

e² studioとCS+間の 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
ソフトウェア&デジタライゼーショングループ
ソフトウェア開発統括部
2025/2/12

R20UT3239JJ0301

アジェンダ

- [本書の目的](#)
- [移行手順](#)
- [制限事項](#)
- [ビルド変数の変換](#)

本書の目的

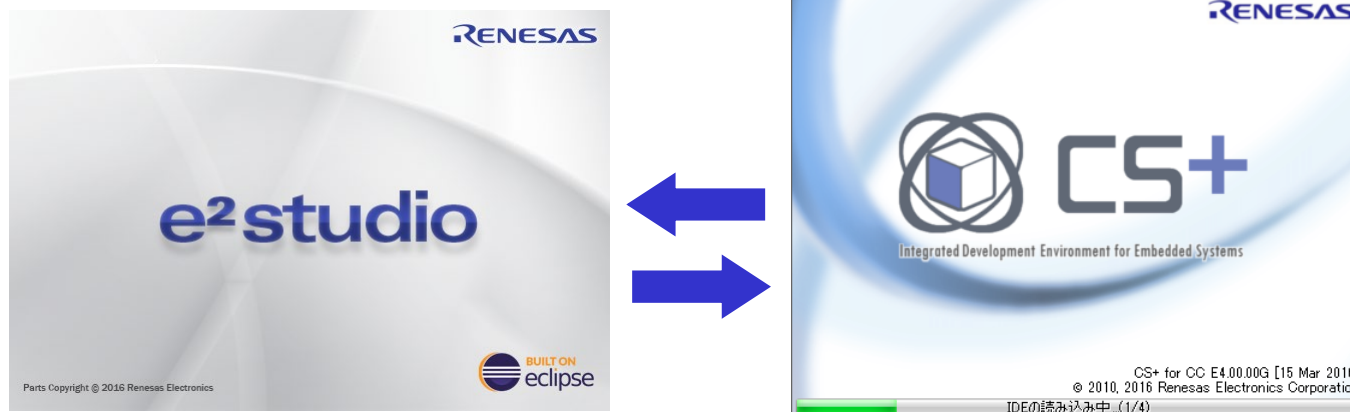
統合開発環境 e² studioおよびCS+は、プロジェクトのインポート／エクスポート機能を介して、一方の製品で作成したプロジェクトを他方の製品に移行することが可能です。

ただし、製品間でプロジェクトを移行するには、製品ごとのプロジェクト管理方法の違いによりいくつかの制限があります。

本書では、e² studioおよびCS+で作成したプロジェクトをe² studioとCS+間で双方向に移行する際の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。



Renesas共通プロジェクト・ファイル

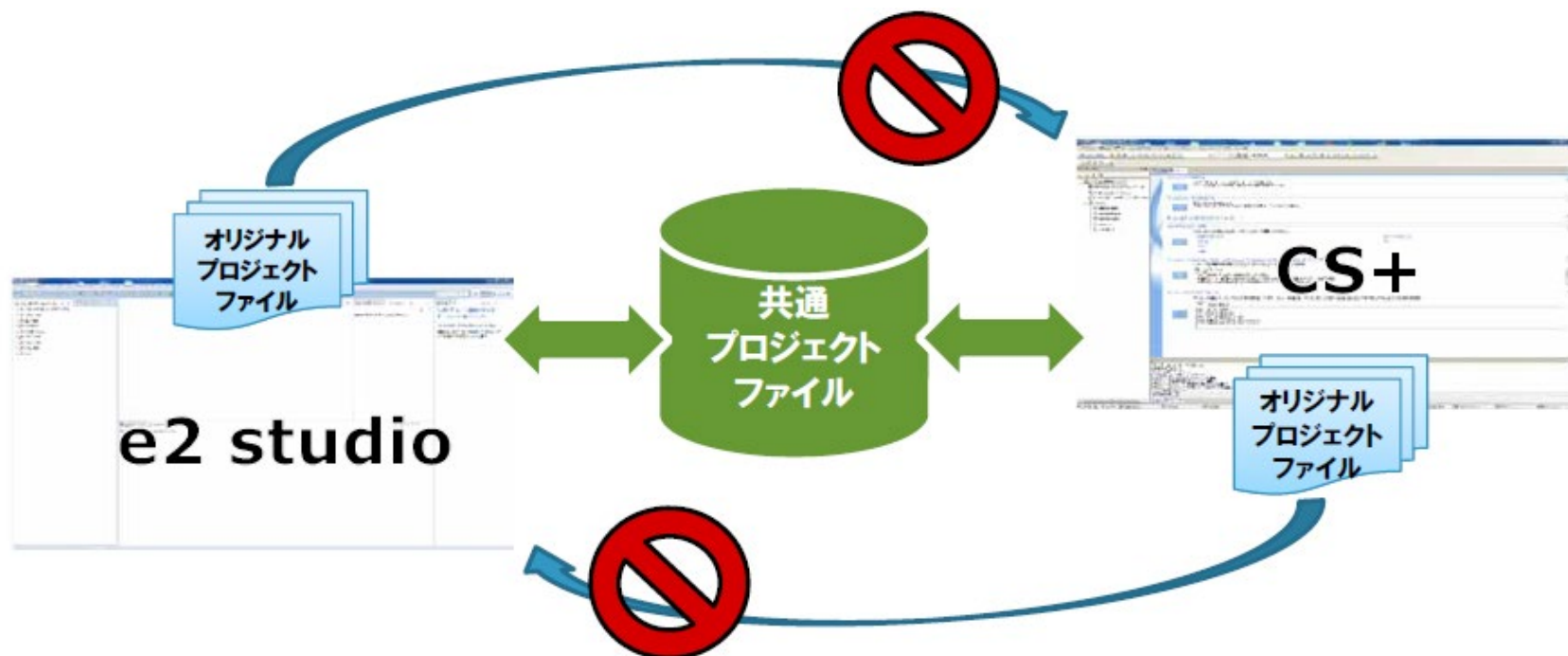
Renesas共通プロジェクト・ファイルとは？

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、一方の製品で作成したプロジェクトを他方の製品で使うことができません。

これを解決するために、Renesas共通のプロジェクト・ファイルを定義しました。そして、そのファイルをインポートおよびエクスポートする機能をそれぞれの製品でサポートしました。

Renesas共通プロジェクト・ファイルを使用して、一方の製品から他方の製品へプロジェクトの移行を可能にしています。



対応バージョン

Renesas共通プロジェクト・ファイルのインポート／エクスポート機能をサポートする各製品のバージョンは以下の通りです。

プロジェクト	エクスポート機能	インポート機能
RXファミリ用プロジェクト (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)	CubeSuite+ V2.00.00～V2.02.01 CS+ V3.00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
	e ² studio V.3.0.0およびそれ以降のバージョン	e ² studio V.3.0.0およびそれ以降のバージョン
RL78ファミリ用プロジェクト (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)	CS+ V3.00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
	e ² studio V.4.1.0およびそれ以降のバージョン	e ² studio V.4.0.0およびそれ以降のバージョン
RH850ファミリ用プロジェクト (RH850ファミリ用Cコンパイラ CC-RH V2.04.00および それ以降のバージョン)	CS+ V8.08およびそれ以降のバージョン	CS+ V8.08およびそれ以降のバージョン
	e ² studio 2023-04およびそれ以降のバージョン	e ² studio 2023-04およびそれ以降のバージョン

※ その他のファミリ用プロジェクトは、Renesas共通プロジェクト・ファイルのインポート／エクスポート機能をサポートしていません。

Renesas CA78K0Rプロジェクトのインポート

e² studioには、CS+（CS+ for CACX）のRL78ファミリ用CコンパイラCA78K0Rを使用したプロジェクトをインポートしてツールチェーンをRL78ファミリ用CコンパイラCC-RLに変換する機能があります。

ここまで説明したRenesas共通プロジェクト・ファイルを使用したプロジェクトの移行とは異なり、CS+（CS+ for CACX）のmtpjファイルを使用してプロジェクトの移行を行います。
このプロジェクトの移行は、CS+（CS+ for CACX）からe² studioへの一方向のみをサポートします。

プロジェクトの出力	プロジェクトのインポート
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

制限事項

- この章では、製品間でプロジェクト移行した結果、移行元の製品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。
- 移行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移行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。

No.1

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

該当/非該当について

- ・ 該当
矢印方向の移行について本制限が当てはまります。
- ・ 非該当
矢印方向の移行について本制限は当てはまりません。

制限内容	e ² studio > CS+	CS+ > e ² 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移行すると、移行先の製品でビルド・エラーを発生する場合があります。	該当	該当

- 回避策
 - プロジェクトを移行する前に移行元の製品上で、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
 - 製品ごとの仕様が異なるためです。
 - e² studioは、出力フォルダ配下にソース・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・ファイルを出力します。
（フォルダ名およびファイル名を変更できません。）
 - 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します。
（フォルダ名およびファイル名を変更できます。）

No.2

リンクが出力するファイルに関する制限

制限内容	e ² studio > CS+	CS+ > e ² studio
リンク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移行すると、移行先の製品でビルド・エラーが発生する場合があります。	該当	該当

- 回避策

- プロジェクトを移行する前に移行元の製品上で、リンクが出力するフォルダ名およびファイル名をデフォルト設定に戻してください。

- 背景

- 製品ごとの仕様が異なるためです。
 - e² studioは、ファイル名を成果物として管理します。
（ファイル名を変更できますが、フォルダ名を変更できません。）
 - CS+は、フォルダ名およびファイル名をリンクのオプションとして管理します。
（フォルダ名およびファイル名を変更できます。）

No.3

フォルダ構成に関する制限

制限内容	e ² studio > CS+	CS+ > e ² studio
フォルダ構成は維持しますが、フォルダの取り扱いを移行先製品の仕様に従って変更します。次ページの表示例をご参照ください。	該当	該当

- 回避策
 - プロジェクトを移行した後に移行先の製品上で、必要に応じてフォルダ構成を変更してください。
- 背景
 - 製品ごとの仕様が異なるためです。
 - e² studioは物理フォルダ／仮想フォルダ／リンク・フォルダという概念の階層構造を持ちます。そのため、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）、または仮想フォルダ（カテゴリと同名の物理フォルダがない場合）に変換します。プロジェクト・フォルダ以下のすべての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。
 - CS+は、カテゴリ（仮想フォルダ）という概念の階層構造を持ちます。そのため、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に変換します。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。

No.3

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 ¥resetprg.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)

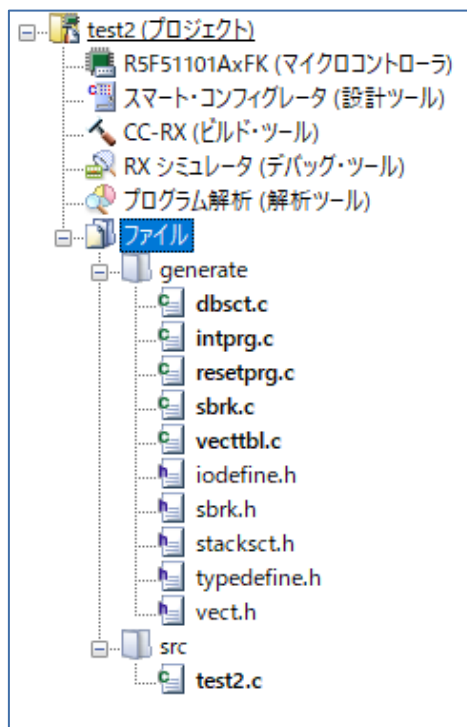
¥lowlvl.src (プロジェクトから除外)

¥lowsrc.c (プロジェクトから除外)

¥lowsrc.h (プロジェクトから除外)

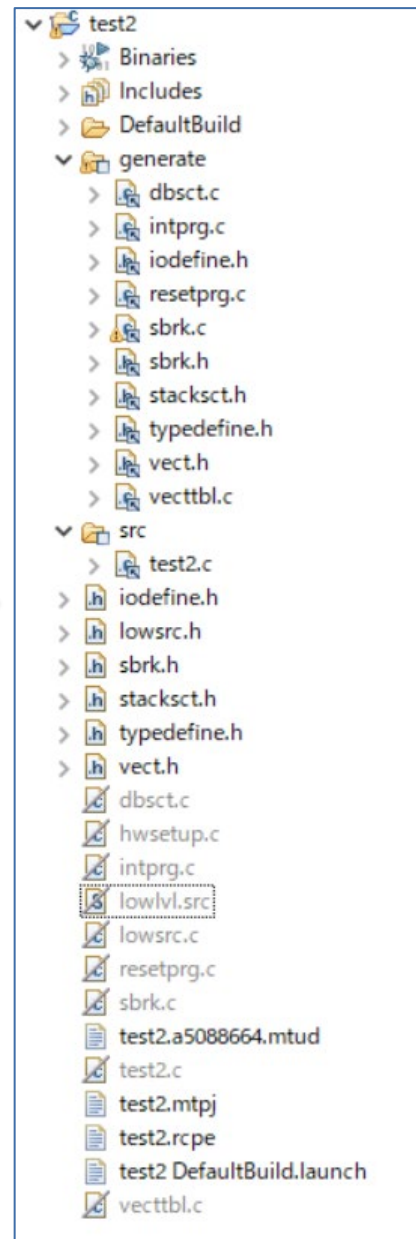
② CS+でプロジェクトを作成

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。



No.3

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 ¥resetprg.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）

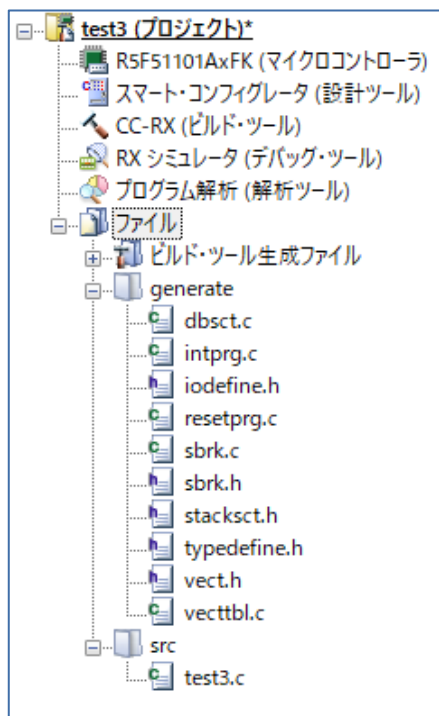
¥lowlvl.src（プロジェクトから除外）

¥lowsrc.c（プロジェクトから除外）

¥lowsrc.h（プロジェクトから除外）

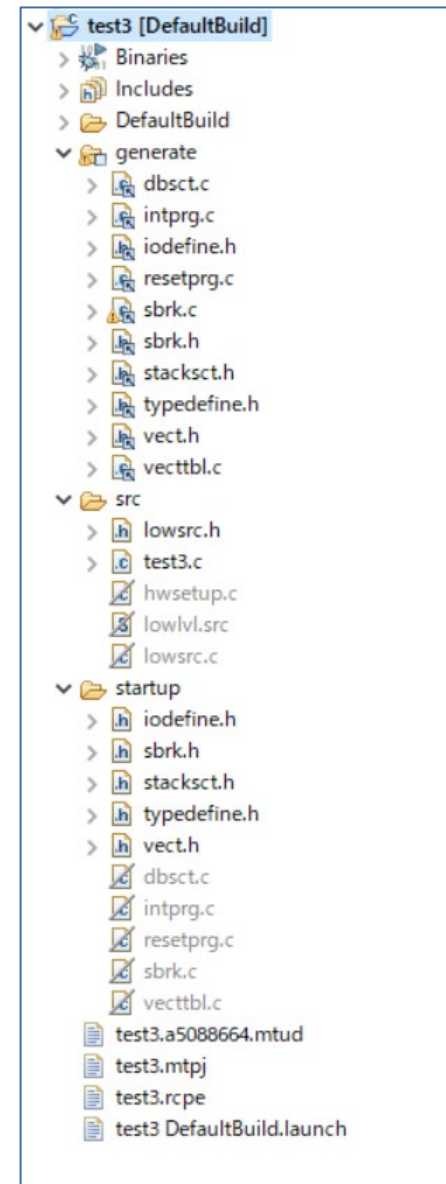
② CS+でプロジェクトを作成

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。



No.4

同名ファイルに関する制限

制限内容	e ² studio > CS+	CS+ > e ² studio
同名のソース・ファイルが複数登録されているプロジェクトを移行すると、移行先の製品でビルド・エラーを発生する場合があります。	該当	非該当

- 回避策

- プロジェクトを移行する前に移行元の製品上で、同名ソース・ファイルの片方のファイル名を重複しない名前に変更してください。

- 背景

- 製品ごとの仕様が異なるためです。
 - e² studioは、ソース・ファイルの階層構造を維持してオブジェクトを出力することで、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・ファイルのプロジェクト登録を許可しています。
 - 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しているため、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、同名ソース・ファイルのプロジェクト登録を禁止しています。

No.5

フォルダ名、フォルダパス、ファイル名に関する制限

制限内容	e ² studio > CS+	CS+ > e ² studio
フォルダ名／フォルダパスやファイル名に使用できない文字（#、\$、%、全角文字）が含まれていると、プロジェクトを移行できない場合があります。	該当	該当

- 回避策
 - プロジェクトを移行する前に移行元の製品上で、フォルダ名／フォルダパスやファイル名を使用不可文字（#、\$、%、全角文字）が含まれない名前へ変更してください。
- 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です。そのため、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。
 - e² studioは、ワークスペース・フォルダおよびプロジェクト・フォルダのパス、またはプロジェクト内のファイル名に全角文字が含まれていると正しく動作しないことがあります。

No.6

リンク対象ファイルに関する制限

制限内容	e ² studio > CS+	CS+ > e ² 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、移行先の製品でリンク対象になりません。	非該当	該当

■ 回避策

- プロジェクトを移行する前に移行元の製品上で、ライブラリ・ファイル、およびオブジェクト・ファイルをオプション設定パネルで指定してください。

■ 背景

- 製品ごとの仕様が異なるためです。
 - 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として扱いません。ライブラリ・ファイルおよびオブジェクト・ファイルは、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として扱います。オプション設定パネルで指定されたライブラリ・ファイル、およびオブジェクト・ファイルもリンク対象になります。

No.7

標準ライブラリのリンク順に関する制限

制限内容	e ² studio > CS+	CS+ > e ² studio
標準ライブラリのリンク順は製品ごとに異なります。	該当	該当

- 回避策
 - プロジェクトを移行した後に移行先の製品上で、リンク順を見直してください。
- 背景
 - 製品ごとの仕様が異なるためです。
 - e² studioは、標準ライブラリを最後にリンクします。
リンク順序のカスタマイズ機能で標準ライブラリのリンク順番を変更することができます。
 - CS+は、標準ライブラリを先頭にリンクします。

No.8

複数プロジェクトの移行に関する制限

制限内容	e ² studio > CS+	CS+ > e ² studio
複数のプロジェクトを1つにまとめてエクスポートすることはできません。	該当	非該当

- 回避策
 - 移行元の製品で1つずつプロジェクトをエクスポートして、移行先の製品で1つずつプロジェクトをインポートしてください。
- 背景
 - 製品ごとの仕様が異なるためです。
 - e² studioは、一度にエクスポートできるプロジェクトは一つです。
 - CS+は、メイン・プロジェクトと複数のサブ・プロジェクトを一度にエクスポートすることが可能です。

No.9

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	e ² studio > CS+	CS+ > e ² 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、移行先の製品で削除されます。	非該当	該当

- 回避策

- プロジェクトを移行した後に移行先の製品で、コマンドを追加してください。

- 背景

- 製品ごとの仕様が異なるためです。
 - 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。

No.10

ビルド前後の追加コマンドに関する制限

制限内容	e ² studio > CS+	CS+ > e ² studio
ビルド前後の追加コマンドに複数行のコマンドが指定されている場合、移行先の製品で2つ目以降のコマンドが削除されます。	非該当	該当

■ 回避策

- プロジェクトを移行する前に移行元の製品で、ビルド前後の追加コマンドに指定した複数行のコマンドを1行のコマンドに変更（例：複数行のコマンドを1つのバッチファイルに記述する）して指定してください。

■ 背景

- 製品ごとの仕様が異なるためです。
 - e² studioは、ビルド前後の追加コマンドに複数行のコマンドを追加する場合、区切り子として「&」を指定する必要があります。
 - CS+は、ビルド前後の追加コマンドに複数行のコマンドを追加することが可能です。

No.11

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	e ² studio > CS+	CS+ > e ² studio
ビルド前後の追加コマンドにPythonコンソールで使用可能なコマンドを指定しているプロジェクトを移行すると、移行先の製品でビルド・エラーを発生する場合があります。	該当	該当

- 回避策

- プロジェクトを移行した後に移行先の製品で、ビルド前後の追加コマンドを見直してください。

- 背景

- 製品ごとの仕様（サポートするコマンド、およびコマンドのパラメータ等）が異なるためです。

No.12

デバッグ設定に関する制限

制限内容	e ² studio > CS+	CS+ > e ² studio
デバッグの設定は、エクスポートされません。	該当	該当

- 回避策
 - プロジェクトを移行した後に移行先の製品で、デバッグの設定を追加してください。
- 背景
 - デバッグの設定情報は、プロジェクトのインポート／エクスポート機能の対象ではありません。

No.13

RTOSプロジェクトに関する制限

制限内容	e ² studio > CS+	CS+ > e ² studio
RTOSを使用したプロジェクトを移行できますが、RTOS情報は削除されます。	該当	該当

- 回避策
 - 移行先の製品でプロジェクトを新規に作成してください。
- 背景
 - RTOS情報は、プロジェクトのインポート／エクスポート機能の対象ではありません。

ビルド変数の変換

ビルド変数の変換 1

CS+からe2 studioへ移行時

移行元製品 (CS+)	意味	移行先製品 (e ² studio)
%FilePath%	ファイルの絶対パス	「\${selected_resource_loc}」に変換されます。 この値は、[プロジェクト・エクスプローラー] ビューで選択されたアイテムに依存して変わります。 e ² studio上で適切なフォルダ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}」に変換されます。 この値は、[プロジェクト・エクスプローラー] ビューで選択されたアイテムに依存して変わります。 e ² studio上で適切なフォルダ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}」に変換されます。 この値は、[プロジェクト・エクスプローラー] ビューで選択されたアイテムに依存して変わります。 e ² studio上で適切なフォルダ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}」に変換されます。 この値は、[プロジェクト・エクスプローラー] ビューで選択されたアイテムに依存して変わります。 e ² studio上で適切な名前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))」に変換されます。
%FileExt%	ファイルの拡張子	「\$(suffix \$(<F))」に変換されます。

[次ページに続く](#)

ビルド変数の変換 1

CS+からe2 studioへ移行時（続き）

移行元製品 (CS+)	意味	移行先製品 (e ² studio)
%MainProjectDir%	メイン・プロジェクトのフォルダの絶対パス	[e ² studio 2023-07およびそれ以前] 「\${ProjDirPath}」に変換されます [e ² studio 2023-10およびそれ以降] 「\${workspace_loc}/\${ProjName}」に変換されます。
%MainProjectName%	メイン・プロジェクト名	「\${ProjName}」に変換されます。
%ProjectDir%	プロジェクトのフォルダの絶対パス	[e ² studio 2023-07およびそれ以前] 「\${ProjDirPath}」に変換されます。 [e ² studio 2023-10およびそれ以降] 「\${workspace_loc}/\${ProjName}」に変換されます。
%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。
%MicomToolPath%	CS+のインストール先フォルダの絶対パス	変換されません。 e ² studio上で適切なフォルダ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。

[次ページに続く](#)

ビルド変数の変換 1

CS+からe2 studioへ移行時（続き）

移行元製品 (CS+)	意味	移行先製品 (e ² studio)
%ActiveProjectDir%	アクティブ・プロジェクトのフォルダの絶対パス	[e2 studio 2023-07およびそれ以前] 「\${ProjDirPath}」に変換されます。 [e2 studio 2023-10およびそれ以降] 「\${workspace_loc:\${ProjName}}」に変換されます。
%ActiveProjectName%	アクティブ・プロジェクト名	「\${ProjName}」に変換されます。
%MainProjectMicomName%	メイン・プロジェクトのマイクロコントローラ名	変換されません。 e2 studio上で適切な名前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	変換されません。 e2 studio上で適切な名前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	変換されません。 e2 studio上で適切な名前に書き換えてください。

ビルド変数の変換 2

e² studioからCS+へ移行時

移行元製品 (e ² studio)	意味	移行先製品 (CS+)
\${workspace_loc}	ワークスペースフォルダの絶対パス	「%ProjectFolder%¥.¥」に変換されます。
\${workspace_loc:/<path>}	ワークスペースフォルダ¥<path>の絶対パス	「%ProjectFolder%¥.¥<path>」に変換されます。
\${workspace_loc}/\${ProjName}/<path>}	プロジェクトフォルダ¥<path>の絶対パス	[e ² studio 2023-07およびそれ以前で作成されたプロジェクトの場合] 「%ProjectFolder%¥.¥%ProjectName%¥<path>」に変換されます。 [e ² studio 2023-10およびそれ以降で作成されたプロジェクトの場合] 「%ProjectFolder%¥<path>」に変換されます。
\${WorkspaceDirPath}	ワークスペースフォルダの絶対パス	「%ProjectFolder%¥.¥」に変換されます。
\${ProjDirPath}	プロジェクトフォルダの絶対パス	「%ProjectFolder%」に変換されます。
\${ProjName}	プロジェクト名	「%ProjectName%」に変換されます。
\${CONFIGDIR}	コンフィグレーションフォルダの絶対パス	「%CondifDir%」に変換されます。
\${ConfigName}	コンフィグレーション名	「%BuildModeName%」に変換されます。
\${selected_resource_loc}	選択されたリソースフォルダの絶対パス	「%FullFile%」に変換されます。
\${selected_resource_name}	選択されたリソース名	「%FileName%」に変換されます。
\${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	変換されません。 CS+上で適切なフォルダに書き換えてください。

次ページに続く

ビルド変数の変換 2

e² studioからCS+へ移行時（続き）

移行元製品 (e ² studio)	意味	移行先製品 (CS+)
\${eclipse_home}	e ² studioのインストール先フォルダの絶対パス	変換されません。 CS+上で適切なフォルダに書き換えてください。
\${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	「%TempDir%」に変換されます。
\${windir}	Windowsシステム・フォルダの絶対パス	「%WinDir%」に変換されます。
\${TCINSTALL}	ツールチェーン・インストール・フォルダの絶対パス	削除されます。 CS+上で適切なフォルダを追加してください。

[Renesas.com](https://www.renesas.com)