

# 統合開発環境 e<sup>2</sup> studio

## Jenkins を使用して e<sup>2</sup> studio のプロジェクトをビルドする方法

### はじめに

Jenkins は、継続的インテグレーション、継続的デリバリーを実現するためのオープンソースのツールです。ソフトウェアのビルド、テスト、配信、または展開に関連するあらゆる種類のタスクを自動化することができます。

このドキュメントでは、Jenkins を使用して、e<sup>2</sup> studio で作成されたプロジェクトのビルドとレポート（ビルド結果のステータス）出力を自動化する方法について説明します。

### 目次

|                                           |    |
|-------------------------------------------|----|
| 1. 概要 .....                               | 2  |
| 1.1 e <sup>2</sup> studio と Jenkins ..... | 2  |
| 1.2 動作確認環境 .....                          | 2  |
| 2. Jenkins セットアップ .....                   | 3  |
| 2.1 Jenkins をインストールする .....               | 3  |
| 2.2 Jenkins の初期設定を行う .....                | 3  |
| 2.3 ユーザーを追加する .....                       | 5  |
| 2.4 Jenkins のシステムを設定する .....              | 6  |
| 2.4.1 同時ビルド数 .....                        | 7  |
| 2.4.2 環境変数 .....                          | 7  |
| 2.5 Jenkins URL .....                     | 8  |
| 2.6 Git プラグインをインストールする .....              | 8  |
| 3. Git セットアップ .....                       | 11 |
| 3.1 Git ツールのインストール有無を確認する .....           | 11 |
| 3.2 Git ツールをインストールする .....                | 11 |
| 4. ビルド自動化 .....                           | 13 |
| 4.1 ビルドジョブ作成 .....                        | 13 |
| 4.2 ビルドジョブ実行 .....                        | 15 |
| 5. 参考情報 .....                             | 18 |
| 5.1 Web サイト .....                         | 18 |
| 改訂記録 .....                                | 19 |

## 1. 概要

### 1.1 e<sup>2</sup> studio と Jenkins

e<sup>2</sup> studio は、Eclipse をベースとした統合開発環境です。様々なオープンソースソフトウェアのプラグインを組み込んで、機能を追加／拡張することができます。

Jenkins は、継続的インテグレーション（CI）、および継続的デリバリー（CD）を実現するためのツールです。「ソフトウェアのリリーススピードの向上」、「開発プロセスの自動化」、「開発コストの削減」といった利用目的に適しています。CI/CD ツールはたくさんありますが、その中でも Jenkins は、汎用性が高いことからユーザーから大きな支持を得ています。ビルドやテストに限らずどんなスクリプトでも実行できるので、ビルドプロセスに合わせてビルドからリリースまで柔軟に自動化を行うことができます。

このドキュメントでは、Jenkins を使用して、e<sup>2</sup> studio で作成したプロジェクトのビルド自動化、およびレポート（ビルド結果のステータス）を確認する方法について説明します。

### 1.2 動作確認環境

このドキュメントで説明する操作手順については、弊社にて以下の環境で確認を実施しています。ただし、オープンソースのソフトウェアとの連携になりますので、弊社が動作を保証するものではありません。あらかじめご了解の程お願い申し上げます。

#### [OS]

- OS Windows10（日本語版）

#### [ツール]

- e<sup>2</sup> studio 2022-01
  - EGit 6.1.0.202203080745-r
- Jenkins 2.332.2
  - Git plugin 4.11.0
- Git 2.35.1

#### [プロジェクト]

例として以下のデバイス、およびツールチェーンを指定したプロジェクトを使用します。

- デバイス RX64M
- ツールチェーン GCC for Renesas RX 8.3.0.202104-GNURX-ELF

## 2. Jenkins セットアップ

この章では、Jenkins のインストールからセットアップ、および Git プラグインのインストール手順について説明します。

### 2.1 Jenkins をインストールする

Jenkins のインストーラは、以下のページからダウンロードページにジャンプしてダウンロードしてください。

<https://www.jenkins.io/>

お客様の環境に応じて正しいパッケージをダウンロードし、インストールしてください。このドキュメントでは、Windows10 環境を対象とします。

Jenkins をインストールする前に Java をインストールしておく必要があります。

- 1) 上記 URL にアクセスします。Windows を選択して、「jenkins.msi」ファイルをダウンロードします。

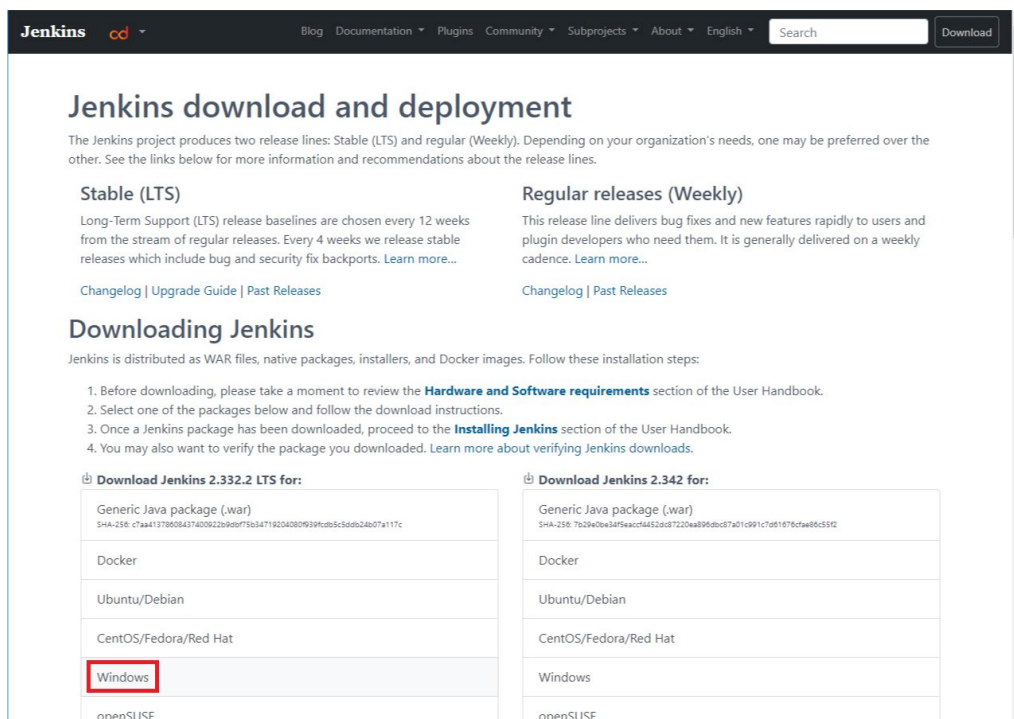


図 1 ダウンロードページ

- 2) ダウンロードした「jenkins.msi」ファイルを実行します。
- 3) インストーラが起動しますので、ウィザードに従ってインストールを完了します。

### 2.2 Jenkins の初期設定を行う

インストールが正常に終了したら、「<http://localhost:8080>」にアクセスしてください。Jenkins が起動します。

- 1) はじめて Jenkins が起動されると [Unlock Jenkins] ページが表示されます。管理者の初期パスワードを求められますので、画面中段に記載されている赤字のパスワードファイルを参照して、[Administrator password] 編集ボックスに入力します。続けて [Continue] ボタンをクリックします。

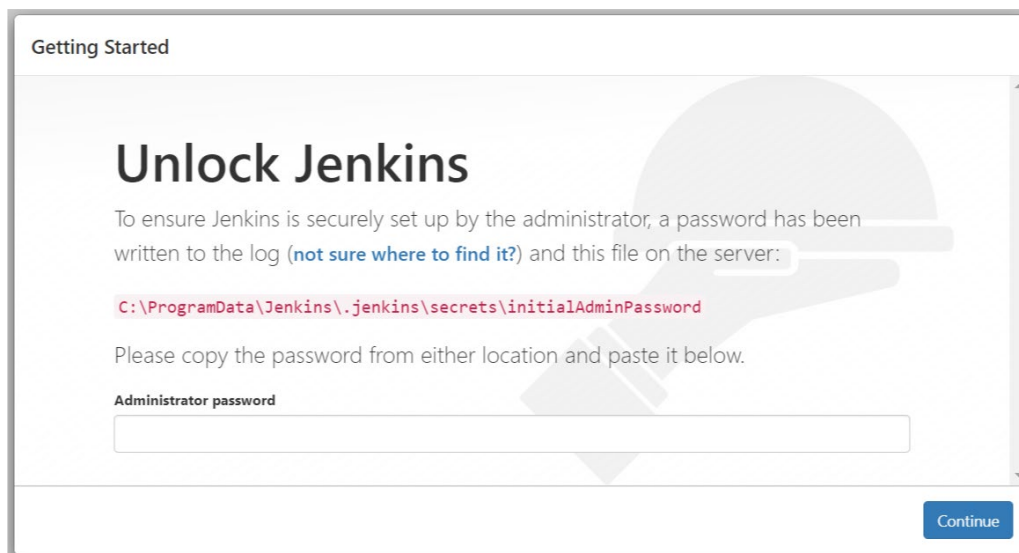


図 2 【Unlock Jenkins】 ページ

- 2) 【Customize Jenkins】 ページが表示されます。プラグインの設定を求められますが、後でプラグインの設定は行えますので、ここでは [X] ボタンをクリックします。

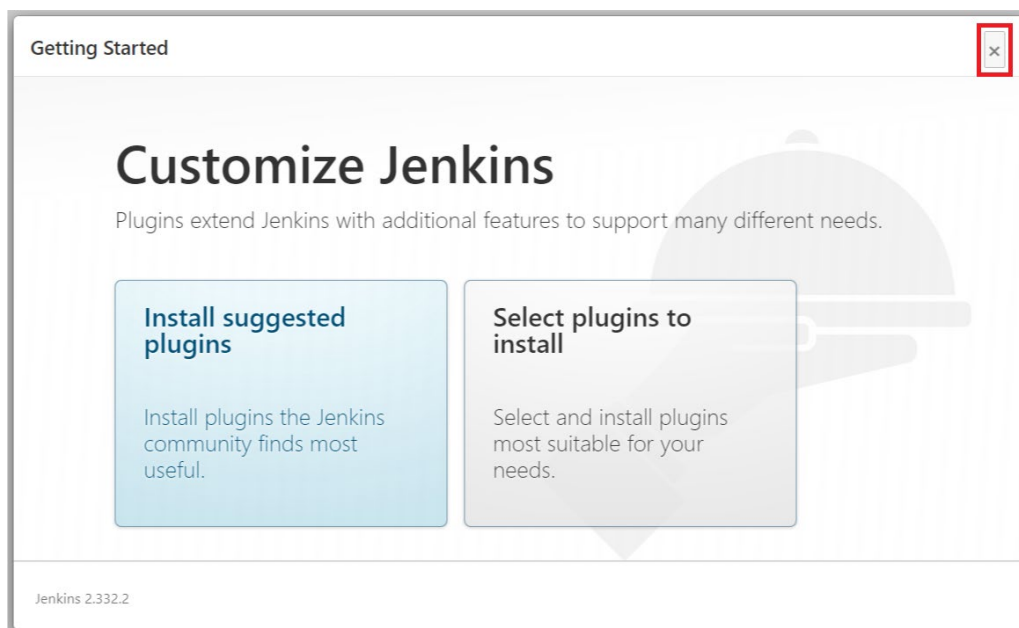


図 3 【Customize Jenkins】 ページ

- 3) 【Jenkins is ready!】 ページが表示されます。【Start using Jenkins】 ボタンをクリックします。

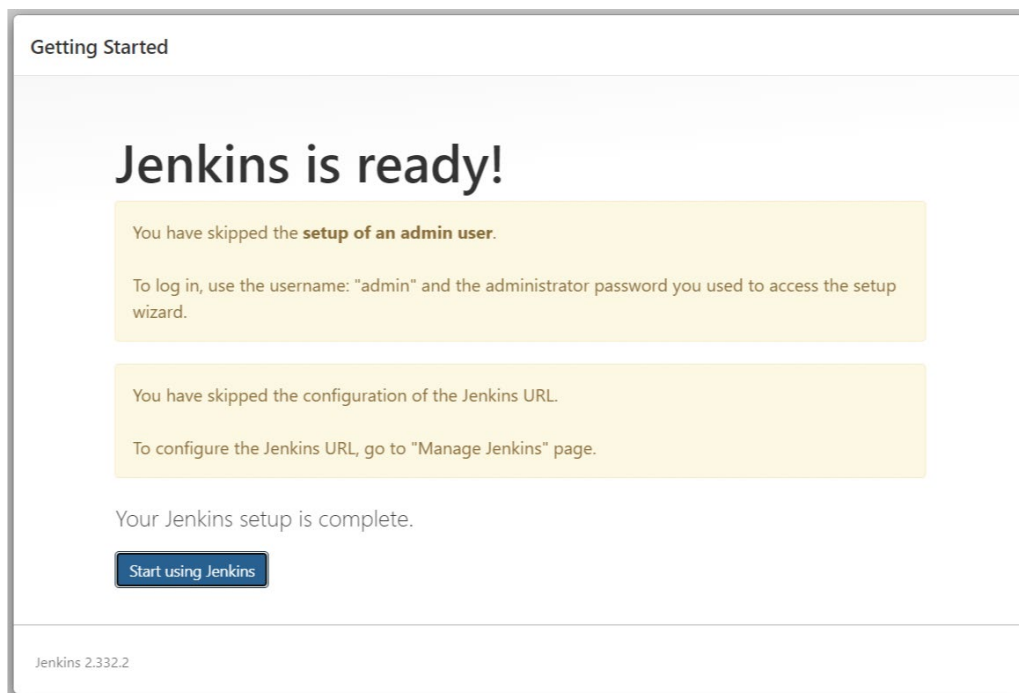


図 4 [Jenkins is ready!] ページ

4) これで Jenkins の初期設定は完了です。

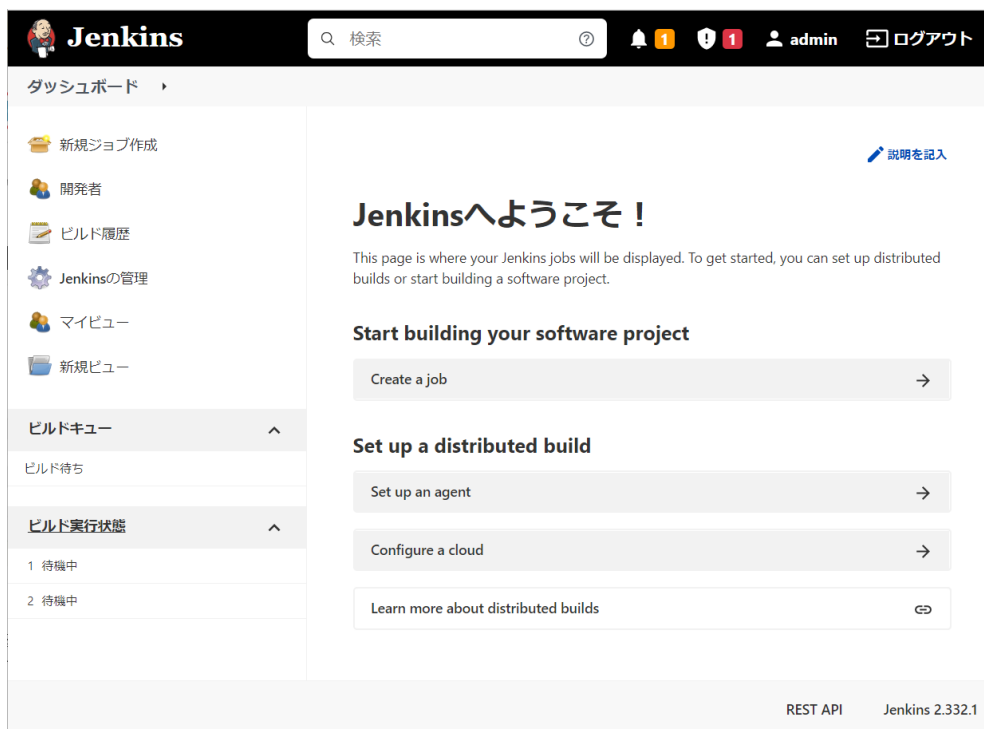


図 5 [Jenkins] ページ

## 2.3 ユーザーを追加する

Jenkins は、デフォルトで 1 人の管理者ユーザーが登録されています。他のユーザーアカウントを追加する場合は、下記手順を実施してください。

1) [Jenkins] ページで [Jenkins の管理] をクリックし、次に [ユーザーの管理] をクリックします。

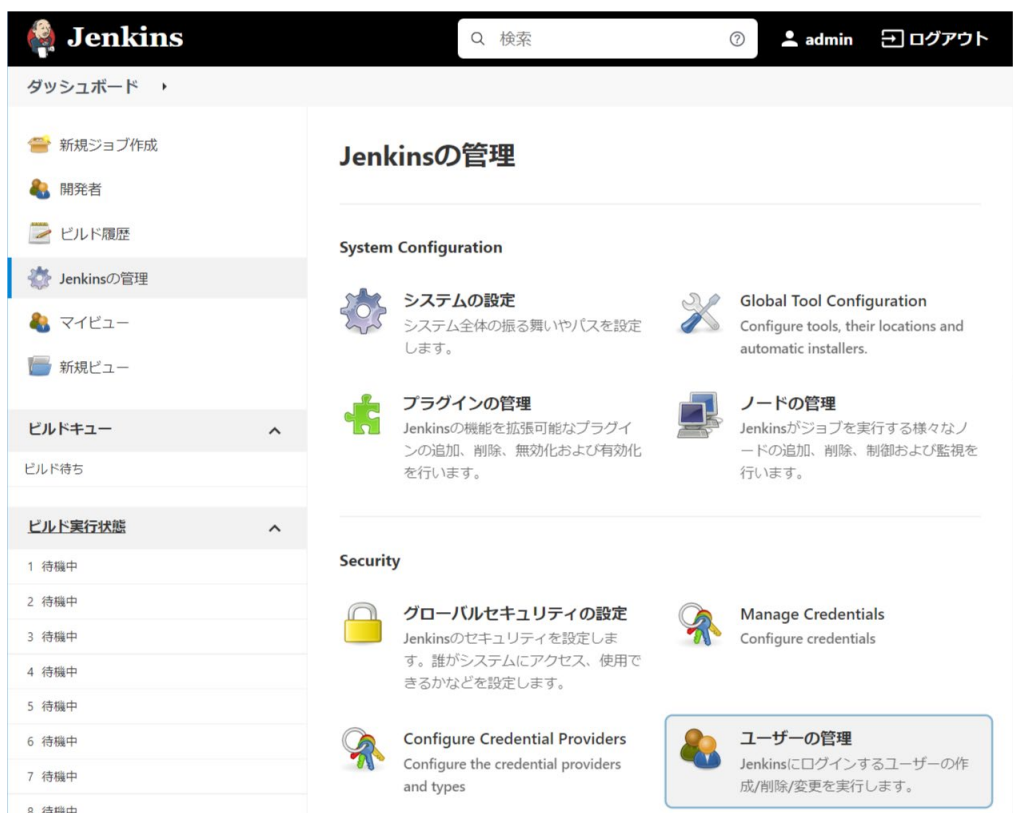


図 6 [Jenkins の管理] ページ

- 2) [Jenkins のユーザーデータベース] ページが表示されます。[ユーザー作成] をクリックします。
  - 3) [ユーザー作成] ページが表示されます。[ユーザー名:]、[パスワード:]、[パスワードの確認:]、[フルネーム:] 編集ボックスに情報を入力し、[ユーザー作成] ボタンをクリックします。
- ユーザーアカウントが作成され、作成されたユーザーで Jenkins にログインできます。

[Jenkins のユーザーデータベース] ページでは、 をクリックすると、ログインパスワードなどのユーザー設定変更、およびその他の設定を変更することができます。

## 2.4 Jenkins のシステムを設定する

[Jenkins の管理] ページの [システムの設定] において、Jenkins のシステム全体の振る舞いやパスを設定することができます。



図 7 [Jenkins の管理] ページ

### 2.4.1 同時ビルド数

同時ビルド数とは、Jenkins で同時に実行できるジョブの総数になります。パフォーマンスをよりよく保つため、この数値はマシンの CPU の数と同じにすることを推奨します。

同時ビルド数

図 8 同時ビルド数

### 2.4.2 環境変数

環境変数は、すべてのジョブに適用されるカスタム環境変数を追加するために使用します。環境変数の設定は、キーと値のペアであり、すべてのジョブのすべてのビルドに適用されます。ビルドから起動するプロセスにも環境変数として追加されます。

#### グローバルプロパティ

☐ ツールパス

☒ 環境変数

キーと値のリスト ?

キー

値

削除

追加

図 9 環境変数の追加

## 2.5 Jenkins URL

Jenkins をインストールしたマシンで Jenkins にアクセスするための URL は、「<http://localhost:8080/>」です。Jenkins をサーバーにインストールした場合は、サーバーのドメイン名または IP アドレスを使用してアクセスします。

環境変数に JENKINS\_URL として上記 URL を設定しておく、Jenkins が送信する電子メールで、\${JENKINS\_URL} と記載すると、環境変数 JENKINS\_URL の値に自動的に置換されます。

## 2.6 Git プラグインをインストールする

Git プラグインは、Jenkins が Git リポジトリにアクセスしてソースコードを取得するために使用します。下記手順に従って、Git プラグインをインストールしてください（インターネット接続が必要です）。

- 1) [Jenkins] ページで、[Jenkins の管理] をクリックします。次に [プラグインの管理] をクリックします。



図 10 [Jenkins の管理] ページ

- 2) [プラグインマネージャー] ページが表示されます。[利用可能] ボタンをクリックして、[フィルター:] 編集ボックスに「Git」を入力します。ページ内容が、「Git」でフィルタリングされます。



図 11 [プラグインマネージャー] ページ



- 3) [Git] をチェックして、[Install without restart] ボタンをクリックします。インストールが開始されます。
- 4) インストールが完了したら、[インストール完了後、ジョブがなければ Jenkins を再起動する] をチェックします。Jenkins が自動的に再起動します。

[ページの先頭へ戻る](#)  
(すぐにインストールしたプラグインを使用できます)

☒ インストール完了後、ジョブがなければ Jenkins を再起動する

図 12 [インストール完了後、ジョブがなければ Jenkins を再起動する]

- 5) Jenkins にログインします。  
(ユーザーを追加していない場合は、ユーザー名 : Admin、パスワード : initialAdminPassword ファイルの内容、でログインできます)

ここから先の操作を実施する前に次章の説明に従って Git を先にインストールしてください。

- 6) [Jenkins] ページで、[Jenkins の管理] をクリックします。次に [Global Tool Configuration] をクリックします。



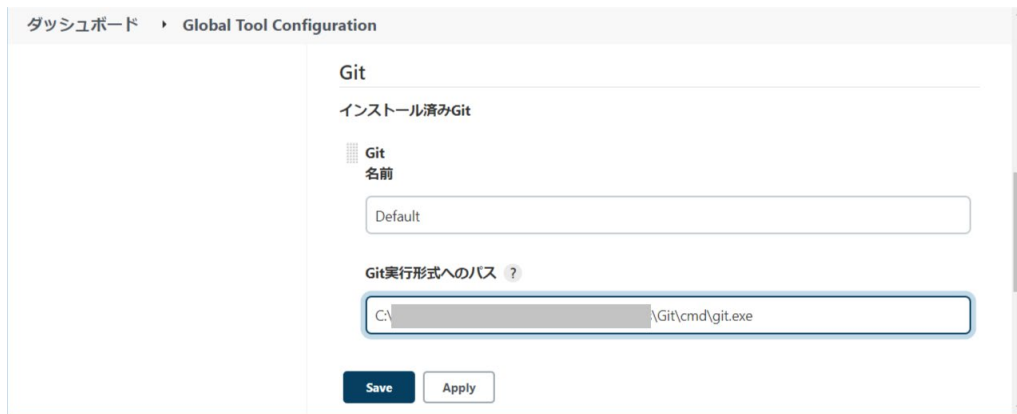
図 13 [Jenkins の管理] ページ

- 7) [Global Tool Configuration] ページが表示されます。Git の設定を確認します。下記エラーが発生している場合は、[Git 実行形式へのパス] に「git.exe」のパスを指定します。



図 14 「There's no such executable git.exe in PATH」エラー

- 8) 正しいパスを指定するとエラーが消えます。



9) [Save] ボタンをクリックします。

これで Jenkins から Git が使用可能になりました。

### 3. Git セットアップ

前の章の最後に、Jenkins に Git を認識させるために Git プラグインをインストールしました。Jenkins が使用する Git ツールは、Jenkins サーバーを実行するホストにインストールする必要があります。Git が既にインストールされているかどうか分からない場合は、次の手順で確認することができます。

#### 3.1 Git ツールのインストール有無を確認する

- 1) Windows のコマンドプロンプト画面で次のコマンドを入力します。

```
> git --version
```

- 2) Git ツールが既にインストールされている場合、バージョン番号が表示されます。例を示します。

```
git version 2.35.1.windows.2
```

- 3) Git ツールがインストールされていない場合、次の応答が表示されます。

```
'git' は、内部コマンドまたは外部コマンド、  
操作可能なプログラムまたはバッチ ファイルとして認識されていません
```

#### 3.2 Git ツールをインストールする

Git ツールがインストールされていない場合は、下記手順に従って、Git ツールをインストールしてください。

- 1) 下記ページからダウンロードページにジャンプして、Windows 版 Git のインストーラをダウンロードします。

<https://git-scm.com/>

- 2) インストーラを実行します。

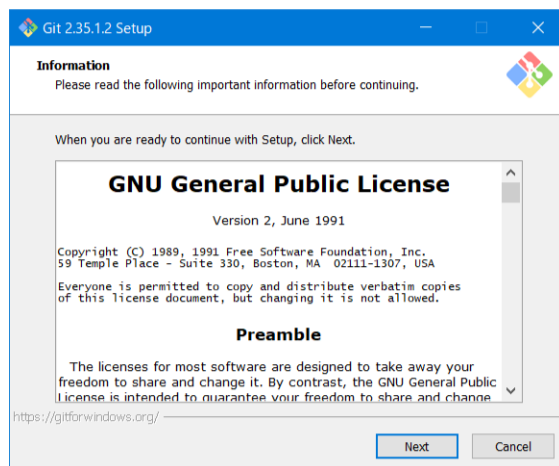


図 15 インストール画面 1

- 3) 画面の指示に従い [Next >] ボタンをクリックして [Adjusting your PATH environment] ページまで進めます。このページでは、[Git from the command line and also from 3rd-party software] チェックボックスをチェックし、[Next >] ボタンをクリックします。

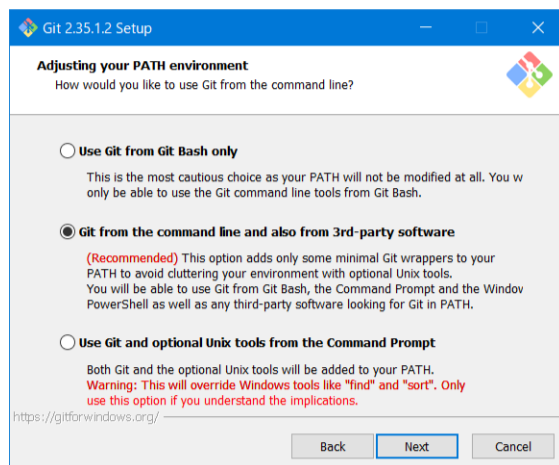


図 16 インストール画面 2

4) ページを進めて、インストールを完了します。

これで Git のインストールは完了です。「3.1 Git ツールのインストール有無を確認する」の説明に従って、Git が使用可能であるか確認してください。

## 4. ビルド自動化

この章では、e<sup>2</sup> studio で作成したプロジェクトを使用して、Git のリモートリポジトリにプロジェクトのソースファイルがプッシュされたら、e<sup>2</sup> studio のヘッドレス・ビルドを実行するジョブを作成する方法について説明します。

### 4.1 ビルドジョブ作成

下記手順に従って、ビルドジョブを作成してください。

- 1) [Jenkins] ページで、[新規ジョブ作成] をクリックします。
- 2) [Enter an item name] 編集ボックスに「e2Build」を入力して、[フリースタイル・プロジェクトのビルド] をクリックします。次に、[OK] ボタンをクリックします。



図 17 新規ジョブ作成

- 3) ジョブの詳細設定を行うための画面が表示されます。



図 18 [General] タブ

- 4) [ソースコード管理] タブをクリックします。[ソースコード管理] 設定が表示されますので、[Git] をチェックし、[リポジトリ URL] 編集ボックスに Git リポジトリのディレクトリを指定します (Git リポジトリにプロジェクトが登録されている必要があります)。



図 19 【ソースコード管理】タブ

- 5) 必要に応じて、認証情報を追加してください。
- 6) 【ビルド・トリガ】タブをクリックします。【ビルド・トリガ】画面が表示されますので、ビルドの実行タイミングを設定します。プロジェクトが Git リポジトリで更新されるたびにビルドを実行するには、【SCM をポーリング】をチェックします。これにより、Git リポジトリの更新が定期的にチェックされ、更新がある場合、ビルドが実行されます。
- 7) Git リポジトリの更新を確認する間隔は、cron 風の構文で指定します。「H/15 \*\*\*\*」を入力します。これは、15 分ごとに Git リポジトリをチェックすることを意味します。詳細については、【スケジュール】編集ボックス右横の【?】ボタンをクリックしてください。



図 20 【ビルド・トリガ】タブ

- 8) 【ビルド】タブをクリックします。【ビルド】画面が表示されますので、【ビルド手順の追加】コンボボックスで、「Windows バッチコマンドの実行」を選択します。

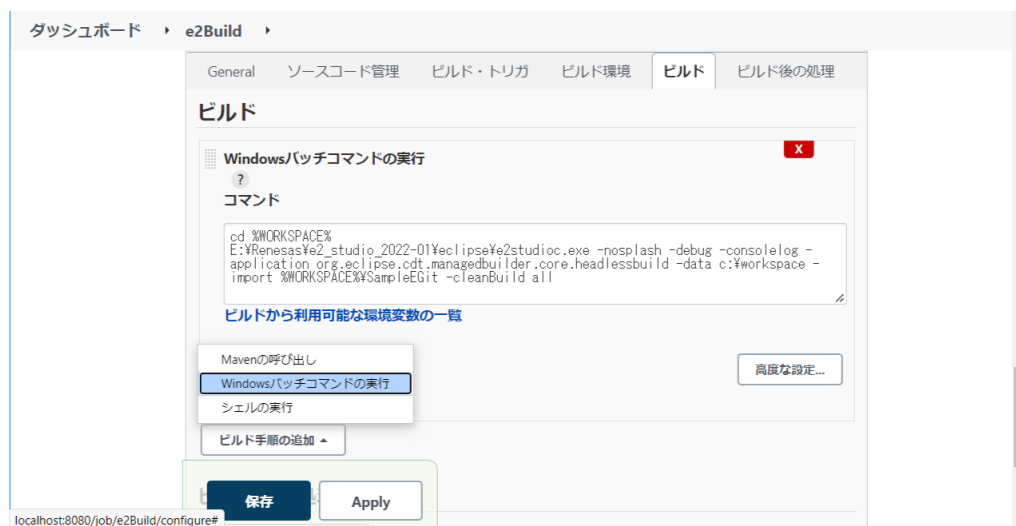


図 21 「ビルド」タブ

- 9) 「コマンド」編集ボックスに、e<sup>2</sup> studio のヘッドレス・ビルド実行のための下記コマンドを入力します。  
(ヘッドレス・ビルドとは、e<sup>2</sup> studio UI を使用せずにコマンド・ラインでプロジェクトをビルドすることです。コマンド・ライン・オプションの詳細は、e<sup>2</sup> studio ヘッドレス・ビルド機能のヘルプをご参照ください。)

```
cd %WORKSPACE%
<e2 studio install folder>%eclipse%e2studioc.exe -nosplash -debug -
consolelog -application org.eclipse.cdt.managedbuilder.core.headlessbuild
-data <workspace folder> -import <project name> -cleanBuild all
```

- <e2 studio install folder> : e2 studio のインストールディレクトリを指定してください。
- e2studioc.exe : e2 studio 2021-10 およびそれ以前のバージョンでは、eclipse.exe を指定してください。
- <workspace folder> : ヘッドレス・ビルドでは、e2 studio UI は使用しませんが、通常と同じようにワークスペースの指定が必要になります。使用するプロジェクトが含まれていないワークスペース・フォルダを指定してください。
- <project\_name> : Git リポジトリに登録されているプロジェクト名を指定してください。ここでは、「SampleEGit」を指定します。

- 10) 「保存」ボタンをクリックします。保存されると、「プロジェクト e2Build」ページが表示されます。

## 4.2 ビルドジョブ実行

下記手順に従って、作成したビルドジョブの実行を確認します。

- 1) 「Jenkins」ページで、「e2Build」をクリックします。
- 2) 「プロジェクト e2Build」ページが表示されますので、「ビルド実行」をクリックします。すぐにビルドが開始されます（これはビルドのテストを目的とした実行です。通常は、ポーリングにより、新しいソースコードが Git リポジトリにコミットされたことが検出されると、自動的にビルドが実行されます）。



図 22 ビルド実行

- 3) ビルドが完了すると、ビルドのステータスが表示されます。ビルド履歴で、#番号をクリックすると、ビルドの詳細が表示されます。

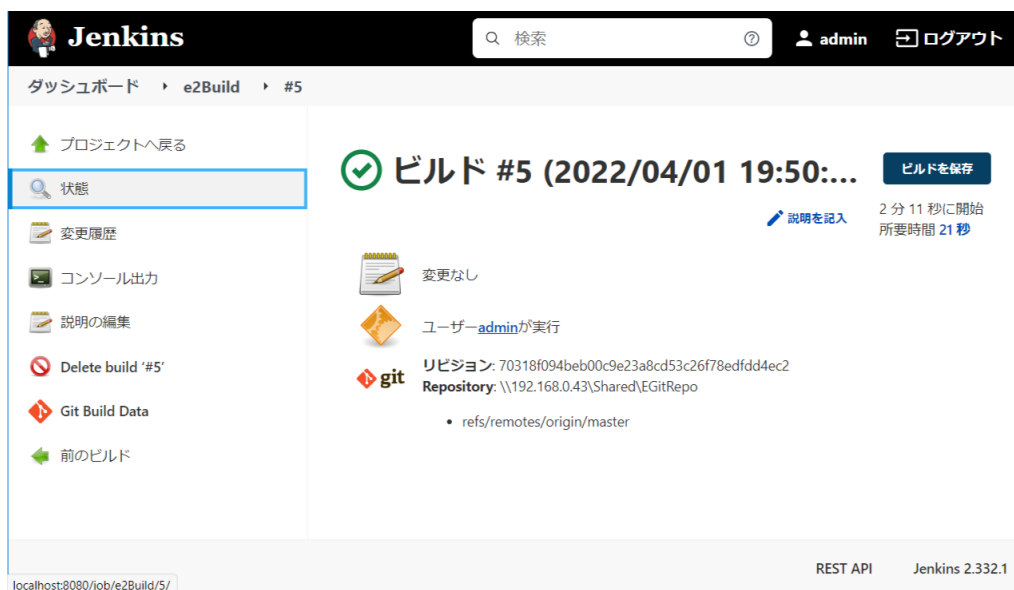


図 23 ビルドステータス

- 4) 「コンソール出力」をクリックすると、ビルド時のコンソールの出力内容が表示されます。



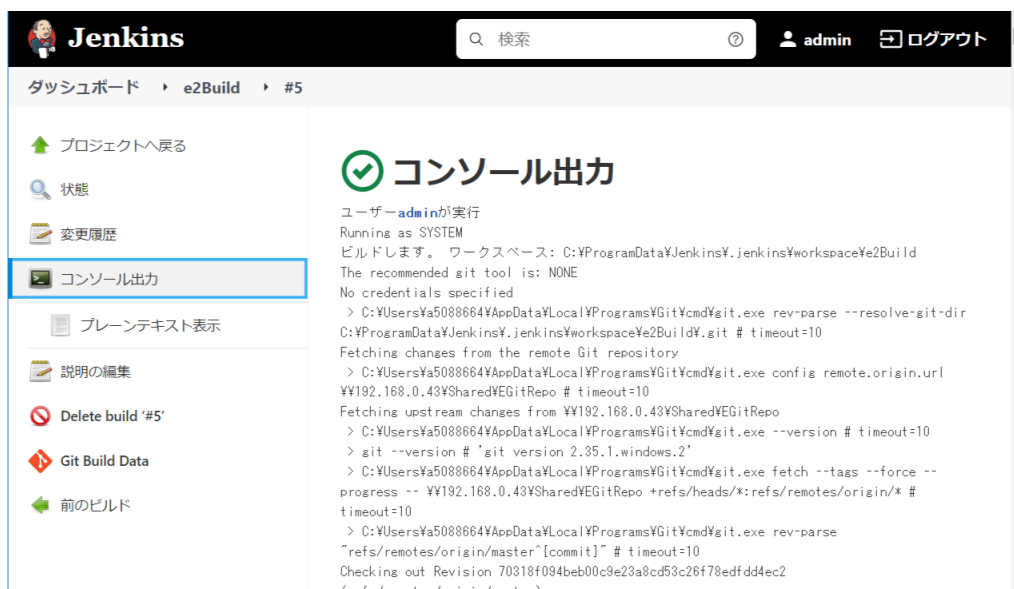


図 24 コンソール出力

- 5) e<sup>2</sup> studio でプロジェクトのソースファイルを更新して、リポジトリに Commit/Push すると自動的にビルドが実行されます。

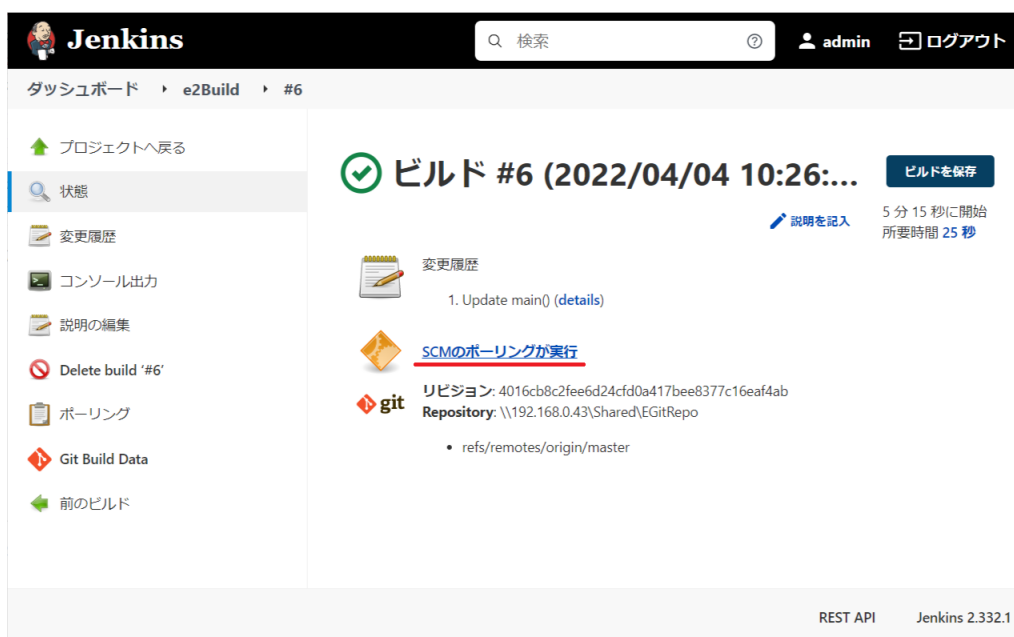


図 25 Git リポジトリ更新後のビルド結果

## 5. 参考情報

### 5.1 Web サイト

- e<sup>2</sup> studio  
<https://www.renesas.com/software-tool/e-studio>
- Jenkins  
<https://jenkins.io/>
- Git  
<https://git-scm.com>

改訂記録

| Rev. | 発行日       | 改訂内容 |                                                                             |
|------|-----------|------|-----------------------------------------------------------------------------|
|      |           | ページ  | ポイント                                                                        |
| 1.01 | Apr.20.22 | すべて  | 「R20AN0524EE0100 How to use Jenkins to automate build and report」をベースに全面見直し |
|      |           |      |                                                                             |

## 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

### 1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

### 2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

### 3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力ブルアップ電源を入れないでください。入力信号や入出力ブルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

### 4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

### 5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後、に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 $V_{IL}$  (Max.) から  $V_{IH}$  (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 $V_{IL}$  (Max.) から  $V_{IH}$  (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

### 7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違くと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

## ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等

高品質水準： 輸送機器（自動車、電車、船舶等）、交通管制（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。

6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものいたします。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.4.0-1 2017.11)

## 本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

[www.renesas.com](http://www.renesas.com)

## お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

[www.renesas.com/contact/](http://www.renesas.com/contact/)

## 商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。