vecttbl.c　¥iodefine.h
  ¥sbrk.h　¥stacksct.h　¥typedefine.h
  ¥vect.h
  ¥srcフォルダ
  ¥test3.c
  ¥hwsetup.c（プロジェクトから除外）
  ¥lowlvl.src （プロジェクトから除外）
  ¥lowsrc.c （プロジェクトから除外）
  ¥lowsrc.h （プロジェクトから除外）

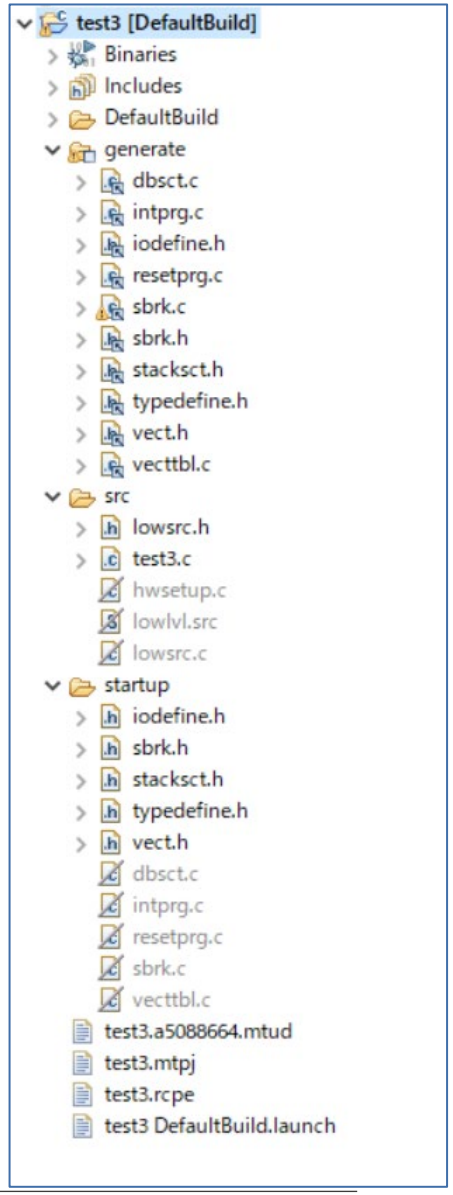
② CS+でプロジェクトを作成

仮想フォルダ「generate」および「src」を作成して、startup（実体）フォルダにあるソース・ファイルを「generate」に登録し、src（実体）にあるソース・ファイルを「src」に登録します。



③ e² studioのプロジェクトに変換

Generateは仮想フォルダ、その中のファイルはリンクファイル、srcは実体のフォルダ、その中のファイルは実体のファイル（一部ビルド対象外）になります。
startupフォルダは実体のフォルダとして表示され、ソース・ファイルはビルド対象外になります。



同名ファイルに関する制限

制限内容	e² studio -> CS+	CS+ -> e² studio
プロジェクトに同名のソース・ファイルが複数登録されている場合、プロジェクトは変換できますが、変換後のビルドでエラーが発生することがあります。	該当	非該当

- 回避方法
  - プロジェクトを変換する前に、ファイル名の重複がないようにファイル名を変更してください。
- 背景
  - CS+は、すべてのオブジェクト・ファイルを1つのフォルダに出力するため、同名のソース・ファイルが存在するとオブジェクト・ファイルの衝突が発生します。そのため、プロジェクトに同名ソース・ファイルを登録することができません。
  - e² studioは、オブジェクト・ファイルをソース・ファイルの階層構造を維持して出力することで、同名のソース・ファイルが存在してもオブジェクト・ファイルの衝突が発生しません。そのため、同名ソース・ファイルを登録することができます。
  - 各製品の仕様が異なるためです。

フォルダ名、ファイルに関する制限

制限内容	e² studio -> CS+	CS+ -> e² studio
フォルダ名やファイル名に使用できない文字（#、$、%）が含まれていると、正しくプロジェクトを変換できない場合があります。	該当	該当

- 回避方法
  - プロジェクトを変換する前に、フォルダ名やファイル名で「#、$、%」を使用しない名前に変更してください。
- 背景
  - 「#、$、%」 は、e² studioのビルド変数、およびCS+のプレースホルダを表すための特別な意味を持つ文字のため、それらの文字はフォルダ名やファイル名に使用できません。

リンク対象ファイルに関する制限

制限内容	e² studio -> CS+	CS+ -> e² studio
プロジェクト・ツリーに登録されたライブラリ・ファイル、およびオブジェクト・ファイルはリンク対象になりません。	非該当	該当

- 回避方法
  - プロジェクトを変換する前に、ライブラリ・ファイル、およびオブジェクト・ファイルをリンカのオプション設定パネルで指定してください。
- 背景
  - e² studioは、プロジェクトに登録されたライブラリ・ファイルおよびオブジェクト・ファイルをリンク対象にしません。そのため、明示的にオプション設定パネルで指定する必要があります。
  - CS+は、プロジェクトに登録されたライブラリ・ファイル、およびオブジェクト・ファイルを自動的にリンク対象にします。また、オプション設定パネルで指定されたファイルもリンク対象にします。
  - 各製品の仕様が異なるためです。

ビルド（各フェーズの前後）の追加コマンドに関する制限

制限内容	e² studio -> CS+	CS+ -> e² studio
ビルドの各フェーズ（コンパイラやアセンブラ等）の前後に追加したコマンドは、引き継ぐことができません。	非該当	該当

- 回避方法
  - プロジェクトを変換した後で、各IDEの仕様に従ってコマンドを追加してください。
- 背景
  - e² studioは、ビルドの前後にコマンドを追加することができます。ビルドの各フェーズの前後にはコマンドを追加することができません。
  - CS+は、ビルドの前後、ビルドの各フェーズの前後にコマンドを追加することができます。
  - 各製品の仕様が異なるためです。

ビルド前後の追加コマンドに関する制限

制限内容	e² studio -> CS+	CS+ -> e² studio
ビルド前後の追加コマンドを引き継ぐことができますが、複数行のコマンドが指定されている場合は、引き継ぐことができません。	非該当	該当

- 回避方法
  - プロジェクトを変換する前に、複数行のコマンドを1つのバッチ・ファイルに記載して、そのバッチ・ファイルを指定してください。
- 背景
  - e² studioは、複数行のコマンドを追加するときは、区切り子として「&」を指定する必要があります。
  - CS+は複数行のコマンドを追加することが可能です。
  - 各製品の仕様が異なるためです。

ビルドの追加コマンド（Pythonコンソール）に関する制限

制限内容	e² studio -> CS+	CS+ -> e² studio
ビルド前後の追加コマンドとしてPythonコンソールで使用可能なコマンドを登録している場合、そのコマンドは引き継ぐことができません。	非該当	該当

- 回避方法
  - なし
- 背景
  - e² studioは、Pythonコマンドによるビルド制御機能をサポートしていません。
  - CS+は、Pythonコンソールで使用可能なコマンドをビルド時に実行するコマンドとして追加することが可能です。
  - 各製品の仕様が異なるためです。

デバッガ設定に関する制限

制限内容	e² studio -> CS+	CS+ -> e² studio
デバッガの設定を引き継ぐことはできません。	該当	該当

- 回避方法
  - プロジェクトを変換した後で、デバッガの設定を追加してください。
- 背景
  - デバッガの設定はプロジェクト変換処理の対象ではありません。

ビルド変数とプレースホルダの変換

CS+ プレースホルダ -> e² studio ビルド変数 変換表（1/2）

CS+ プレースホルダ	意味	e² studio ビルド変数
%FilePath%	ファイルの絶対パス	${selected_resource_loc}に変換します。正しい意味の設定に書き換えてください。
%FileRelativePath%	ファイルのプロジェクトからの相対パス	${selected_resource_loc}に変換します。正しい意味の設定に書き換えてください。
%FileDir%	ファイルが存在するフォルダの絶対パス	${selected_resource_loc}に変換します。正しい意味の設定に書き換えてください。
%FileName%	ファイル名（パスは含めない）	${selected_resource_name}に変換します。正しい意味の設定に書き換えてください。
%FileLeaf%	拡張子を除くファイル名（パスは含めない）	$(basename $(<F))に変換します。
%FileExt%	ファイルの拡張子	$(suffix $(<F))に変換します。
%MainProjectDir%	メイン・プロジェクトのフォルダの絶対パス	${ProjDirPath}に変換します。
%MainProjectName%	メイン・プロジェクト名	${ProjName}に変換します。
%ProjectDir%	プロジェクトのフォルダの絶対パス	${ProjDirPath}に変換します。
%ProjectName%	プロジェクト名	${ProjName}に変換します。
%BuildModeName%	ビルド・モード名	${ConfigName}に変換します。
%ConfigDir%	コンフィグレーションのフォルダの絶対パス	${workspace_loc:/${ProjName}/${ConfigName}}に変換します。

CS+ プレースホルダ -> e² studio ビルド変数 変換表（2/2）

CS+ プレースホルダ	意味	e² studio ビルド変数
%MicomToolPath%	CS+のインストール先フォルダの絶対パス	${MicomToolPath}に変換します。e² studio ビルド変数に「${MicomToolPath}」は定義されていません。正しい意味の設定に書き換えてください。
%TempDir%	テンポラリ・フォルダの絶対パス	${TEMP}に変換します。
%WinDir%	Windowsシステム・フォルダの絶対パス	${windir}に変換します。
%ActiveProjectDir%	アクティブ・プロジェクトのフォルダの絶対パス	${ProjDirPath}に変換します。アクティブを表すビルド変数がありません。正しい意味の設定に書き換えてください。
%ActiveProjectName%	アクティブ・プロジェクト名	${ProjName}に変換します。アクティブを表すビルド変数がありません。正しい意味の設定に書き換えてください。
%MainProjectMicomName%	メイン・プロジェクトのマイクロコントローラ名	${MainProjectMicomName}に変換します。e² studio ビルド変数に「${MainProjectMicomName}」は定義されていません。正しい意味の設定に書き換えてください。
%ProjectMicomName%	プロジェクトのマイクロコントローラ名	${ProjectMicomName}に変換します。e² studio ビルド変数に「${ProjectMicomName}」は定義されていません。正しい意味の設定に書き換えてください。
%ActiveProjectMicomName%	アクティブプロジェクトのマイクロコントローラ名	${ActiveProjectMicomName}に変換します。e² studio ビルド変数に「${ActiveProjectMicomName}」は定義されていません。正しい意味の設定に書き換えてください。

- パスを設定するオプションにパスを意味するプレースホルダ以外が指定されている場合は、「${ProjDirPath}/<ビルド変数>」に変換します。

e² studio ビルド変数 -> CS+ プレースホルダ 変換表（1/2）

e² studio ビルド変数	意味	CS+ プレースホルダ
${workspace_loc}	ワークスペースフォルダの絶対パス	%ProjectFolder%/../に変換します。
${workspace_loc:/<path>}	ワークスペースフォルダ¥<path>の絶対パス	%ProjectFolder%¥..¥<path>に変換します。
${workspace_loc:/${ProjName}/<path>}	プロジェクトフォルダ¥<path>の絶対パス	%ProjectFolder%¥..¥%ProjectName%¥<path>に変換します。
${WorkspaceDirPath}	ワークスペースフォルダの絶対パス	%ProjectFolder%/..に変換します。
${ProjDirPath}	プロジェクトフォルダの絶対パス	%ProjectFolder%に変換します。
${ProjName}	プロジェクト名	%ProjectName%に変換します。
${CONFIGDIR}	コンフィグレーションフォルダの絶対パス	%CondifDir%に変換します。
${ConfigName}	コンフィグレーション名	%BuildModeName%に変換します。
${selected_resource_loc}	選択されたリソースフォルダの絶対パス	%FullFile%に変換します。正しい意味の設定に書き換えてください。
${selected_resource_name}	選択されたリソース名	%FileName%に変換します。正しい意味の設定に書き換えてください。
${selected_resource_path}	選択されたリソースフォルダワークスペースホルダからの相対パス	変換されずに、${selected_resource_path}のまま出力されます。正しい意味の設定に書き換えてください。
${eclipse_home}	e² studioのインストール先フォルダの絶対パス	変換されずに、${eclipse_home}のまま出力されます。正しい意味の設定に書き換えてください。

e² studio ビルド変数 -> CS+ プレースホルダ 変換表（2/2）

e² studio ビルド変数	意味	CS+ プレースホルダ
${TEMP}	テンポラリ・フォルダの絶対パス	%TempDir%に変換されます。
${TMP}	テンポラリ・フォルダの絶対パス	削除されます。正しい意味の設定を追加してください。
${windir}	Windowsシステム・フォルダの絶対パス	%WinDir%に変換されます。

Renesas.com