

# Integrated Development Environment e<sup>2</sup> studio

## How to use EGit in e<sup>2</sup> studio

### Introduction

EGit is an Eclipse Team provider for the Git version control system. Git is a distributed Source Control Management (SCM), which means every developer has a full copy of all history of every revision of the code, making queries against the history very fast and versatile.

This application note guides user to use EGit in Renesas e<sup>2</sup> studio environment.

### Contents

1. Overview.....	2
1.1 Purpose .....	2
1.2 Operating Environment .....	2
1.3 EGit references.....	2
2. Setup .....	3
2.1 Installing EGit .....	3
2.2 Specifying user identification.....	4
3. Create repository .....	6
3.1 Create a "master" repository .....	6
3.2 Cloning a repository .....	7
4. Version control of a project.....	10
4.1 Adding a project to Git repository .....	10
4.2 Committing a change .....	13
4.3 Reviewing a change .....	14
4.4 Excluding a file from version control.....	14
5. Synchronizing works among users.....	16
5.1 Rebasing from remote repository.....	16
Revision History .....	20

## 1. Overview

### 1.1 Purpose

This document guides user to use EGit for source code version control in e<sup>2</sup> studio.

### 1.2 Operating Environment

Renesas have confirmed the operation procedure explained in this document in the following environment.

OS	Windows10 x64
e <sup>2</sup> studio	2021-07
EGit	5.12.0.202106070339-r
Target devices	RX64M
Toolchain	GCC for Renesas RX C/C++ Toolchain v 8.3.0.202102

### 1.3 EGit references

Further information about EGit can be referred in <http://www.eclipse.org/egit/documentation/>

EGit can be downloaded from following link:

- <http://download.eclipse.org/egit/updates> (main update site)

## 2. Setup

This section shows how to install EGit to e<sup>2</sup> studio.

To integrate EGit into e<sup>2</sup> studio environment, please specify the following option at the installer when installing e<sup>2</sup> studio.

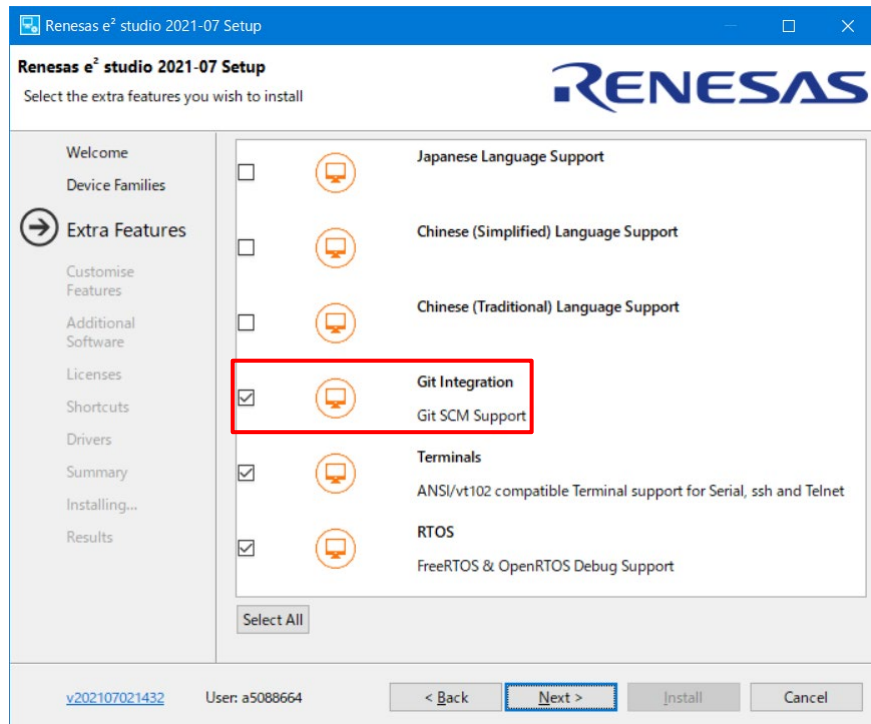


Figure 1 [Extra Features] page

### 2.1 Installing EGit

EGit have already installed int e<sup>2</sup> studio as the above step. If you will update EGit in e<sup>2</sup> studio, please follow below steps:

- 1) Invoke e<sup>2</sup> studio and select [Help] > [Install New Software...]
- 2) In the [Install - Available Software] dialog, type download link into [Work with:] textbox (e.g. "<http://download.eclipse.org/egit/updates>"), then click [Add...] button.
- 3) In the [Add Repository] dialog, type "EGit updates" (or any other meaningful name) into [Name:] textbox, then click [Add] button.

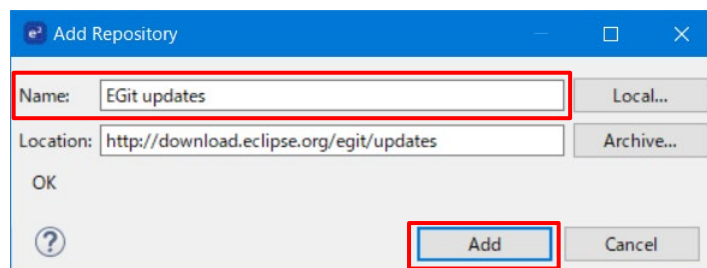
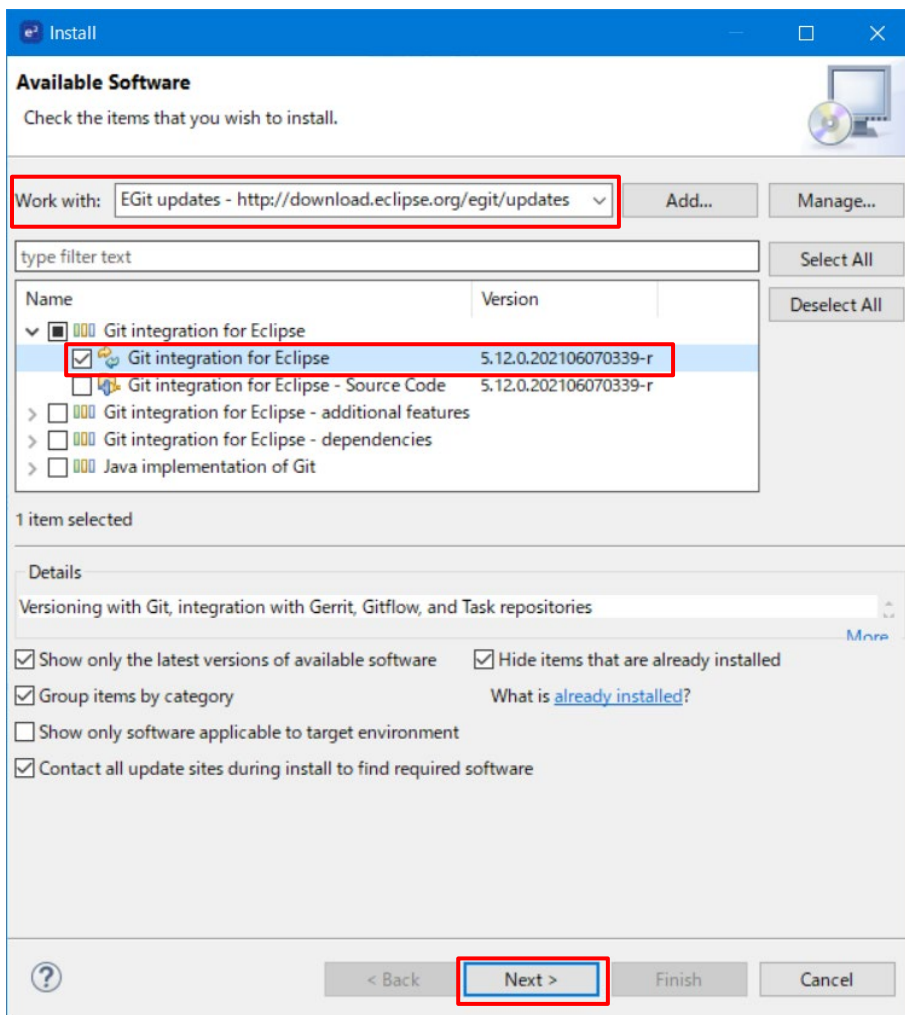


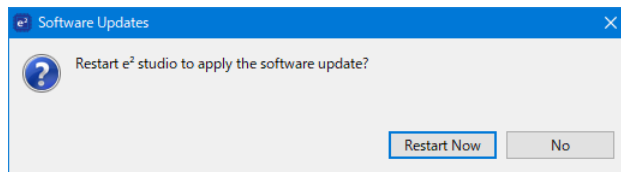
Figure 2 [Add Repository] dialog

- 4) List of available software will be shown, expand "Git integration for Eclipse" and select "Git integration for Eclipse". Then click [Next >] button, and click [Next >] button again in [Install] dialog.



**Figure 3 [Install - Available Software] dialog**

- 5) Select "I accept the terms of license agreement" in [Install - Review Licenses] dialog. Click [Finish] button. Then wait for EGit to be installed.
- 6) Once installation is completed, the following message is shown as a confirmation to restart e<sup>2</sup> studio. Click [Restart Now] button.



**Figure 4 [Software Updates] message**

- 7) Then user can open EGit perspective by clicking [Window] > [Perspective] > [Open Perspective] > [Others...], then select "Git".

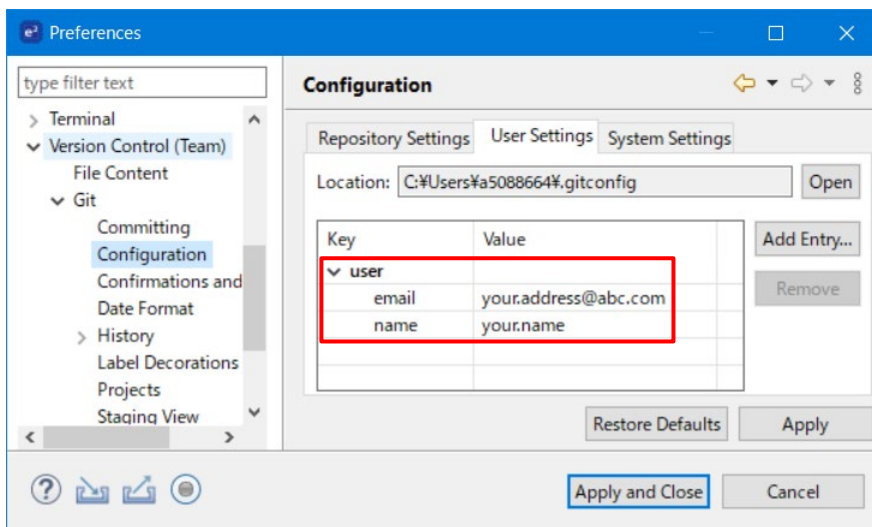
## 2.2 Specifying user identification

Whenever the history of the repository is changed (technically, whenever a commit is created), Git keeps track of the user who created that commit. The identification consists of a name (typically a person's name) and an e-mail address. This information is stored in file "<user-home-folder>\.gitconfig" under dedicated keys.

User can edit this information using the Git configuration by following these steps:

- 1) Click [Window] > [Preferences]

- 2) In the [Preferences] dialog, expand [Version Control (Team)] > [Git], select [Configuration].
- 3) Select [User Settings] tab and click [Add Entry...] button.
- 4) In the [Add a configuration entry] dialog, input following information:
  - [Key:] "user.name"
  - [Value:] input user's nameThen click [Add] button.
- 5) Click [Add Entry...] again to add email address:
  - [Key:] "user.email"
  - [Value:] input user's email addressThen click [Add] button.



**Figure 5 [Preferences - Configuration] dialog**

- 6) Click [Apply and Close] button to store these information to .gitconfig file.

### 3. Create repository

This section shows how to create a new Git repository from a Client machine.

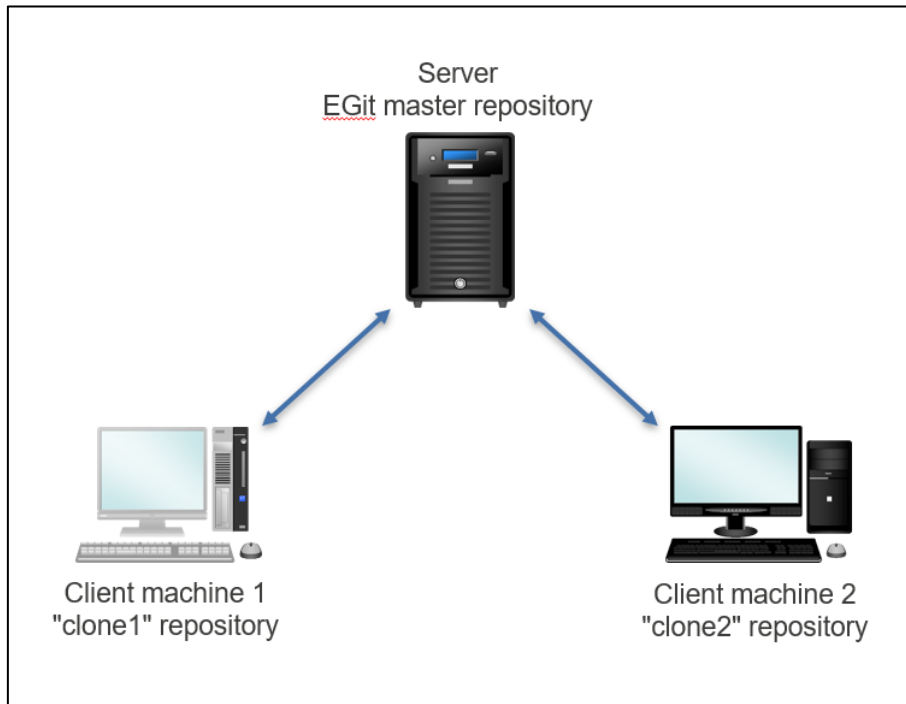


Figure 6 Network architecture for EGit repository

#### 3.1 Create a "master" repository

To use the repository as a "master" repository, it should be put on a network folder which is accessible from other "client" machine, or user can use other online services such as GitHub, Amazon CodeCommit... etc. Below instructions are applied for repository put in a local network folder.

- 1) From a "client" machine which has e<sup>2</sup> studio and EGit installed, launch e<sup>2</sup> studio, then click [Window] > [Perspective] > [Open Perspective] > [Others...], then select "Git".
- 2) In the [Git Repositories] view, click [Create a new Git repository and add it to this view] button.

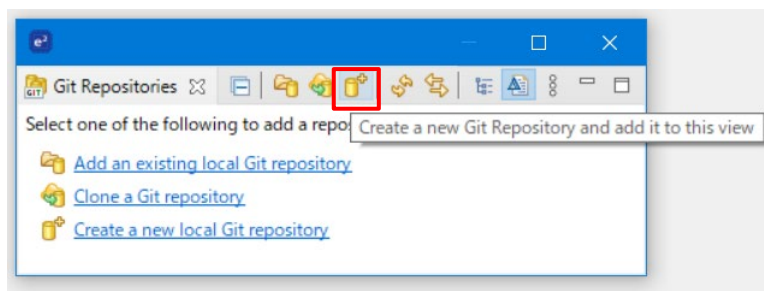
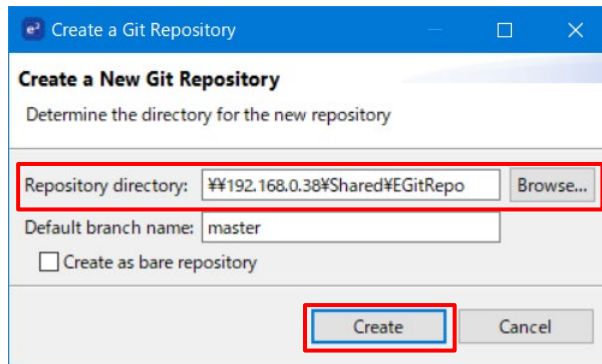


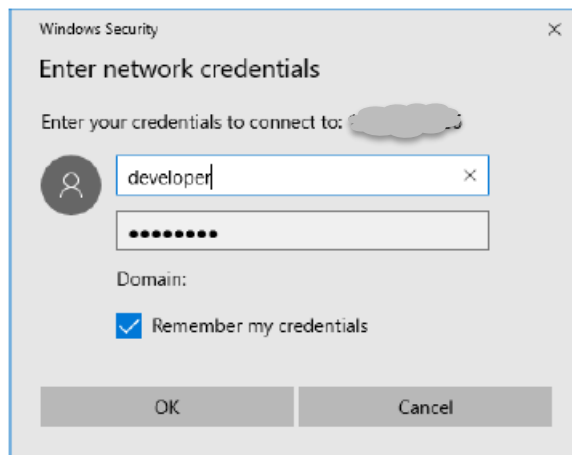
Figure 7 [Create a new Git repository and add it to this view] button

- 3) In the [Create a Git Repository] dialog, browse to the network folder that you would like to create the repository, for example "\\192.168.0.38\Shared\EGitRepo", then click [Create] button. If you would like not to code in this repository (just use it for version control, not to put the source code itself), click "Create as bare repository".



**Figure 8 [Create a Git Repository] dialog**

- 4) User might be asked for login credential to access the folder. Input username and password, check [Remember my credentials] checkbox then click [OK] button.

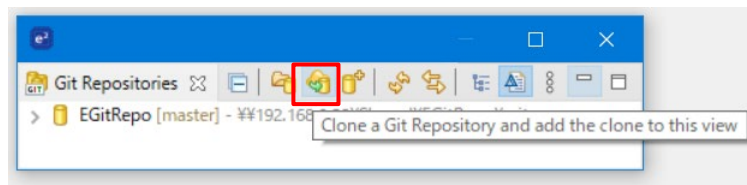


**Figure 9 Enter network credentials**

### 3.2 Cloning a repository

This section shows how to clone an existing repository and add a new project into version control.

- 1) Switch to "Git" perspective, click [Clone a Git Repository and add the clone to this view] button in the [Git Repositories] view.



**Figure 10 [Clone a Git Repository and add the clone to this view] button**

- 2) In the [Clone Git Repository - Source Git Repository] dialog, input the location of the "master" repository created in section 3.1. Then click [Next >] button.

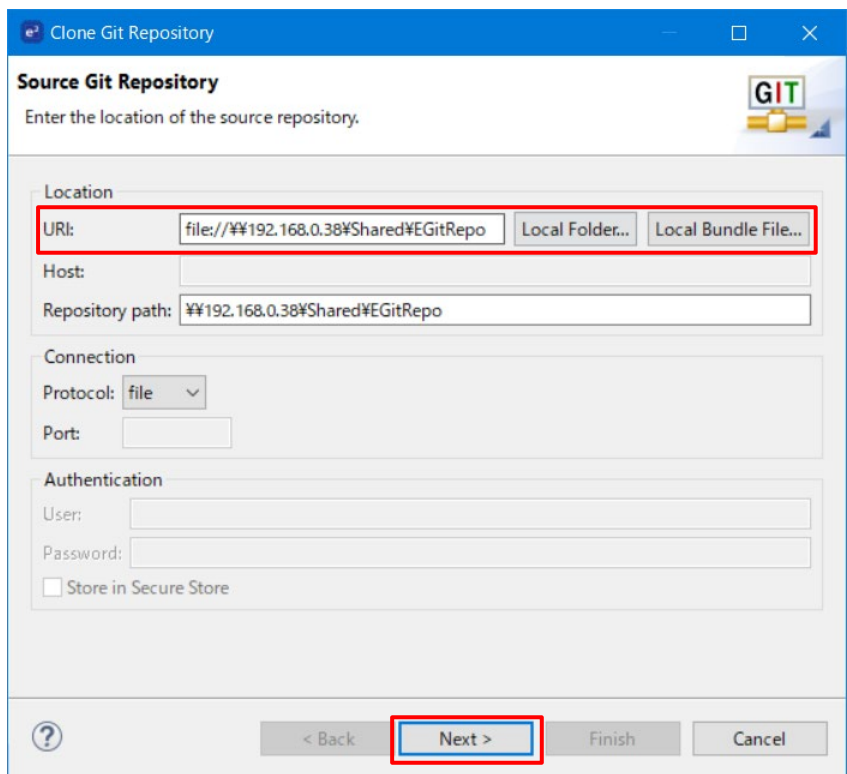


Figure 11 [Clone Git Repository - Source Git Repository] dialog

- 3) In the [Clone Git Repository - Branch Selection] dialog, if the repository is newly created, there will be no branches, just click [Next >] button. Otherwise, you can select the branch before clicking [Next >] button.

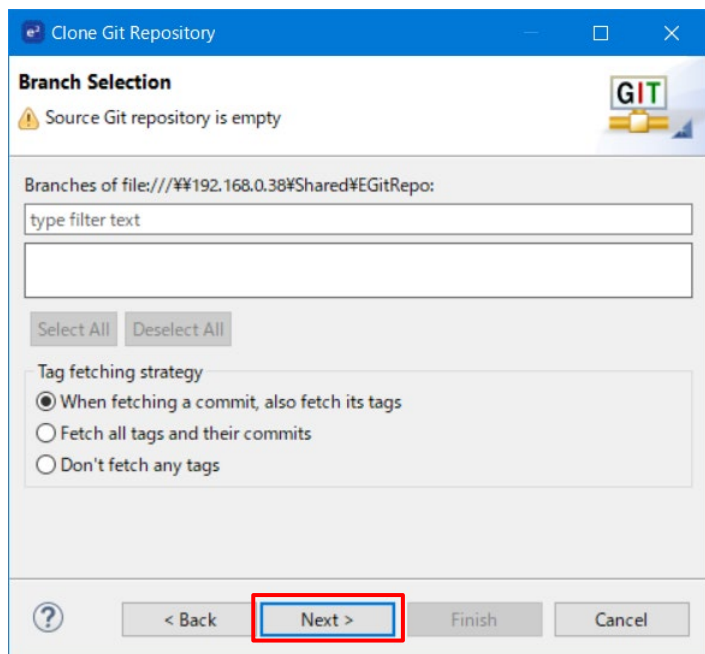


Figure 12 [Clone Git Repository - Branch Selection] dialog

- 4) In the [Clone Git Repository - Local Destination] dialog, select the folder in "client" machine to put the cloned repository, for example "C:\workspace\EGitClone\clone1". Then click [Finish] button.



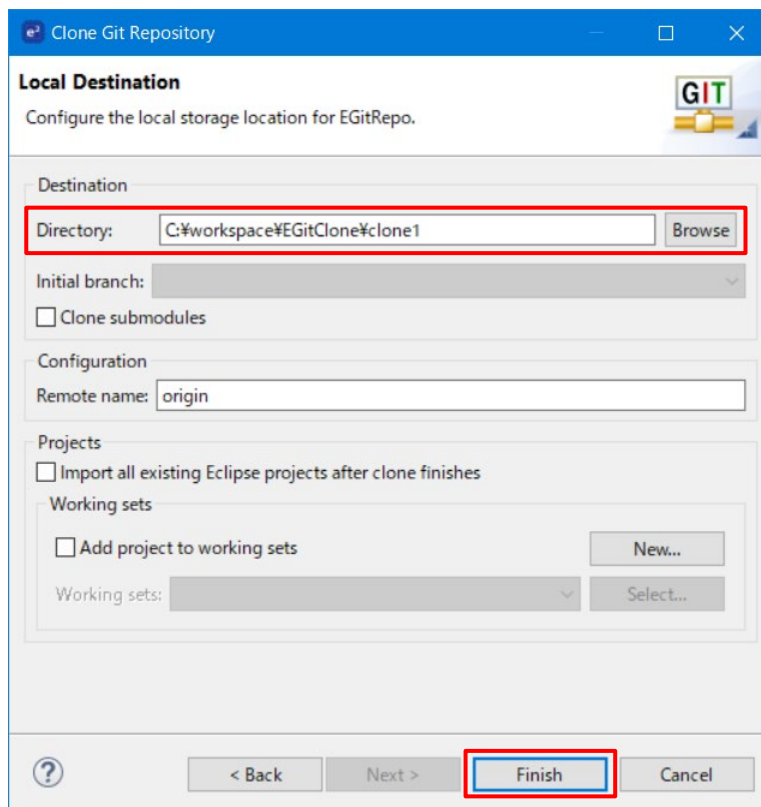


Figure 13 [Clone Git Repository - Local Destination] dialog

### 4. Version control of a project

This section shows how to control the version of a project with EGit.

You can specify a file which you will control the version of with [Ignored Resources] page in [Preferences] dialog. The recommended setting is to manage the source files, header files, and .xxxxx (.project and .cproject) as version control files.

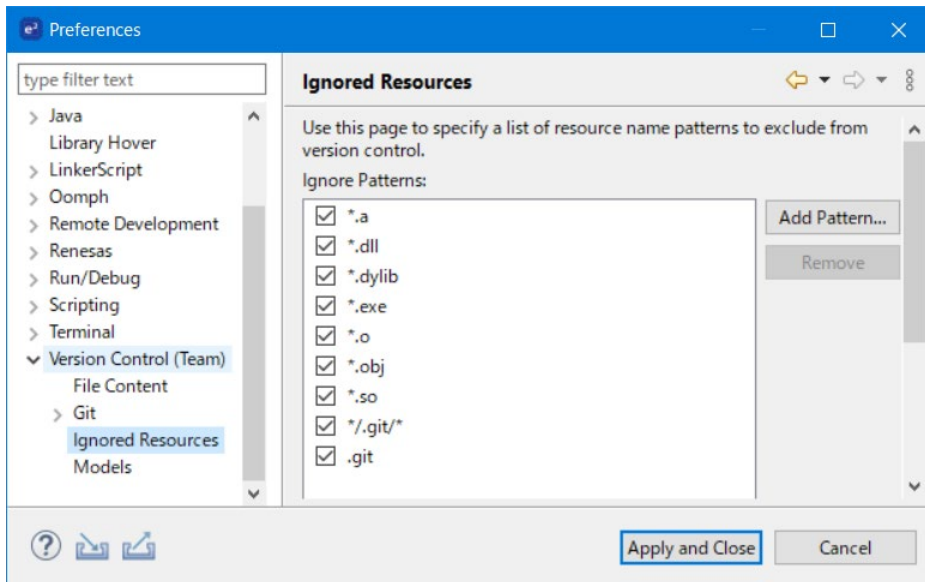


Figure 14 [Ignored Resources] page

#### 4.1 Adding a project to Git repository

- 1) Switch to "C/C++" perspective, click [File] > [New] > [C/C++ project].
- 2) In the [New C/C++ Project - Templates for New C/C++ Project] dialog, choose "Renesas RX" in the left-hand margin and "GCC for Renesas RX C/C++ Executable Project" and click [Next >] button.

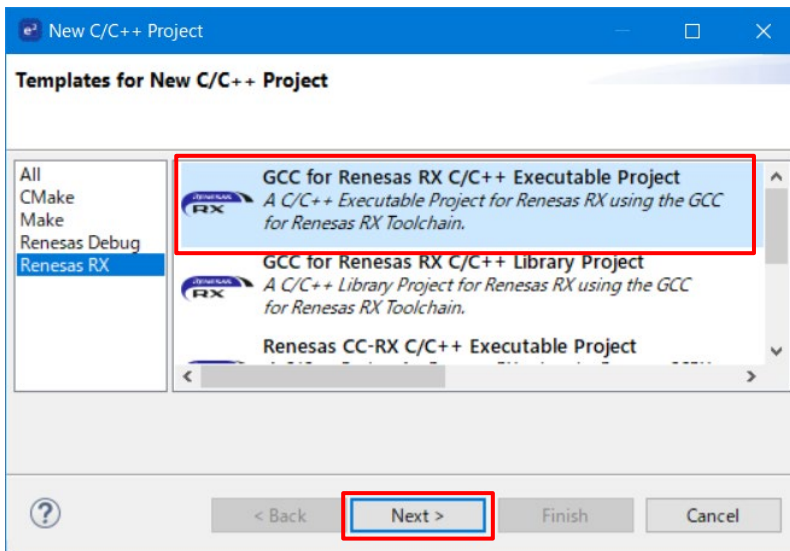
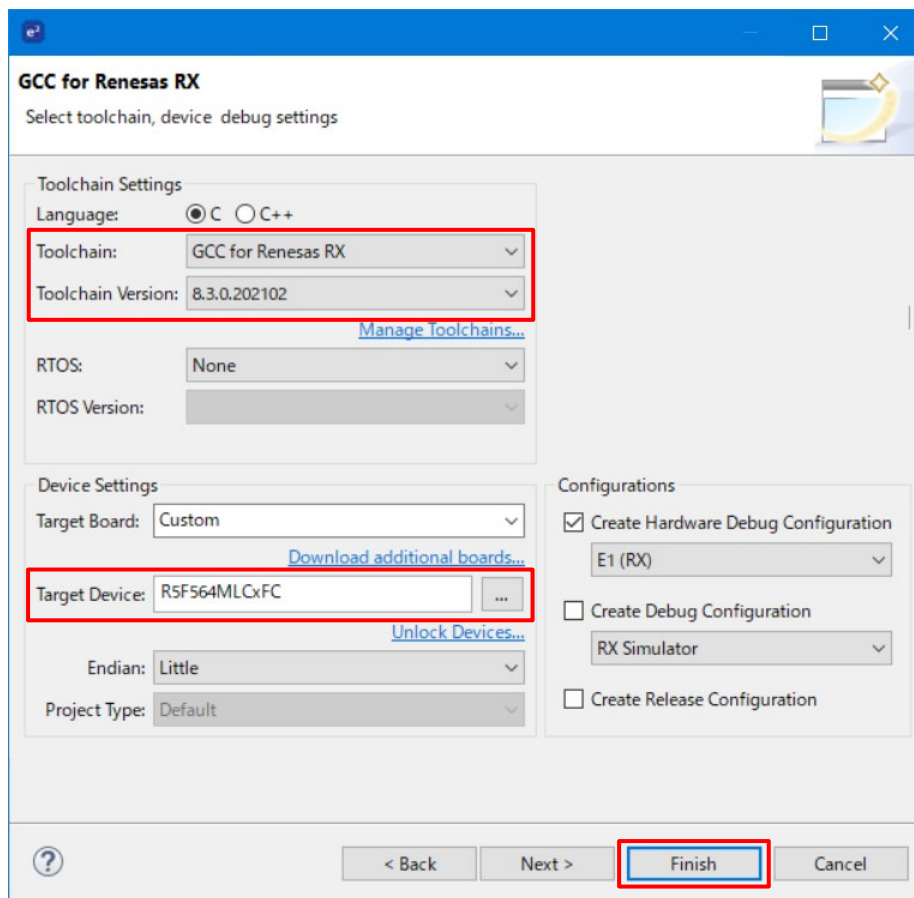


Figure 15 [New C/C++ Project - Templates for New C/C++ Project] dialog

- 3) In [Project name:] enter the name "SampleEGit" and click [Next >] button.
- 4) In the [Select toolchain, device & debug settings] page, enter the following information (other values can remain at default):
  - [Toolchain:] "GCC for Renesas RX"

- [Toolchain version:] "8.3.0.202102"
- [Target Device:] "R5F564MLCxFC"



**Figure 16 [Select toolchain, device & debug settings] page**

- 5) Click [Finish] button. The project is created and is accessible in the [Project Explorer] view.
- 6) Build the project to make sure that there are no build issues.
- 7) Clean the project, to remove auto generated files, before committing to version control.
- 8) Right-click on the project and select [Team] → [Share project...]
- 9) In the [Share Project - Configure Git Repository] dialog, select the new project "SampleEGit", then select the new cloned repository in [Repository] drop-down-list. Click [Finish] button.

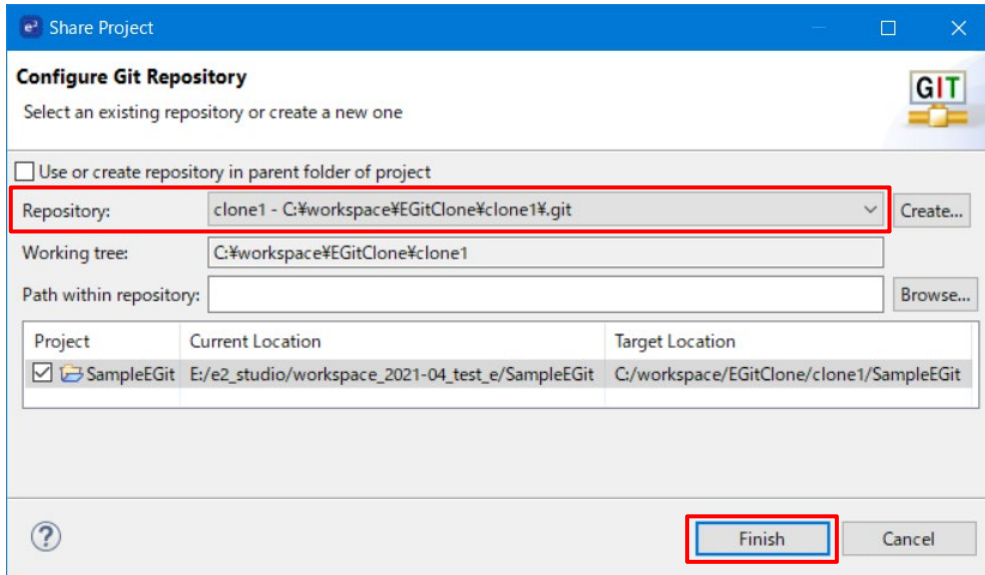


Figure 17 [Share Project - Configure Git Repository] dialog

- 10) Switch to "Git" perspective. In the [Git Repositories] view, click on the repository that you have selected in step 9). In the [Git staging] view, click [Refresh] button, the new files in the project will be listed in [Unstaged Changes] listbox.  
Note: You can press [Link with Editor and Selection] button to update the unstaged changes automatically.

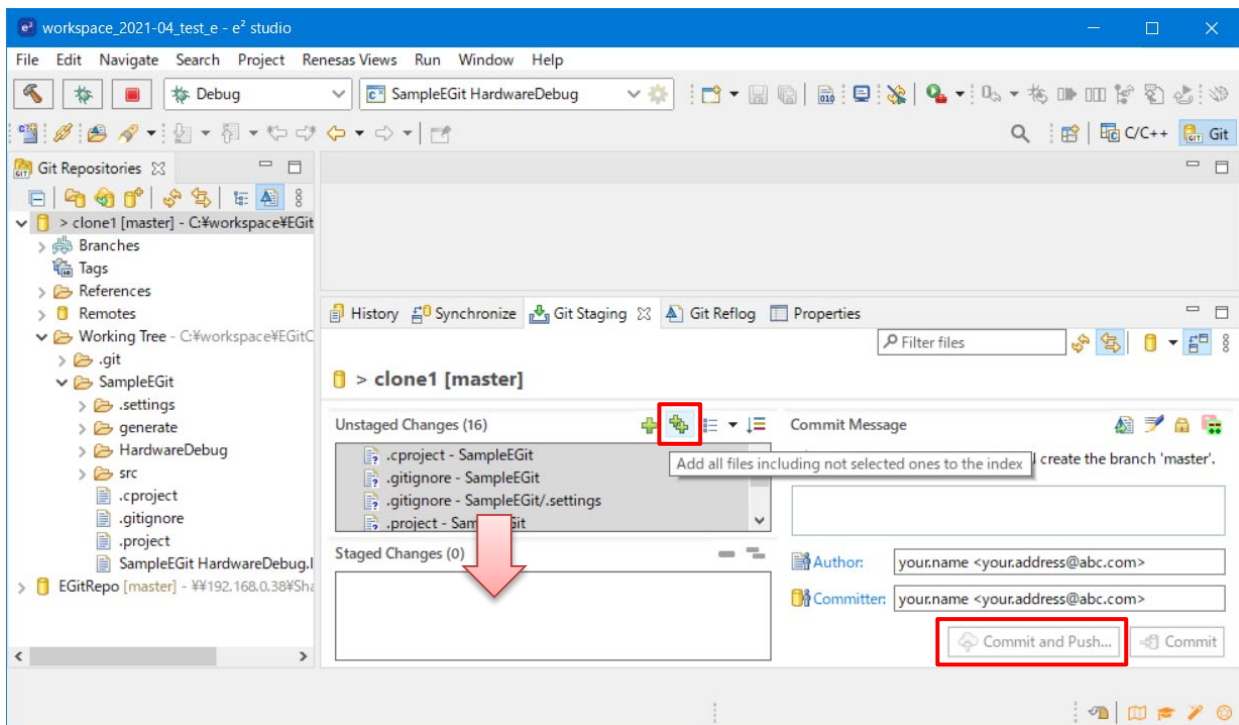


Figure 18 Commit new project to repository

- 11) Click [Add all files including not selected ones to the index] button, input messages into the [Commit Message] textbox (to comment for commit details), then click [Commit and Push] (if you would like to commit to local and remote repository) or [Commit] (if you would like to commit to local repository).
- 12) If it is the first time you push to remote repository, EGit will ask user to configure the remote repository and push behavior.  
Select the "master" repository created in section 3.1 in "Remote:", select "Rebase" in [When pulling:] drop-down-list, then click [Preview >]. Note that the "master" branch is created at this step if this is the first commit to the "master" repository.

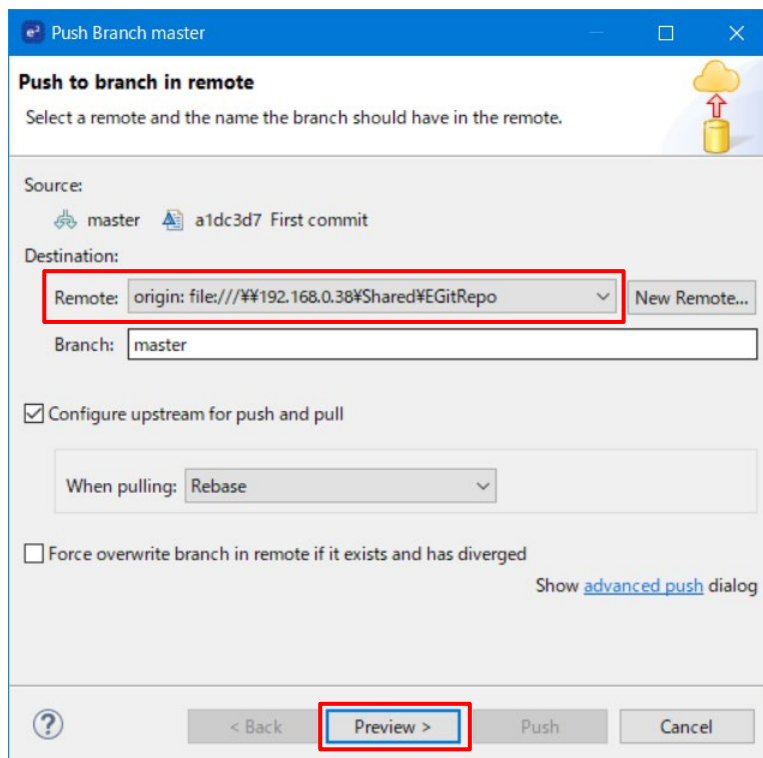


Figure 19 [Push Branch master - Push to branch in remote] dialog

- 13) Click [Push] button in the [Push Branch master - Push Confirmation] dialog.
- 14) Click [Close] button in the [Push Results...] dialog.

## 4.2 Committing a change

- 1) Open a file from the project (e.g. "SampleEGit.c") and modify it in "C/C++" perspective. Save the file.

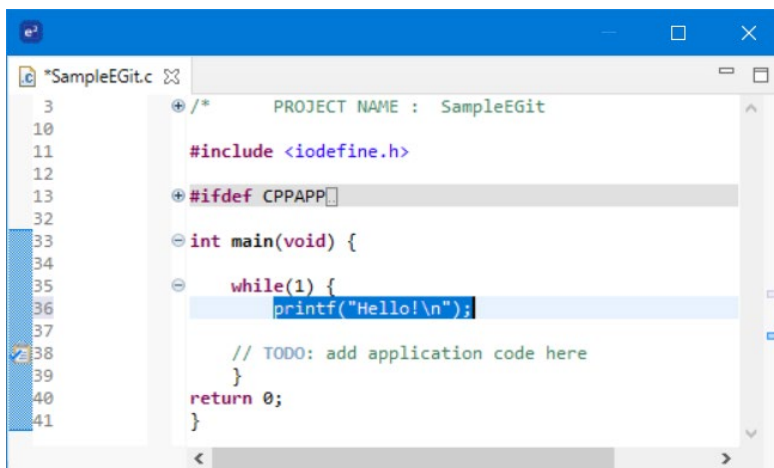


Figure 20 Edit SampleEGit.c

- 2) In the [Git staging] view, click [Refresh] button, the file will be listed in [Unstaged changes] listbox.
- 3) Click [Add selected files to the index] button, input messages into the [Commit Message] textbox and click [Commit and Push] button.
- 4) Click [Close] button in the [Push Results...] dialog.

### 4.3 Reviewing a change

- 1) In the "Git" perspective, click on a file to check a change made on it, then open the [History] view. The history of the file will be shown.

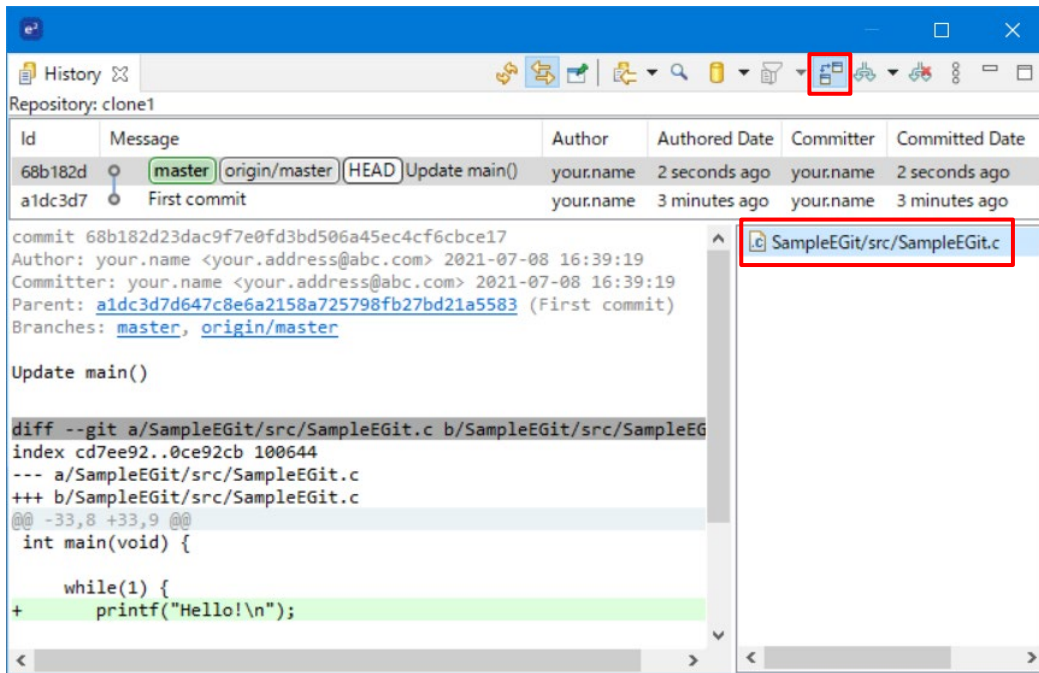


Figure 21 [History] view

- 2) Click on [Compare Mode] button of [History] view, then double click on the file to view the last change in "C Compare Viewer".

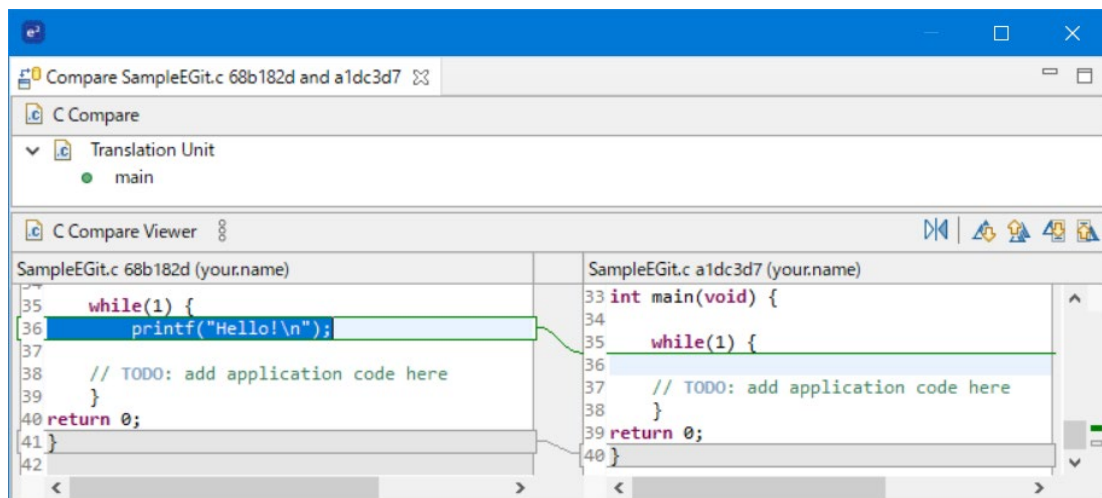


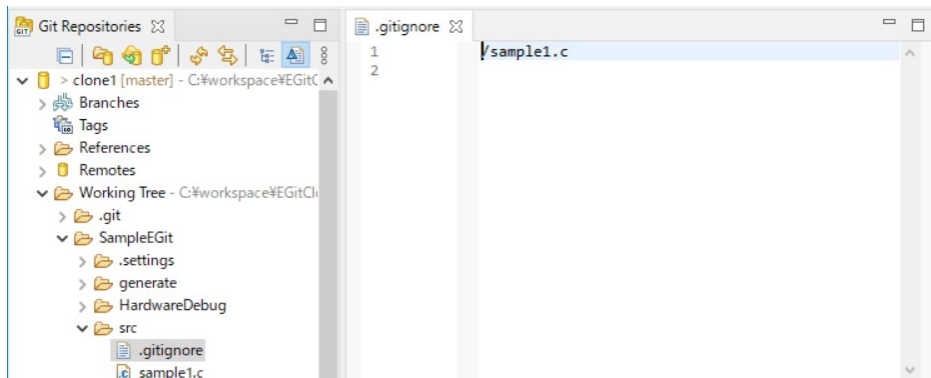
Figure 22 C Compare Viewer

### 4.4 Excluding a file from version control

To exclude a file from version control, follow below steps:

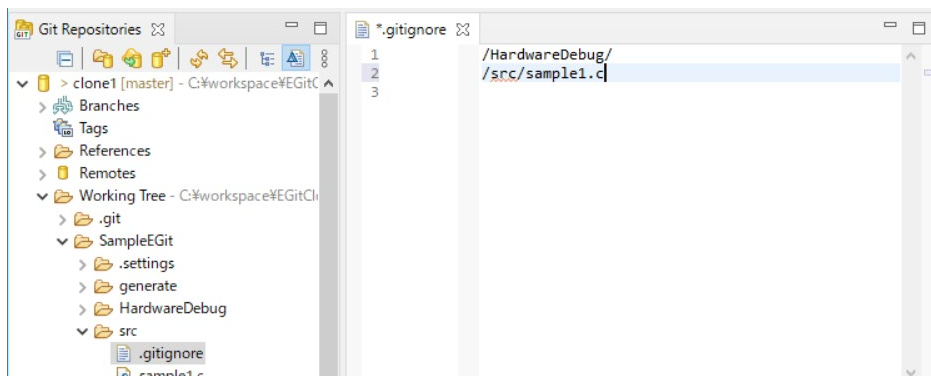
- 1) Switch to "C/C++" perspective, right click on the file, select [Team] > [Ignore]
- 2) Switch to "Git" perspective, click [Refresh] button and a new file name .gitignore is created in the same folder as the file you have selected to exclude from version control. Its content is the path to ignored file.





**Figure 23 .gitignore file**

- 3) Every time you perform step (1), a .gitignore file will be created in the same folder with the ignored file (if it does not exist). To use only one .gitignore file for a project, you can create only one .gitignore file in project folder, and update it with the path to ignored file whenever you would like to exclude a file from version control.



**Figure 24 .gitignore file**

- 4) Commit the .gitignore file to repository as section 4.2.

### 5. Synchronizing works among users

The workflow on EGit is suggested as follows:

- 1) Create a bare repository in a common server.
- 2) The user who starts the project will clone the repository, add the project into version control using the clone repository, and commit (and push) the changes.
- 3) Other users clone the repository in step (1) for first time working with this repository, perform updates, fetch from remote repository to get new updates from other users, rebase to synchronize between local and remote tracking branches (if there are conflicts, resolve them and continue the rebase), then commit and push all the changes to repository.

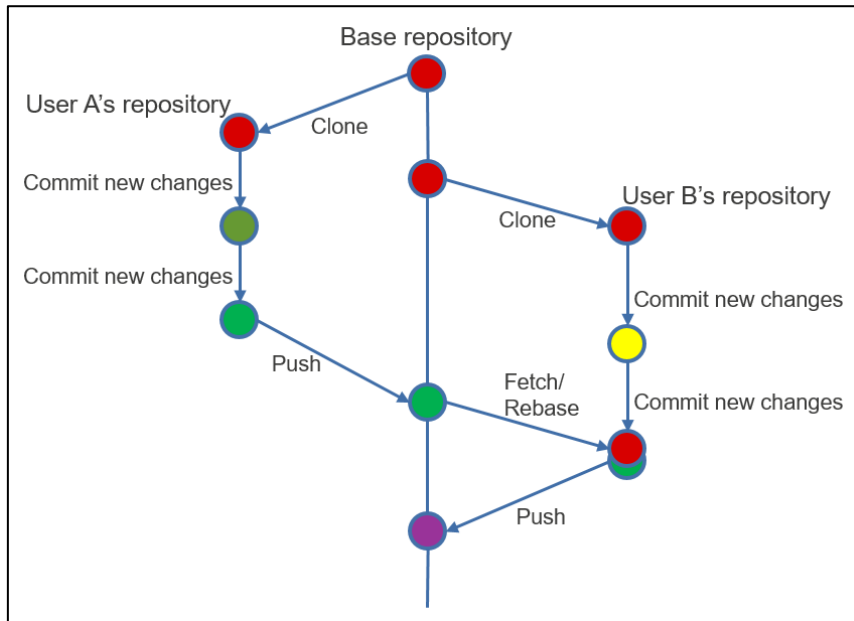


Figure 25 Illustration of rebasing

Following section will show how to rebase from the remote repository to local repository.

#### 5.1 Rebasing from remote repository

Rebasing is needed when the information about remote repository is out of date because other users have pushed new information to it. When you work on the local repository, before pushing something to remote repository, it's suggested to rebase first to avoid conflicts.

To produce the scenario as Figure 24, follow below steps:

- 1) Clone 2nd repository from the "master" repository created in section 4 as instruction in section 5.1 (cloned repository in "client machine 2" illustrated in Figure 6).

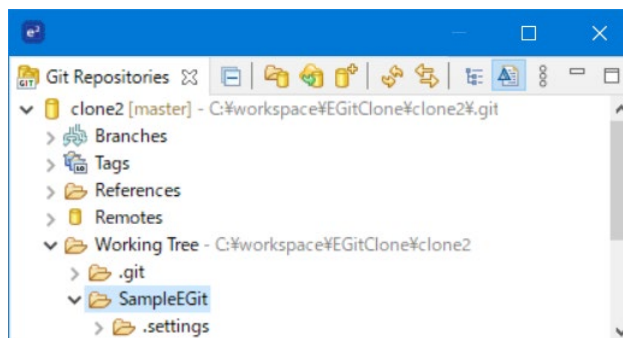
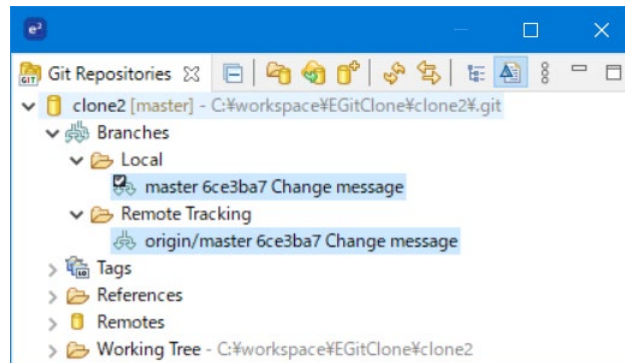


Figure 26 Clone 2nd repository

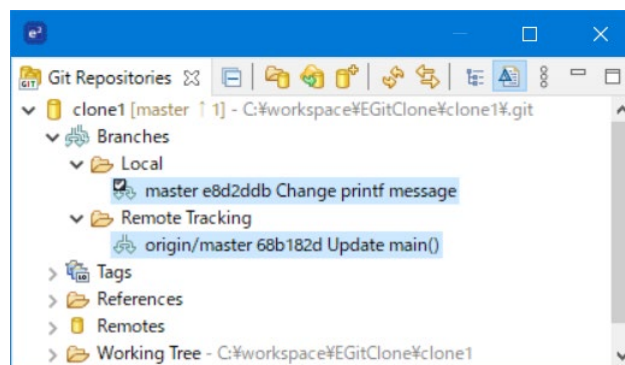


- 2) In the working directory of "clone2", open a file (e.g. "SampleEGit.c"), modify it, commit and push to remote repository. You can see the "Local" branch and "Remote tracking" branch of "clone2" is the same.



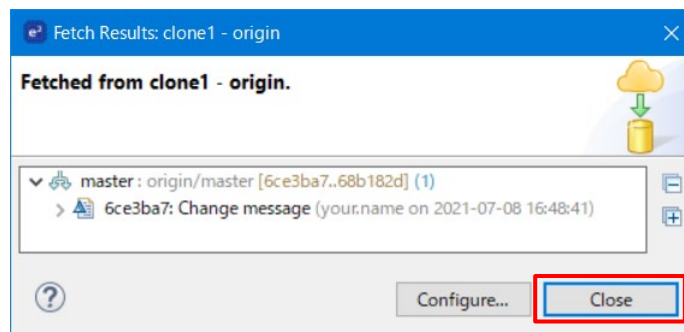
**Figure 27 Clone 2nd repository after edit SampleEGit.c**

- 3) In the working directory of "clone1", open "SampleEGit.c", modify it in another place, commit and push it to remote repository. You will get error message about conflict. The local and remote tracking branch of "clone1" are now different.



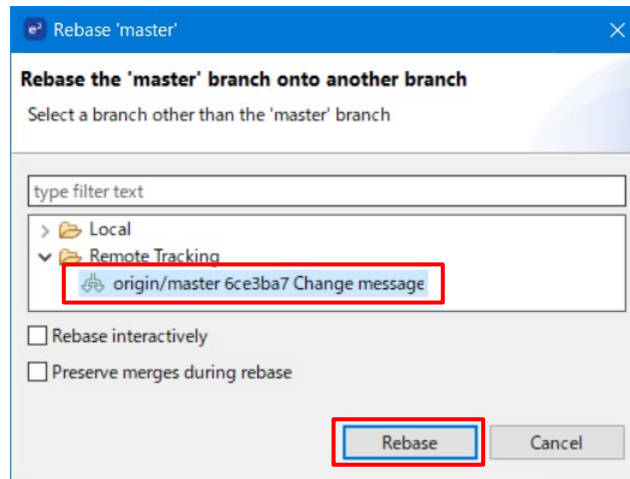
**Figure 28 Clone 1st repository after edit SampleEGit.c**

- 4) Right click on the repository "clone1", select [Fetch from origin]. The [Fetch Results...] dialog will show the information from remote repository committed by "clone2" in above steps. Click [Close] to close the [Fetch Results...] dialog.



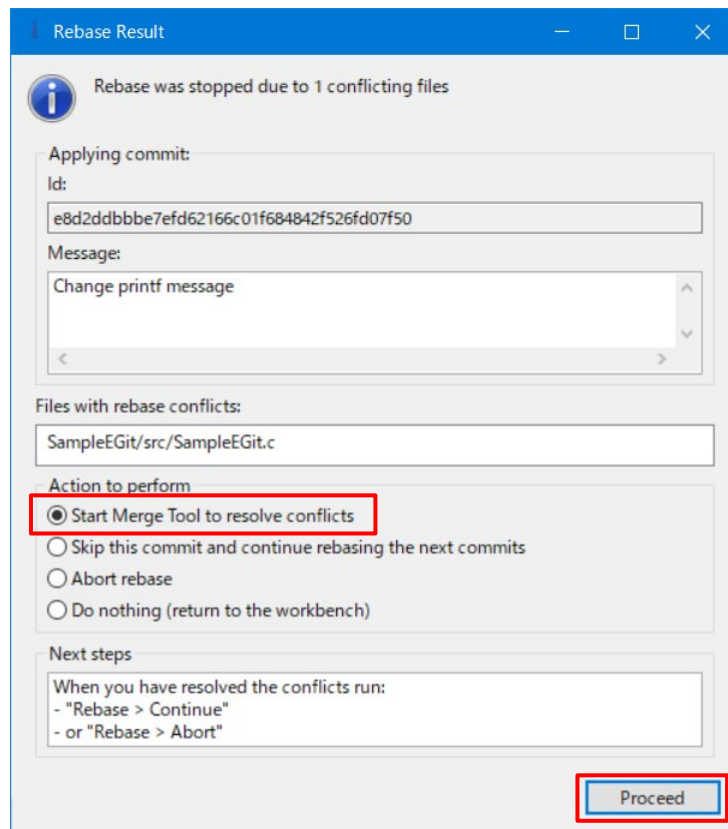
**Figure 29 [Fetch Results...] dialog**

- 5) Now the "Remote tracking" branch of "clone1" is updated same as "clone2". Right click on "clone1", select [Rebase]
- 6) In the [Rebase...] dialog, select the branch from remote repository to rebase. Click [Rebase] button.



**Figure 30 [Rebase...] dialog**

- 7) The [Rebase Result] dialog will show the conflicts in file "SampleEGit.c". You can have 3 options: merge to solve the conflict, skip the commit from "clone1", abort this rebase, or just return to workbench without doing anything. Select [Start Merge Tool to resolve conflicts]. Click [Proceed] button.



**Figure 31 Rebase Result] dialog**

- 8) The "C Compare Viewer" view will be opened and show the difference between "Local" vs. "Remote tracking" branch of "clone1". The right window is from "Remote tracking" and the left one is from "Local". You edit the text in left window as your expectation, then press Ctrl-S on the keyboard to save it.

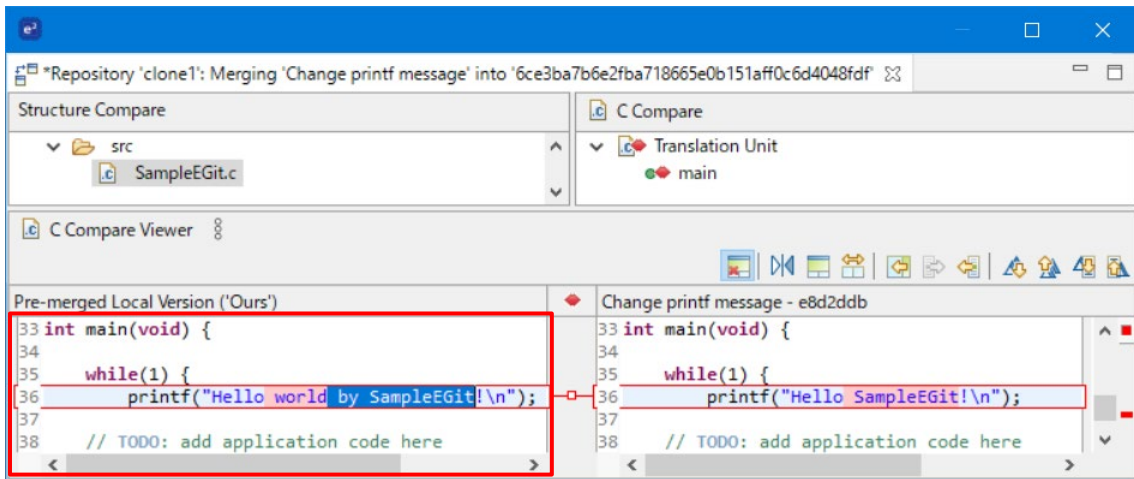


Figure 32 C Compare Viewer

- 9) Click [Add selected files to the index] button, and click [Continue] button.

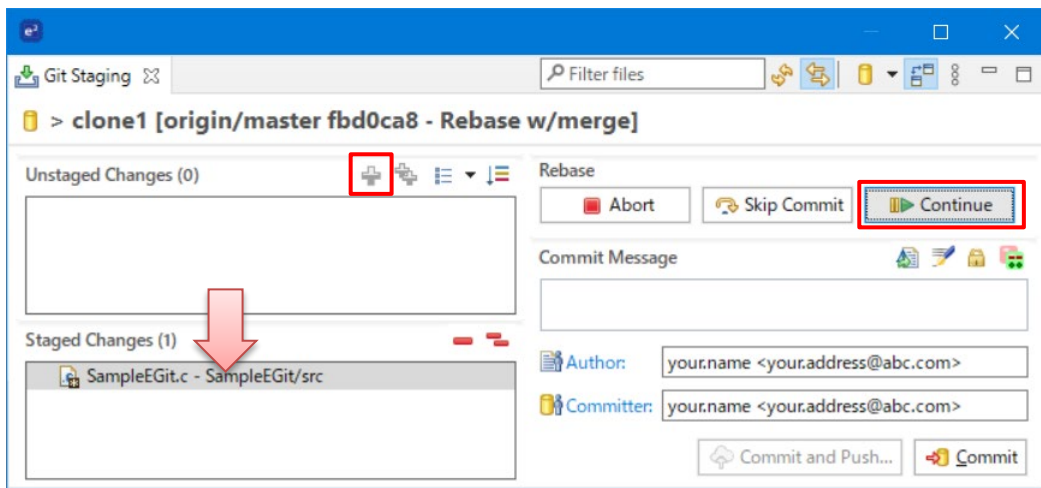


Figure 33 Rebase

- 10) Push the changes from “clone1” to remote repository by right click on “clone1”, select [Push to origin]. Click [Close] in the [Push Results...] dialog. Now the "Local" and "Remote tracking" branch of "clone1" are the unified.

**Revision History**

Rev.	Date	Description	
		Page	Summary
1.00	Oct 25, 2018		Initial document
1.01	Jul 26, 2021	All	Update to 2021-07 procedure and behavior

# General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

## 1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity.

Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

## 2. Processing at power-on

The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.

## 3. Input of signal during power-off state

Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

## 4. Handling of unused pins

Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

## 5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

## 6. Voltage application waveform at input pin

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.).

## 7. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

## 8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

## Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
  2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
  3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
  4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
  5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
    - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
    - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
- Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
  7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
  8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
  9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
  10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
  11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
  12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

## Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,  
Koto-ku, Tokyo 135-0061, Japan  
[www.renesas.com](http://www.renesas.com)

## Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

## Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:  
[www.renesas.com/contact/](http://www.renesas.com/contact/).