



Razor Hybrid Installation and Configuration Guide

Note: This guide will cover an Enterprise version of the Administration Tool. Not all features of this tool are available in the Evaluation Pack or Professional Versions

Overview

Razor is a lightweight solution. To try out and test it, simply install all three applications, the Admin, Server, and Client (along with Git and MariaDB) directly on your desktop. There is no need to set up a separate server and you can run Razor in either a traditional "Classic" or "Hybrid" development mode. You can start with the default set of forms, workflows, and automated email notifications, and then customize whatever you need from there.

System Requirements:

Windows Server:

- 1) Windows Server 2012 or better
- 2) Operating System Disk (C Drive) > 200GB
- 3) Separate Git repository drive > 100GB Not on the C drive
- 4) Do not use a shared drive for the Git repository

Windows Desktop – Test Drive

- 1) Windows 7, 10 or 11
- 2) Operating System Disk (C Drive) > 100GB
- 3) Separate Git repository drive > 50 GB not on the C drive

Linux Server:

- 1) CentOS 8 Stream, Ubuntu xxx, Rocky 8 +
- 2) Operating System Disk Size 100GB or Better
- 3) Separate Git repository drive > 100GB Not on the root drive

Note: Linux servers and clients will require the installation of Mono.

Downloads:

Download and install the following three Razor software applications by clicking on each icon.



Razor Admin Tool



Razor Server Software



Razor Client Software



If you haven't already done so, be sure to download and install Git and MariaDB. By clicking on each one of the two icons (just above) you will be able to download and install both applications.

Razor Hybrid Administration Tool Installation

1. Once the downloads are completed and ready, you can begin the installation process as follows.
2. First Install the Git package where the Razor server will be running.
 - a. Select all the default settings and complete the installation
 - b. Note DO NOT change the default installation folder
3. Install the MariaDB on to the C drive but install the Database on the separate volume by following these instructions
 - a. Create the database directory on the separate volume i.e.,
G:\MariaDBRazorDatabases\Data
 - b. Next proceed to install the MariaDB on the C Drive but on the following step:

User settings

Default instance properties

MariaDB 10.6 (x64) database configuration

MariaDB Server

☒ **Modify password for database user 'root'**

New root password: Enter new root password

Confirm: Retype the password

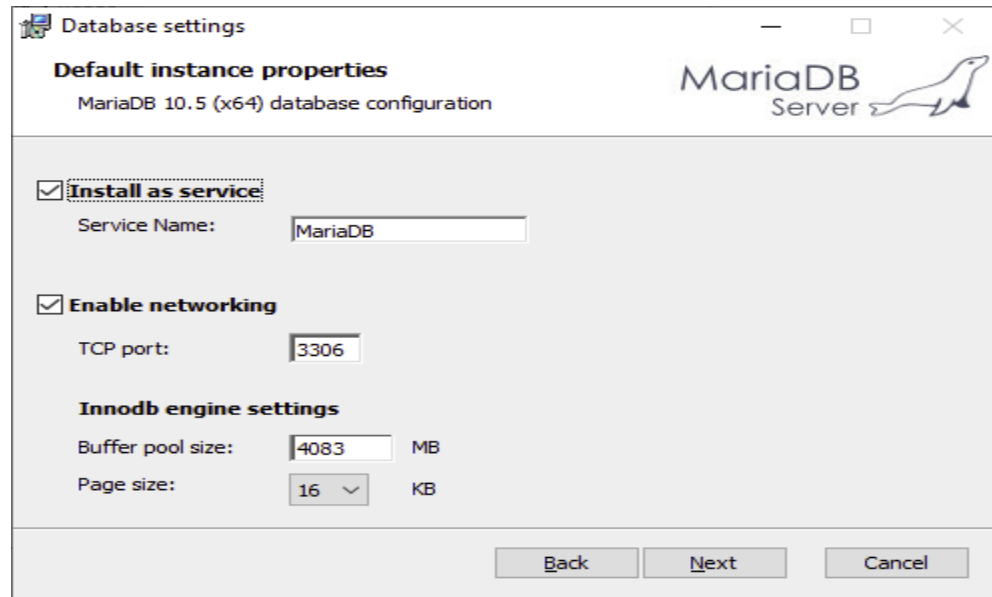
☒ **Enable access from remote machines for 'root' user**

☒ **Use UTF8 as default server's character set**

Data directory

- c. On the default instance properties window, enable access..., and Use UTF8 as the character set. Also, **make sure that you remember the password** entered
- d. Use the browse button to point to the new data directory
- e. Since this is where most of the growth is using a separate volume makes some sense.
- f. Click Next

- g. On the second default Instance Properties page make sure it looks exactly like below



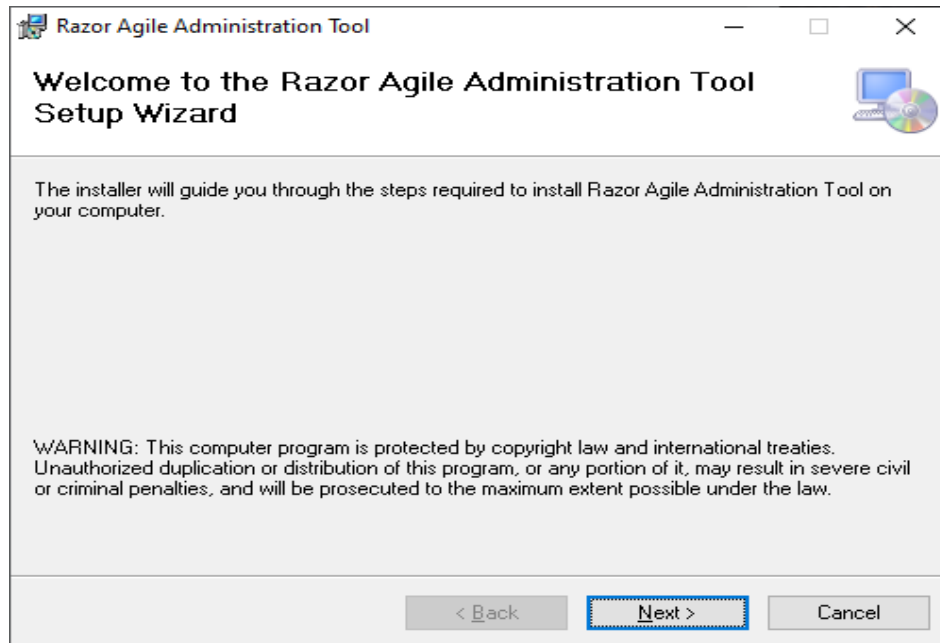
The screenshot shows a window titled "Database settings" with the subtitle "Default instance properties" and "MariaDB 10.5 (x64) database configuration". The MariaDB Server logo is in the top right corner. The window contains the following settings:

- ☒ **Install as service**
Service Name:
- ☒ **Enable networking**
TCP port:
- InnoDB engine settings**
Buffer pool size: MB
Page size: KB

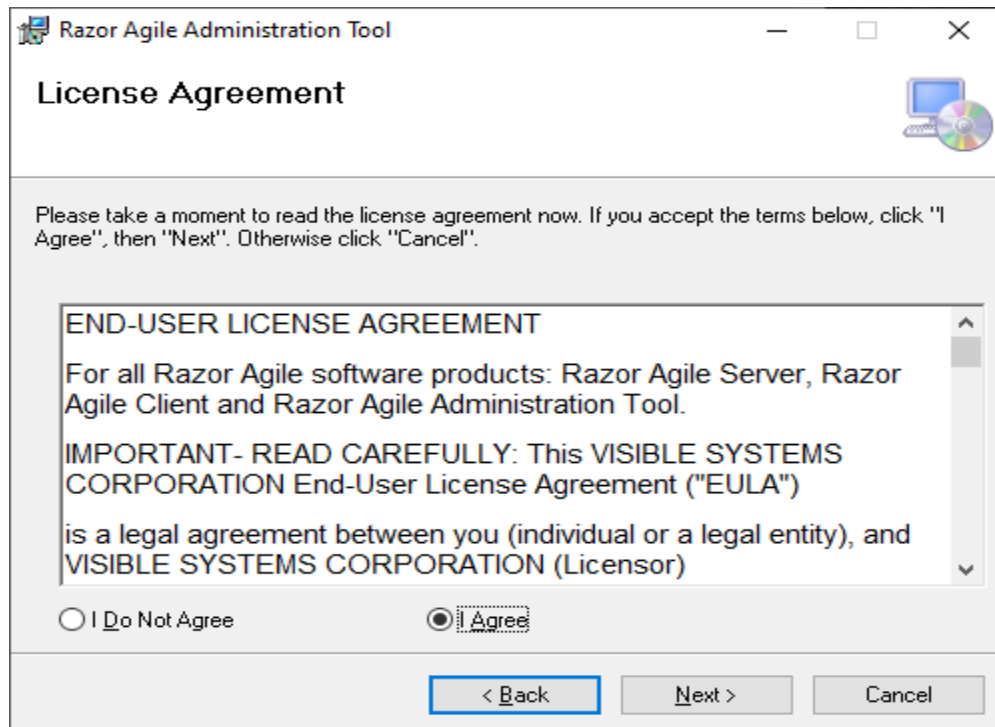
At the bottom right, there are three buttons: "Back", "Next", and "Cancel".

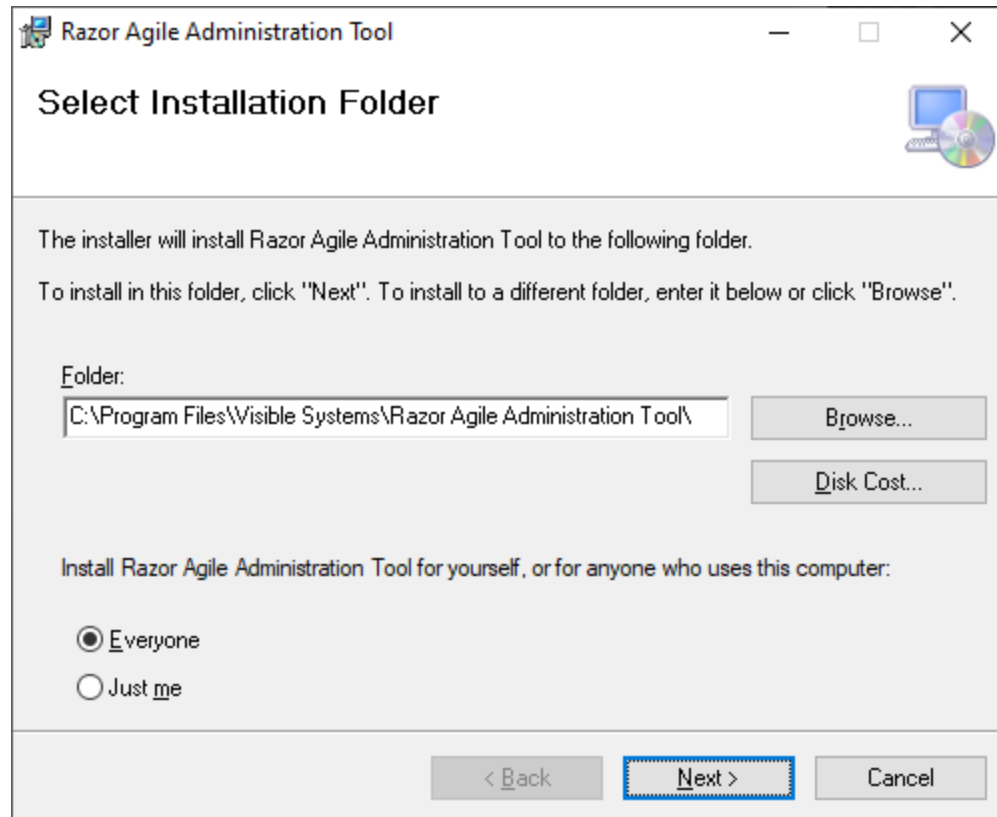
- h. Click Next
i. Click Install

4. Next install the Razor Administration tool



- a. Click on Next
- b. Accept the License Agreement and click next





- c. Leave the Default settings DO NOT change the installation Folder
- d. Click Next

Razor Agile Administration Tool

Razor 6 Database Configuration

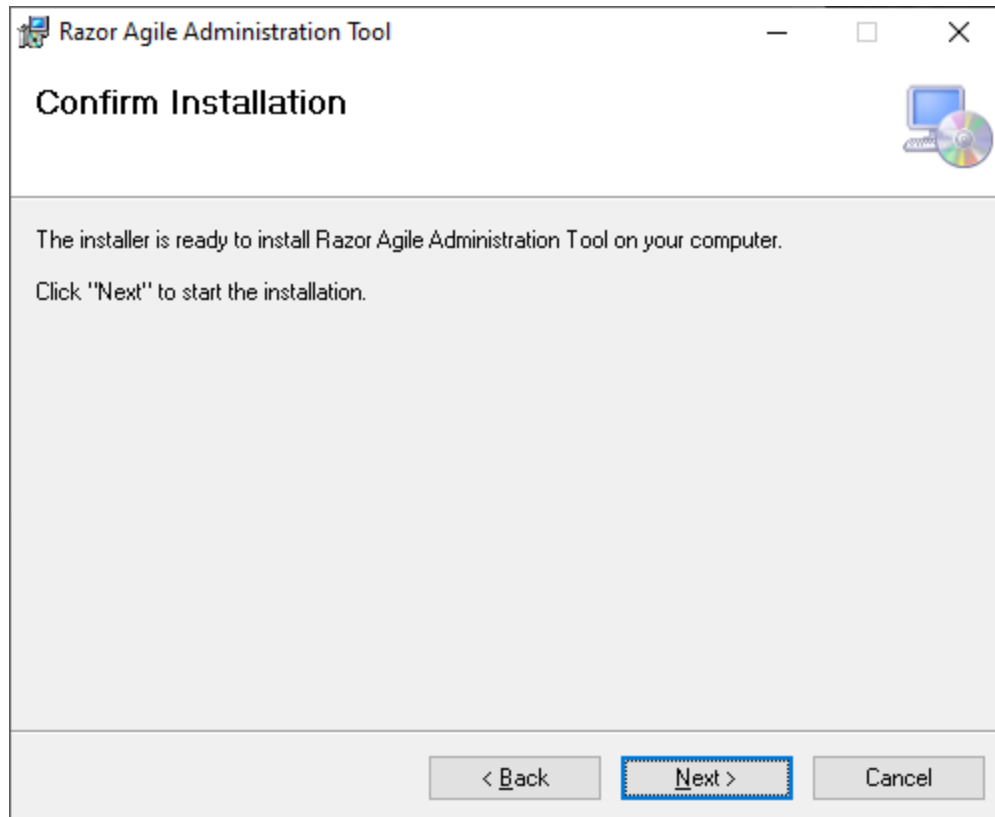
Enter the location of the Razor6 Database (or 127.0.0.1 if local) and the connect port number

Database Server:

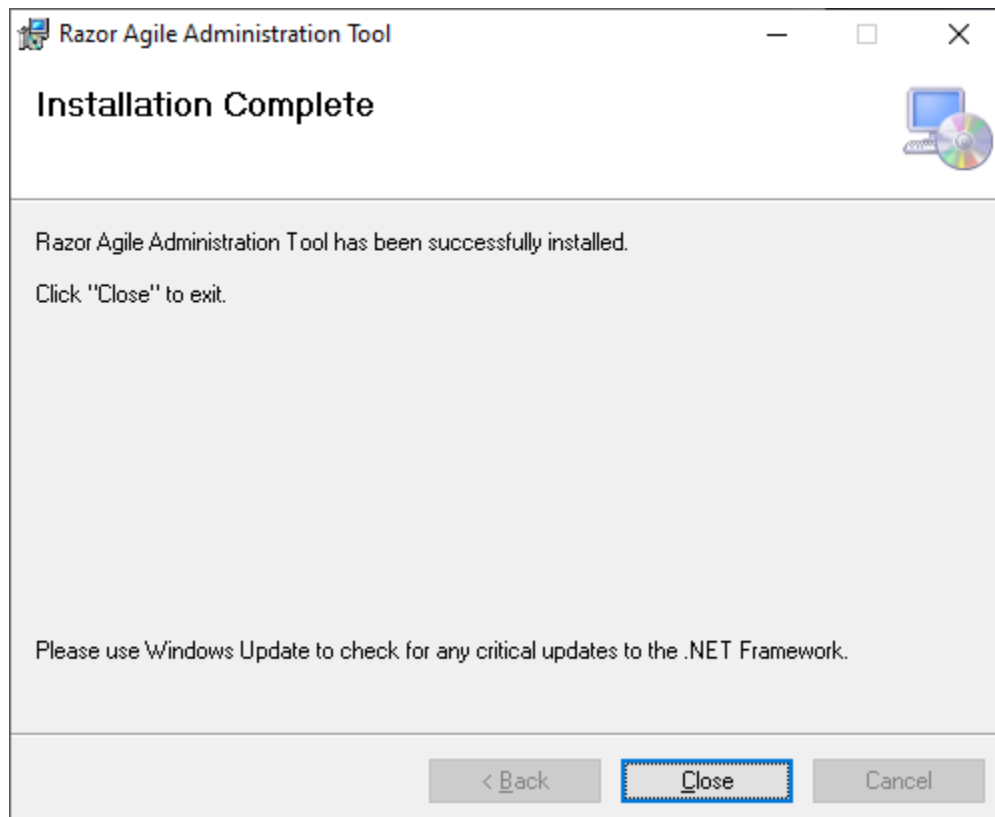
Database Connect Port Number:

< Back **Next >** Cancel

- e. Since the database is local, leave these settings as they are. If they are not change them to the Database server and port number as needed
- f. Click Next
- g. The Razor Administration tool is now ready to install - Click Next



- h. Click Close to complete the installation



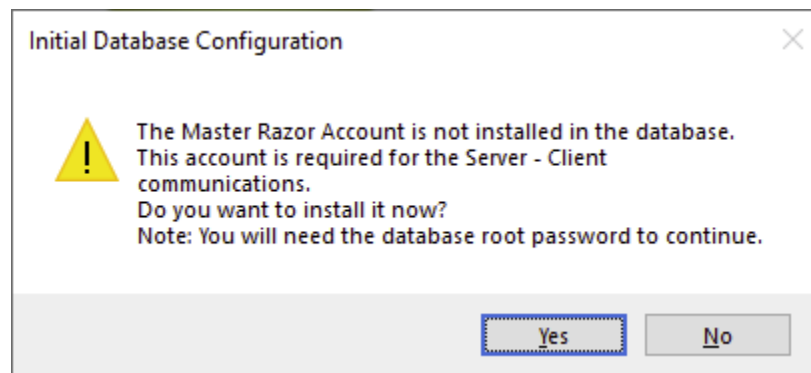
5. There should be an Icon on your desktop that looks like:



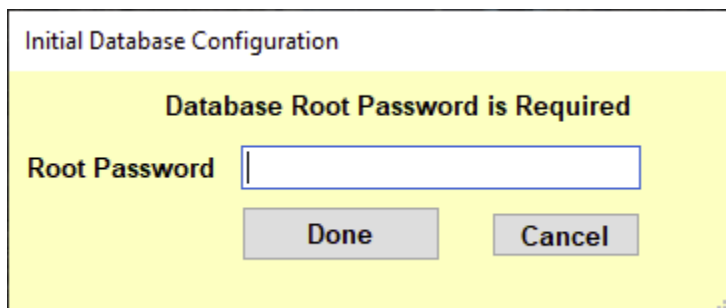
6. Double Click on the Icon to begin the final phase of the installation

Razor Hybrid Administration Tool Final Configuration

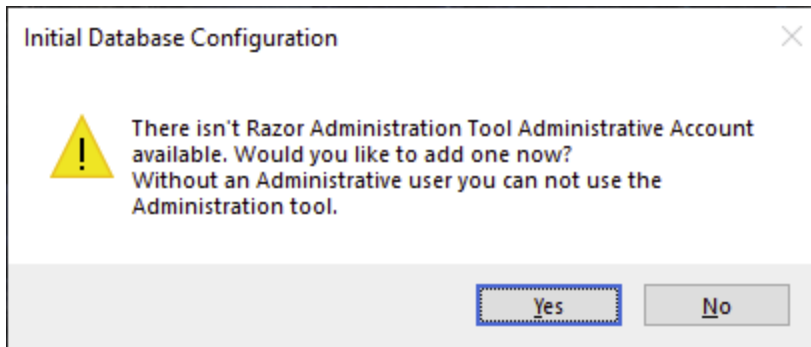
1. When the Icon is clicked for the first time there are some final configuration items to complete.



2. This first configuration is to install the user that communicates between the Admin tool, server and client. Without it, Razor Hybrid cannot function. You must have the database root password to install this user (See step b above). If you click on No, the program will exit
3. Click Yes



4. Enter the root password for MariaDB from step v above.
5. Click on Done
6. Click on close and the next window will appear. Select yes



7. The next screen is asking for a Razor Administration tool administrator. This account is only used for the Administration Tool. Click on Yes.

A dialog box titled "Creating Razor Administrator Administrative Account." with a yellow background. It contains several input fields and dropdown menus. The fields are: "User ID:" with a dropdown menu showing "razoradmin"; "Password:" with a text input field; "Confirm Password:" with a text input field; "Razor Group:" with a dropdown menu showing "Admin"; "User's Full Name (Optional)" with a text input field; and "User's Email (Optional)" with a text input field. At the bottom, there are three buttons: "Done", "Clear", and "Cancel".

8. The suggested user is presented, but you can change the user if desired. The login and passwords have restrictions such as login length and password strength. Once the login and passwords are set, click Done.

You must provide this login and password to the Razor Administration Tool each time you use the tool.

Welcome to the Razor Hybrid Administration Tool

Razor Administration

Razor Hybrid Edition: Unlicensed

Licensing and Encryption

☐ Licensing Tier Information

☐ Request a Quote

☐ Add Razor Licensing

☐ Request Evaluation License

☐ Current License Status

☐ Request Free License

☐ Configure Encryption

☐ Update Razor Licensing

☐ Request Permanent License

Server and Database Configuration

☐ Set or Reconfigure Server IP address

☐ Set or Reconfigure Database IP address or Port

☐ Remote Server Configuration

User Authentication

☐ Configure Authentication Method

☐ Import Razor Password File

☐ Add or Update a Local User

☐ Delete a Local User

Razor Projects

☐ Create a New Git Based Project

☐ Add New Group to Existing Git Project

☐ Import Razor 5 Configurations for use in new Git Projects

Database Maintenance

☐ Backup Database

☐ Schedule Backups

☐ Restore from Backup

☐ Import Issue Design SQL file

Database Editors

☐ Edit Custom Configurations

Close

Click here to log in

About

Notice that most of the configuration options are greyed out at this time and the tool itself is labeled as Unlicensed. If you are unfamiliar with the licensing options for Razor Hybrid, select the button for Licensing Tier Information for an overview of the options.

Razor Hybrid License Tier Information				
Free License	Evaluation License	Basic License	Professional License	Enterprise License
Single Server Only Windows or Linux Server Scripting Enabled Local, LDAP or AD Accounts File Branching 128 Bit Encryption Database Backup and Restore Guiffy Diff Tool Free 1 year license 3 Seats	Single Server Only Windows or Linux Server Import Razor Classic Forms Scripting Enabled Local, LDAP or AD Accounts File Branching Database Editor 128 or 192 Bit Encryption Database Backup and Restore Guiffy Diff Tool License for 60 days 25 seats	Single Server Only Windows or Linux Server Local Accounts Only 128 Bit Encryption Database Backup and Restore Guiffy Diff Tool	Single Server Only Windows or Linux Server Import Razor Classic Forms Scripting Enabled Local, LDAP or AD Accounts File Branching Database Editor 128 or 192 Bit Encryption Database Backup and Restore Guiffy Diff Tool	Multiple Servers Windows or Linux Servers Import Razor Classic Forms Scripting Enabled Local, LDAP or AD Accounts File Branching Database Editor 128, 192 or 256 Bit Encryption Database Backup and Restore Guiffy Diff Tool
Close				

If you have never used Razor Classic nor Razor Hybrid, then please select request an Evaluation License. The license issued will be good for 60 days and provide for 25 license tokens. You can install as many clients as needed just remember each tool in Razor Hybrid will consume a token when a user logs into that tool. An Example:

A user starts their client and logs into Sprints and Versions. They will have consumed 2 of the 25 licenses available.

If you are requesting an Evaluation License then Close the License Tier Information screen and click on the Request Evaluation License button and fill out the form presented

Requesting a license for the Evaluation Edition

Customer:

Contact Email:

Phone Number:

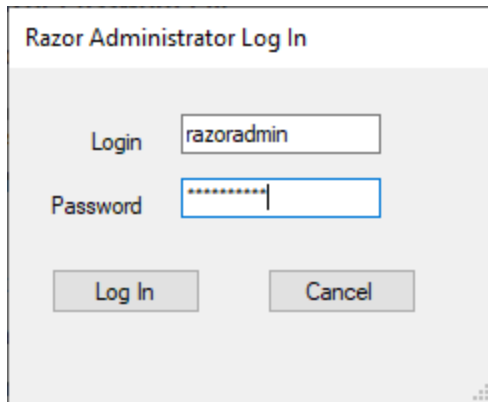
Address:

Note: If this Server does not have an Email client installed then select the button "Save to a File". Then copy and paste the information into a new email message and send the Email to: razor_support@visible-systems.com

At this point there are 2 choices once the form is filled out. If there is an email client installed here click on direct email. It will format the message and bring up the email so you can verify that all is good before sending it out. Otherwise, save to a file and then copy the file to a system that has an email client and send it to razor_support@visible-systems.com

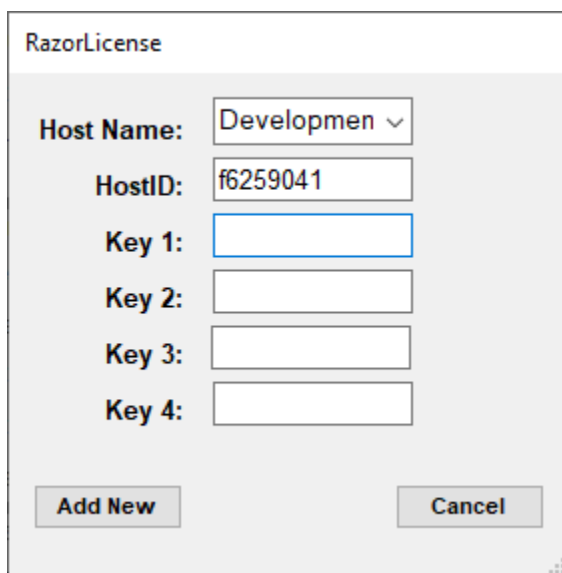
Once you receive the return email from Razor support it will include the details of the license issued. There will be four keys that must be installed in order to activate your license tier. So, let's begin

1. Click on Add Razor License and the log in window will pop up. Or you can click on "Click here to log in"



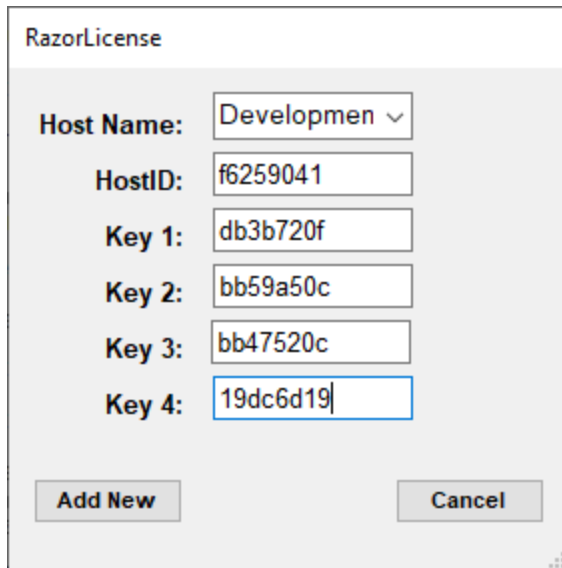
A screenshot of the 'Razor Administrator Log In' dialog box. It features a title bar with the text 'Razor Administrator Log In'. Below the title bar, there are two input fields: 'Login' with the text 'razoradmin' and 'Password' with masked characters '*****'. At the bottom of the dialog, there are two buttons: 'Log In' and 'Cancel'.

2. The Licensing window will pop up. Ensure that the HostID and Hostname matches the one in the email sent to you with the license keys.



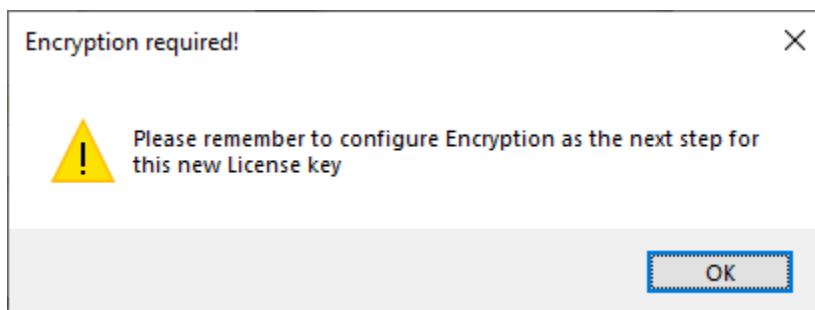
A screenshot of the 'RazorLicense' dialog box. It has a title bar with the text 'RazorLicense'. The dialog contains several fields: 'Host Name' with a dropdown menu showing 'Developmen', 'HostID' with the text 'f6259041', and four empty text boxes labeled 'Key 1:', 'Key 2:', 'Key 3:', and 'Key 4:'. At the bottom, there are two buttons: 'Add New' and 'Cancel'.

3. Next simply copy and paste the keys into each key slot and click on Add new



A dialog box titled "RazorLicense" with a light gray background. It contains several input fields: "Host Name:" with a dropdown menu showing "Developmen", "HostID:" with the text "f6259041", "Key 1:" with "db3b720f", "Key 2:" with "bb59a50c", "Key 3:" with "bb47520c", and "Key 4:" with "19dc6d19". At the bottom, there are two buttons: "Add New" and "Cancel".

4. The next window will remind you to install the encryption key for all server / client communications



A dialog box titled "Encryption required!" with a close button (X) in the top right corner. It features a yellow warning triangle icon with an exclamation mark. To the right of the icon, the text reads: "Please remember to configure Encryption as the next step for this new License key". At the bottom right, there is an "OK" button.

5. Click on install and the Administration Tool will unlock to your available license tier

Razor Hybrid Edition: Evaluation License

Licensing and Encryption

- ☐ Licensing Tier Information
- ☐ Add Razor Licensing
- ☐ Current License Status
- ☐ Configure Encryption
- ☐ Request a Quote
- ☐ Request Evaluation License
- ☐ Request Free License
- ☐ Update Razor Licensing
- ☐ Request Permanent License

Server and Database Configuration

- ☐ Set or Reconfigure Server IP address
- ☐ Set or Reconfigure Database IP address or Port
- ☐ Remote Server Configuration

User Authentication

- ☐ Configure Authentication Method
- ☐ Import Razor Password File
- ☐ Add or Update a Local User
- ☐ Delete a Local User

Razor Projects

- ☐ Create a New Git Based Project
- ☐ Add New Group to Existing Git Project
- ☐ Import Razor 5 Configurations for use in new Git Projects

Database Maintenance

- ☐ Backup Database
- ☐ Schedule Backups
- ☐ Restore from Backup
- ☐ Import Issue Design SQL file

Database Editors

- ☐ Edit Custom Configurations

Close

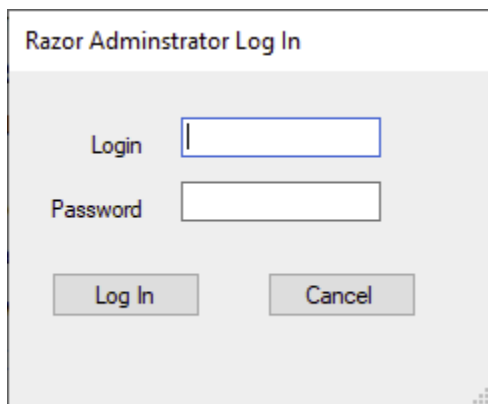
Logged in as: razoradmin

About

Using the Administration Tool

1) Logging into the Administration Tool

The administrator can log into the tool by either clicking on any of the desired features or by clicking on the “Not Logged In” text located at the bottom of the tool. The administrator is presented with a request to log in

A screenshot of a Windows-style dialog box titled "Razor Administrator Log In". The dialog has a light gray background. It contains two text input fields: the first is labeled "Login" and the second is labeled "Password". Below the input fields are two buttons: "Log In" on the left and "Cancel" on the right. The dialog box has a standard Windows border with a title bar.

Use the login and password that you created when the tool was installed (i.e., see page 8)

2) Licensing and Encryption

- a) Add Razor Licensing. Allows the Administrator to add licensing for other servers. This will be discussed when the server is released
- b) Update Razor Licensing. When the license is expired and you have received new keys, this is where the keys are updated
- c) Current License Status. Shows how many users are allowed, how many users are connected and expiration dates for the licenses.
- d) Configure Encryption. See next section
- e) Request Permanent License. Provides a template for requesting permanent keys.

Note: You can contact Visible Systems Corporation Sales by calling 617-902-0767 or sending an email to razor_sales@visible-systems.com.

The license agreement may be found at:

https://visible-systems.com/Products/Razor/license_agreement.htm

3) Configure Encryption

This function configures the pass-phrase between the client and server to fully encrypt the communications between the client and server. At the moment only 128-bit encryption is allowed. In the future, Enhanced encryption will provide 192-bit protection, and Ultimate encryption will provide 256-bit protection for all database data. This added security will ensure that data is securely transmitted between the Razor clients and the Razor server.

Configure Encryption Parameters

Secure Communications Encryption Configuration

Select the Encryption Strength

☒ Standard Encryption (Default)
☐ Enhanced Encryption
☐ Ultimate Encryption

Pass Key Entry

Bit Count

NOTE: Pass Key text box will turn green when the key is the correct length

Install Cancel Encrypt Help

The pass-key is entered by the user and should be a random set of characters using upper case, special characters and numbers. It should be a totally meaningless phrase. Here is an example:

Configure Encryption Parameters

Secure Communications Encryption Configuration

Select the Encryption Strength

☒ **Standard Encryption (Default)**
☐ **Enhanced Encryption**
☐ **Ultimate Encryption**

Pass Key Entry

Bit Count

NOTE: Pass Key text box will turn green when the key is the correct length

4) User Authentication

- Configure Authentication Method. Razor Hybrid will use the internal MariaDB user/password database by default
- LDAP, Secure LDAP and Single Sign On (Linux) or NIS to authenticate users.
- Import Razor Password Files. The Administration tool can import the existing Razor 5 Password file leaving current settings for users intact.
- Add or update a Local user. Used by the Razor administrator to manage the local users.
- Delete a Local User. The administrator can delete a local user here.

Note: Managing users at the Active Directory, LDAP etc. levels are managed at their source.

Note: When importing a Razor 5 password file, the file must have read permissions and be accessible by the Administration Tool

- 1) Click on the Configure Authentication Method and this screen is displayed

Select Authorization Source

Configuration Authentication Source

☐ Local Razor Database

☐ LDAP

☐ Secure LDAP

☐ Linux Log on Services (SSO, NIS, Local)

LDAP Server

Secure LDAP

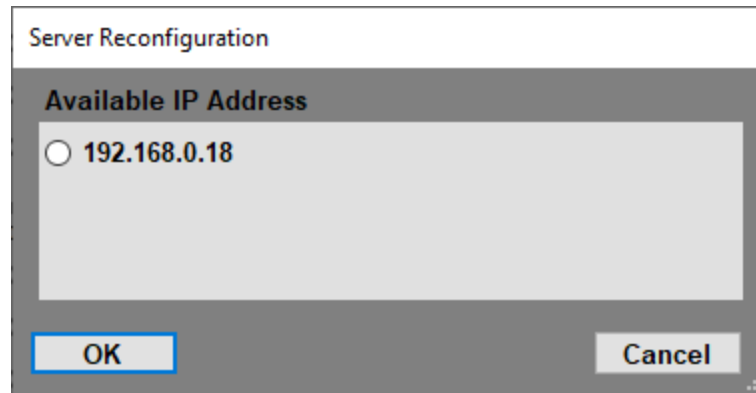
Server Name or IP

Base DSN

Linux Server

Test Connection **Cancel**

- 2) Select any of the Authentication methods
 - a) Local Razor Database contains all users in MariaDB
 - b) LDAP will use the Domain LDAP server
 - c) Secure LDAP will use a secure connection to the LDAP server
 - d) For NIS, Single Sign on enter the Server name
- 3) Next Select Set or Reconfigure Server IP Address
 - a) The first form will display all of the Active IP address that are available for this server.
 - b) Select the IP address that all Razor Hybrid communications are to happen over



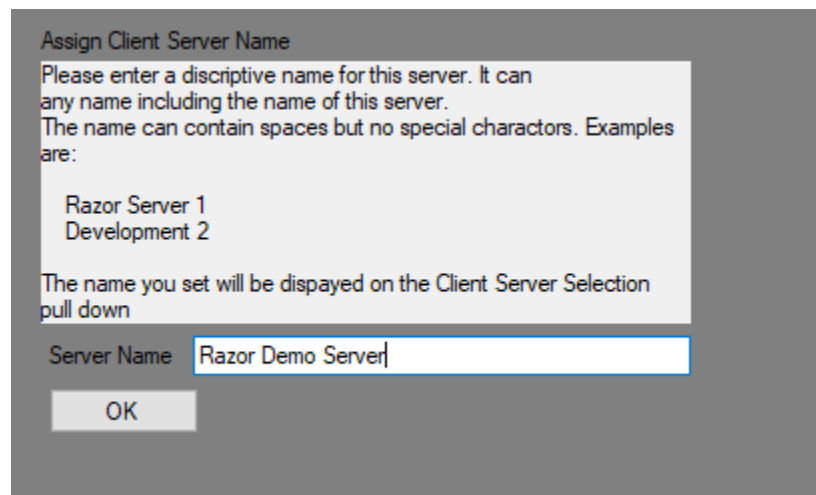
Server Reconfiguration

Available IP Address

☐ 192.168.0.18

OK Cancel

- c) Click on OK
- d) Next set the name of the server that is to be presented to the users. This name does not have to be the host name of the server



Assign Client Server Name

Please enter a descriptive name for this server. It can be any name including the name of this server. The name can contain spaces but no special characters. Examples are:

Razor Server 1
Development 2

The name you set will be displayed on the Client Server Selection pull down

Server Name

OK

5) Optional Import of Existing Issue formats

Razor Hybrid can use existing Issue formats from Razor Classic in all new Projects created. The import process is simple, but will require some adjustments to the final layout of the imported format due to the differences between the old GUI type and the more modern format now available. The Admin tool attempts to make the best guess on the new format, but sometimes not so great. The good news is that the updates to the format are done through the Admin Tool and are quick to do.

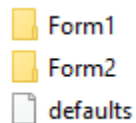
So, let's get started.

- a) Create a directory that is accessible to the Admin Tool. We will suggest Razor Imports
- b) Next create a directory under Razor Imports that has the same name as the Issur form you are importing. For Example:

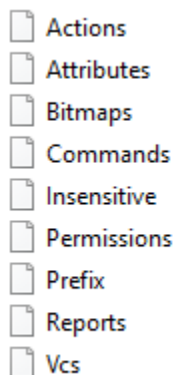
Issues*IssueFormForm1

The Sub directory is labeled Form1

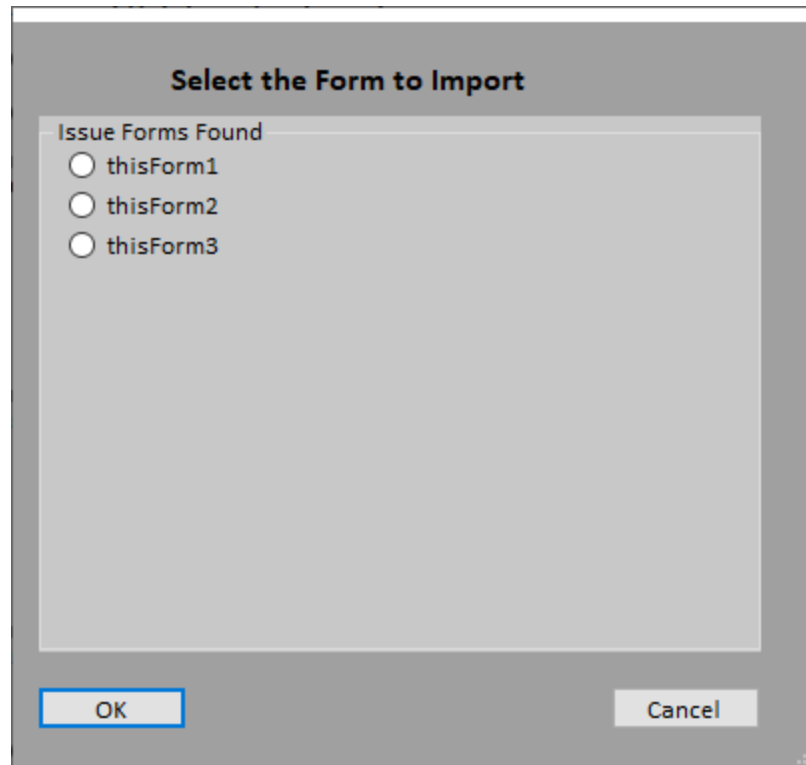
- c) If you have multiple forms create directories for each with the defaults file at the root



- d) Note the W or X defaults file is renamed to defaults.
- e) Next copy the files that you wish to import into the proper sub directory



- f) Next start up the Admin tool and select Import Razor Classic Configuration for use in new Git Projects
- g) Log in if needed
- h) Navigate to the Razor Imports directory and click OK
- i) The form should appear listing all the forms possible to import from the defaults file



- j) Notice there are 3 forms in this defaults file. Make sure that the Attributes, Permissions, Insensitive, Roles and Command files match the from you wish to import
- k) Select the form and click on OK. The form will and all supporting information is placed into the database

6) Razor Projects

a) Create a New Git Based Project

Add New Project

Project Directory

Git Project Name

Group Name

Issue Type Select ☒ Standard ☐ Use Custom Format

Available Formats

Versions Configuration ☒ Files Must Change ☒ Must Attach Issues

Baselines Config. ☒ Select Files Based on Issues Only ☒ Use Scripts

Administrator's Email

Issue Prefix

- 1) The Browse for Repository option provides a Directory browser so the Administrator can select the location where the new Project is to be created
- 2) Project Directory will display where the new Project is to be created
- 3) Git Universe Name - All characters are allowed except spaces
- 4) Group Name - All characters are allowed except spaces
- 5) Issue Type Select.
 - a) Standard will have the new Project use the built-in form for Issues, Versions and Baselines
 - b) Use Custom Format. If the Administrator has imported an existing Razor5 Defaults, Insensitive etc. Files, then when this option is selected the available formats will present themselves in the Available formats pull down.
- 6) Versions Configuration: Files must change before checking in and Must Attach an Issue to a file check out

- Add New Project

Browse for Repository

Project Directory

Git Project Name

Group Name

Issue Type Select

Available Formats

Versions Configuration

Baselines Config.

Administrator's Email

Issue Prefix

G:\

TestProject

DemoProject

☒ Standard ☐ Use Custom Format

☒ Files Must Change ☒ Must Attach Issues

☒ Select Files Based on Issues Only

☒ Use Scripts

razoradmin@email.com

Demo

Install

Cancel

b) Add a new group to an Existing Project.

Add Group To Existing Project

GIT Universe Selection

Group Name

Issue Type Select ☒ Standard ☐ Use Custom Format

Import Formats

☒ Complete (Issues, Versions, Baselines)
☐ Issues Only
☐ Versions and Baselines Only

Versions Configuration ☒ Files Must Change ☒ Must Attach Issues

Baselines Config. ☒ Select Files Based on Issues Only
☒ Use Scripts

Administrator's Email

Issue Prefix

Install **Cancel**

- 1) Select the Project to add the new group to add to
- 2) Enter a group name
- 3) Issue type select. Same as above
- 4) Select what to add. Complete adds (Issues, Versions and Baselines), Issues only or Versions and Baselines only
- 5) Versions Configuration
- 6) Razor Administrator's Email
- 7) Issue Prefix as before

Add Group To Existing Project

GIT Universe Selection RazorDemoProject ▾

Group Name DemoGroup2

Issue Type Select ☒ Standard ☐ Use Custom Format

Import Formats ▾

☐ Complete (Issues, Versions, Baselines)
☒ Issues Only
☐ Versions and Baselines Only

Versions Configuration ☐ Files Must Change ☐ Must Attach Issues

Baselines Config. ☒ Select Files Based on Issues Only
☒ Use Scripts

Administrator's Email razoradmin@email.com

Issue Prefix newDemo

Install Cancel

7) Fixing an Imported Issues Form

- 1) Bring up the Razor Client and log into Issues
- 2) An imported form may look something like this

New Issue

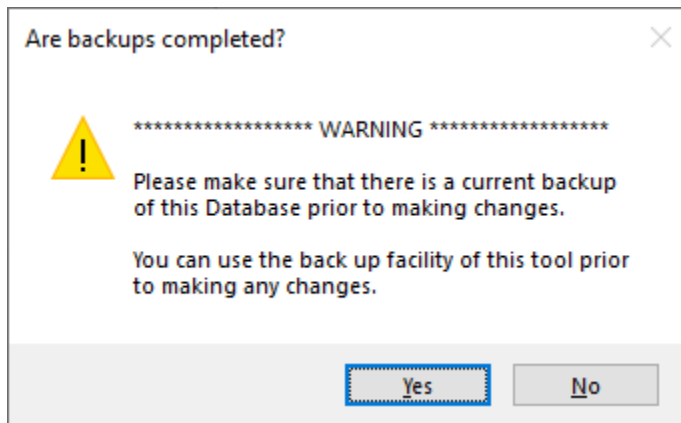
☐ Review ☒ Modify Print

Title

Originator arded

Production-Line Production-Station Functional-Test ▼

- 3) Note the text boxes are a bit shifted to the left and some items are located under others. Close the open issue but do not log out of the issues program.
- 4) To fix this bring up the Admin tool and select Edit Custom Configurations
- 5) A warning message will appear



- 6) Please heed this message and see the section on backing up the databases
- 7) The next screen will be which database to edit select attributes

Select Database

Database Selection

☐ actions
 ☐ roles

☐ attributes

☐ commands

☐ formindextable

☐ icons

☐ insensitive

☐ permissions

Close

8) The database editor will now be presented.

Database Editor (Use with Caution)

	formName	formType	formIndex	AttributeParent	AttributeType	AttributeName	Attr
	standard	Issues	1	IssueMain	PANEL	MainPanel	
	standard	Issues	1	MainPanel	TEXT_PANE	TEXT2	TEXT
	standard	Issues	1	MainPanel	TEXT_PANE	TEXT1	TEXT
	standard	Issues	1	MainPanel	GROUPBOX	AssignmentGroupBox	Issue
	standard	Issues	1	AssignmentGroupBox	TEXT_FIELD	AssignEmail	
	standard	Issues	1	AssignmentGroupBox	TEXT_FIELD	AssignedTo	
	standard	Issues	1	AssignmentGroupBox	LABEL	AssignEmailLabel	Email
	standard	Issues	1	AssignmentGroupBox	LABEL	AssignLabel	Assi
	standard	Issues	1	AssignmentGroupBox	GROUPBOX	ScopeGroupBox	Scop
	standard	Issues	1	AssignmentGroupBox	GROUPBOX	ScopeGroupBox	Scop

9) Scroll down until you see the field you need to change

Database Editor (Use with Caution)

	formName	formType	formIndex	AttributeParent	AttributeType	AttributeName	AttributeLabel
	thisForm1	Issues	4	IssueMain	PANEL	MainPanel	
	thisForm1	Issues	4	MainPanel	LABEL	TitleLabel	Title
	thisForm1	Issues	4	MainPanel	TEXT_FIELD	Title	
	thisForm1	Issues	4	MainPanel	LABEL	OriginatorLabel	Originator
	thisForm1	Issues	4	MainPanel	TEXT_FIELD	Originator	
	thisForm1	Issues	4	MainPanel	LABEL	ForwardedLabel	
▶	thisForm1	Issues	4	MainPanel	CHECK_BOX	Forwarded	Forwarded
	thisForm1	Issues	4	MainPanel	LABEL	Production-LineLabel	Production-Line
	thisForm1	Issues	4	MainPanel	TEXT_FIELD	Production-Line	
	thisForm1	Issues	4	MainPanel	CHECK_BOX	Production-Status	Production-Status

Close

Add a Row

Delete Row(s)

Update Database

10) Next slide over so the xLocation, yLocation are visible

Database Editor (Use with Caution)

utilLines	DateFormat	AttributeFormat	xLocation	yLocation
		0	13	13
		50	10	5
		1	36	5
		40	10	40
		1	64	40
		0	295	40
		1	295	40
		15	10	72
		1	92	72
		1	295	72

Close Add a Row Delete Row(s) Update Database

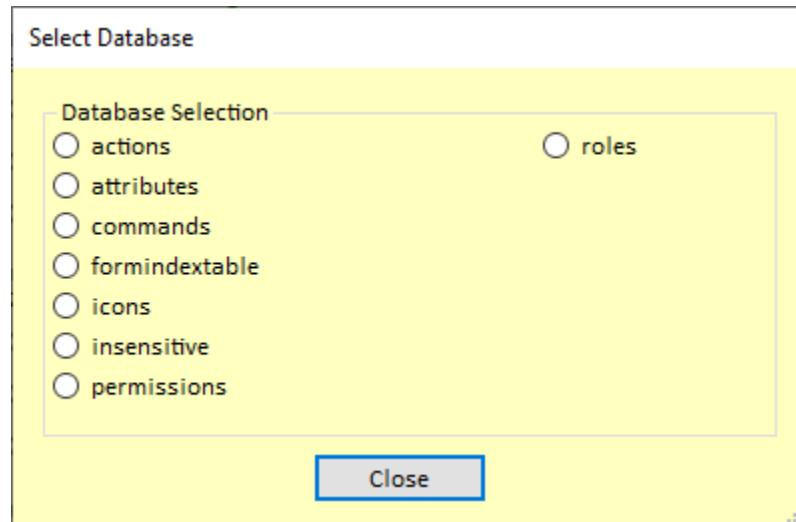
- 11) Click on the X location field for the attribute and increase its value to move the item to the left or decrease its value to move it to the right. If you need to move the Y location decreasing the value move the attribute up and increasing moves the attribute down.
- 12) Click on Update Database and select new issue on the Client tool. If the attribute is in the correct place you are done. Otherwise close the issue and repeat steps 11 and 12 until every look great.

8) Database Maintenance

- 1) Backup Database: Allows for complete backups of all Razor databases or the backup of individual databases
- 2) Restore Database: Will restore an existing backup

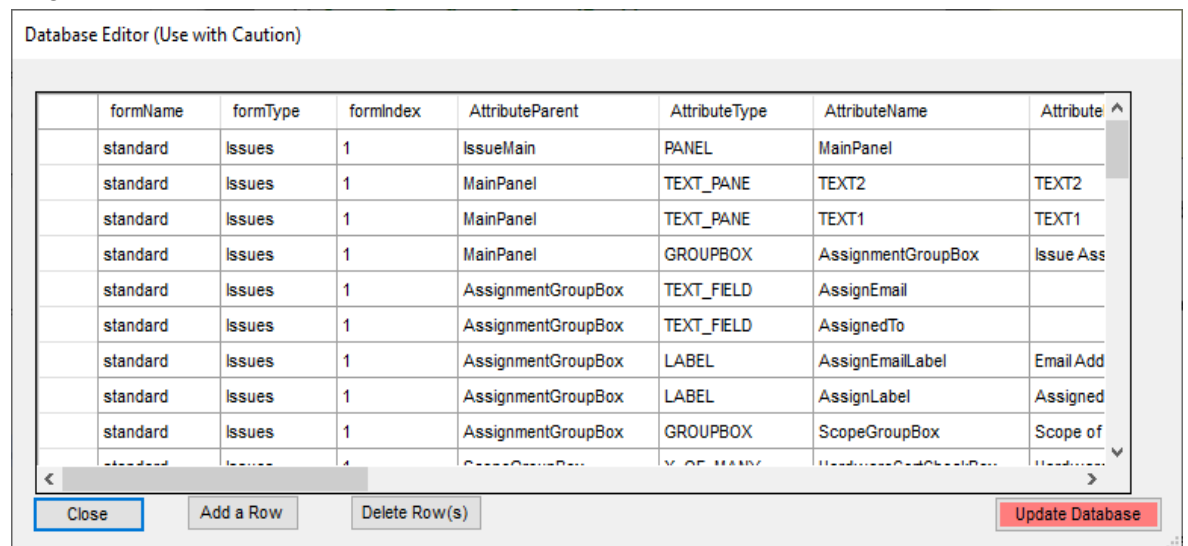
9) Adding Commands to the Command Pulldown

- a) Before you begin you will need two pieces of information
 - i) The Issue form name
 - ii) The Issue Index
- b) Start the Admin Tool and select Edit Databases



The 'Select Database' dialog box has a yellow background. It contains a section titled 'Database Selection' with a list of radio buttons: 'actions', 'attributes', 'commands', 'formindextable', 'icons', 'insensitive', and 'permissions'. To the right of this list is another radio button labeled 'roles'. At the bottom center is a 'Close' button.

- c) To get the form name select Attributes



The 'Database Editor (Use with Caution)' window displays a table with 8 columns: 'formName', 'formType', 'formIndex', 'AttributeParent', 'AttributeType', 'AttributeName', and 'Attribute'. The table contains 10 rows of data. Below the table are buttons for 'Close', 'Add a Row', 'Delete Row(s)', and 'Update Database'.

	formName	formType	formIndex	AttributeParent	AttributeType	AttributeName	Attribute
	standard	Issues	1	IssueMain	PANEL	MainPanel	
	standard	Issues	1	MainPanel	TEXT_PANE	TEXT2	TEXT2
	standard	Issues	1	MainPanel	TEXT_PANE	TEXT1	TEXT1
	standard	Issues	1	MainPanel	GROUPBOX	AssignmentGroupBox	Issue Ass
	standard	Issues	1	AssignmentGroupBox	TEXT_FIELD	AssignEmail	
	standard	Issues	1	AssignmentGroupBox	TEXT_FIELD	AssignedTo	
	standard	Issues	1	AssignmentGroupBox	LABEL	AssignEmailLabel	Email Add
	standard	Issues	1	AssignmentGroupBox	LABEL	AssignLabel	Assigned
	standard	Issues	1	AssignmentGroupBox	GROUPBOX	ScopeGroupBox	Scope of
	standard	Issues	1	AssignmentGroupBox	GROUPBOX	ScopeGroupBox	Scope of

- d) Write down the information Note case for the Form name and select close
- e) Next select Commands

Database Editor (Use with Caution)

	formName	whichTool	CommandName	CommandScript
*				

Close Add a Row Delete Row(s) Update Database

- f) Click into the form name box and enter the form name
- g) Next select the whichTool column and enter 1 for Issues 2 For Versions or 3 for Baseline commands
- h) Next select CommandName column and enter the information to show in the command pull down
- i) Finally select CommandScript column and enter the actual script location
 - i) Note the direction of the slashes / not \

Database Editor (Use with Caution)

	formName	whichTool	CommandName	CommandScript
	standard	1	This Is a Test	F:/RazorScripting/ssueCommands/TestScript.sh
*				

Close Add a Row Delete Row(s) Update Database

- j) If you need to add more lines select add row, otherwise click on update Database

10) Enterprise Server Remote Server Configuration

When using Razor Hybrid over several sites an occasion where Site A may need to assist developers at site B. Razor Hybrid licensing has an Enterprise option where this can be accomplished.

There are limited cautions that the developers at site A should adhere to which will be discussed below.

- a) First step is to bring up the Admin Tool on Site A's Razor Server
- b) Next Click Request a Quote button

Quote Request

Select License Type

☐ Basic License Number of Seats

☐ Professional License

☐ Enterprise License

Customer:

Contact Email:

Phone Number:

Address:

Note: If this Server does not have an Email client installed then select the button "Save to a File". Then copy and paste the information into a new email message and send the Email to: razor_support@visiblesystemscorp.com

- c) Next select the Enterprise License button and select the number of servers in this case it will be 2. The local server (A) and then the remote server B

Quote Request x

Select License Type

☐ Basic License

Number of Seats

10 ▾

☐ Professional License

Server Count

2 ▾

☒ Enterprise License

Customer:

Contact Email:

Phone Number:

Address:

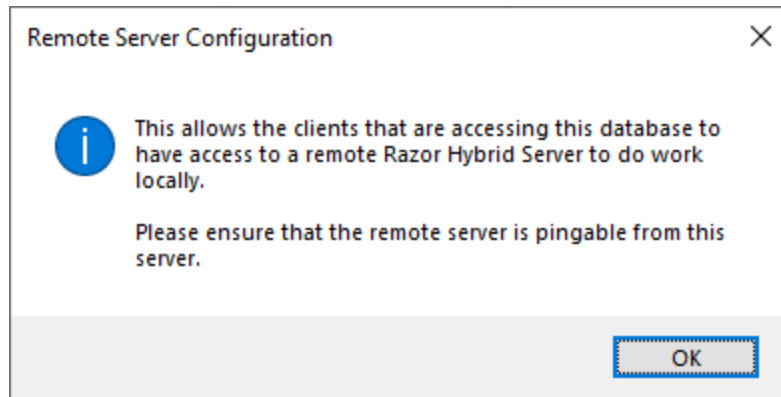
Note: If this Server does not have an Email client installed then select the button "Save to a File". Then copy and paste the information into a new email message and send the Email to: razor_support@visiblesystemscorp.com

Save to a File

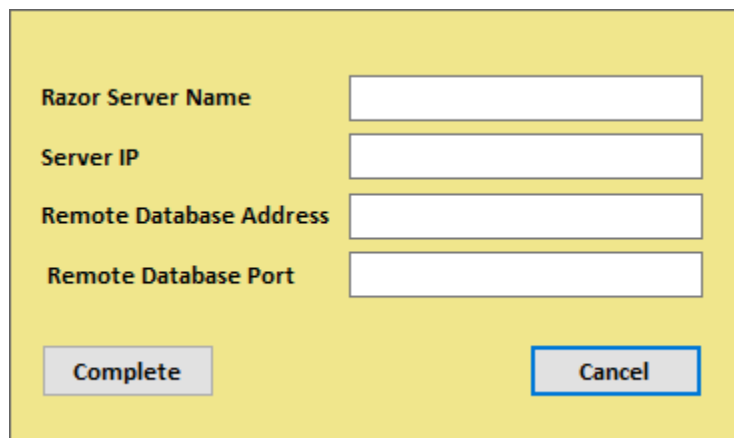
Direct Email

Cancel

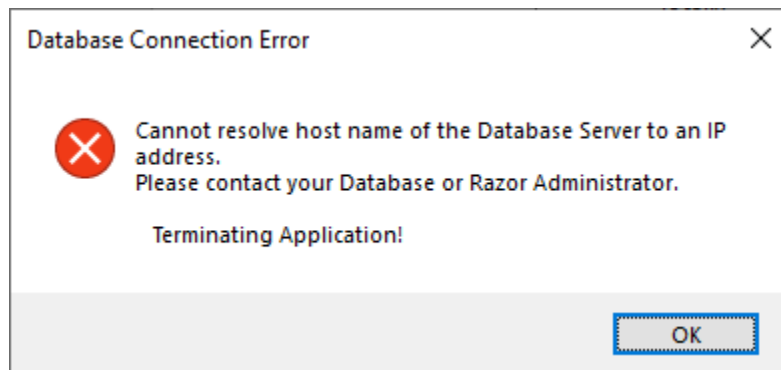
- d) Fill out the remainder of the form and send it off to the good folks at Visible Systems
- e) Once you have the quote and it is paid, then Visible Systems will send you a new set of license keys for Server A
- f) Using the Admin Tool select the Update Razor Licensing button and enter the new keys supplied.
- g) The Admin should refresh and display Enterprise License at the top and a new button should be enabled Remote Server Configuration
- h) Select this button and read the notice presented



- i) At this point the local Razor Admin needs the IP address of the remote server, the IP address of the remote database, if different from the remote server and the database port number and click on OK

A form with a yellow background. It has four input fields with labels: "Razor Server Name", "Server IP", "Remote Database Address", and "Remote Database Port". At the bottom, there are two buttons: "Complete" and "Cancel".

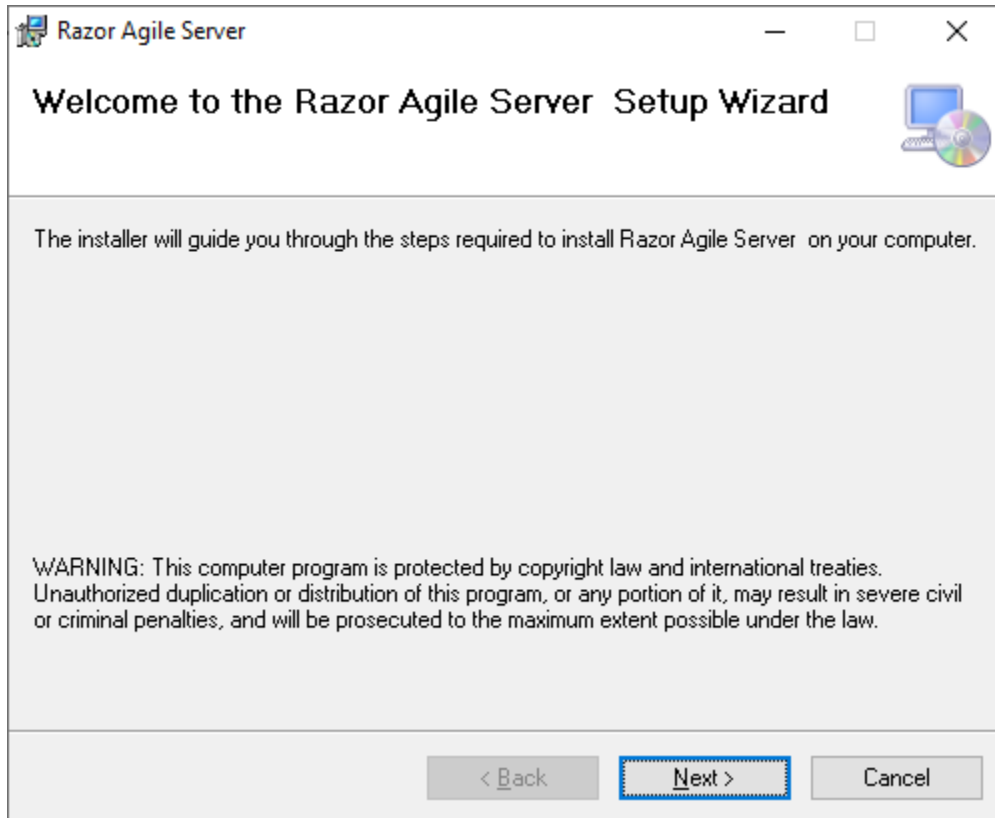
- j) This form is where the local Razor Admin enters the information on the remote site. The Razor Server Name can be any Text String that the local developers will understand that they are connecting to a remote site
- k) Next Click on complete
- l) If there is any problems with connecting to the remote server the following is presented.



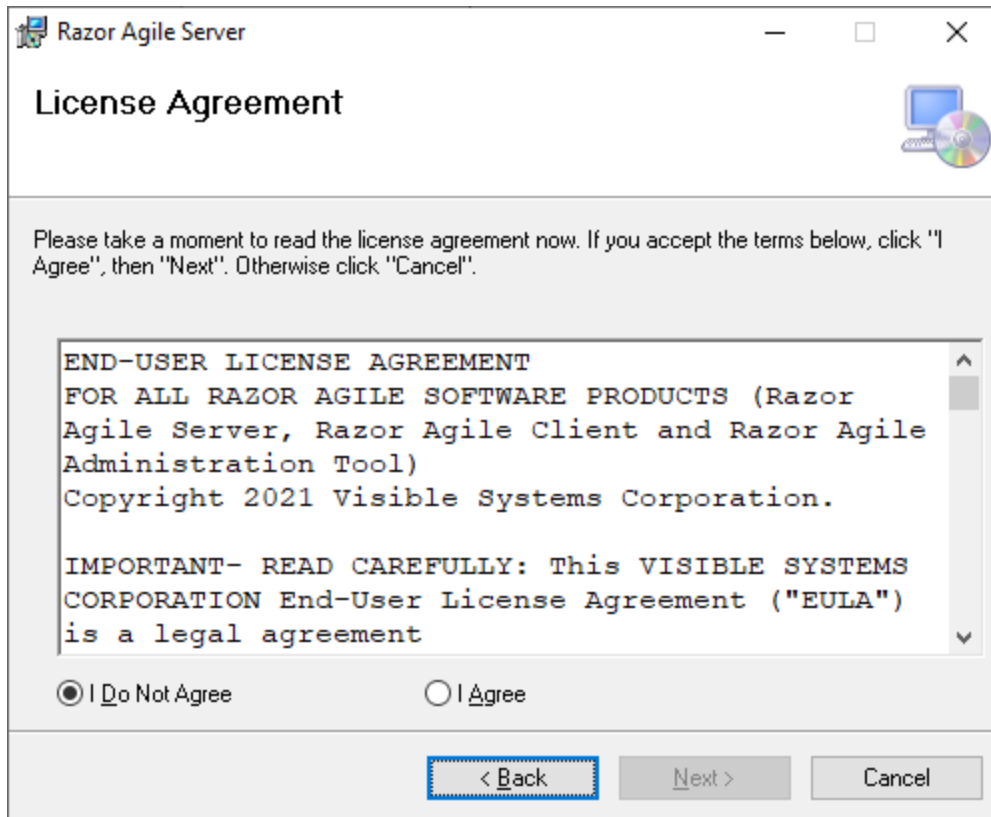
- m) At this point the local Razor Admin needs to ensure that all of the information that was entered is correct and that the remote server can be contacted from the local site
- n) Cautions When Using Remote Sites
The only real caution is for the developers . The need to be diligent and not have tools open for the remote site and the local site open at the same time. Razor Hybrid uses every tool it can to ensure that there is no cross talk between the 2 sets of tools. If a developer must be looking the local site and remote site Visible systems recommends that the developer have 2 clients open; one pointing to the remote site and the other pointing to the local site. It is no recommended to have one client with tools open to both sites.

Razor Server Installation

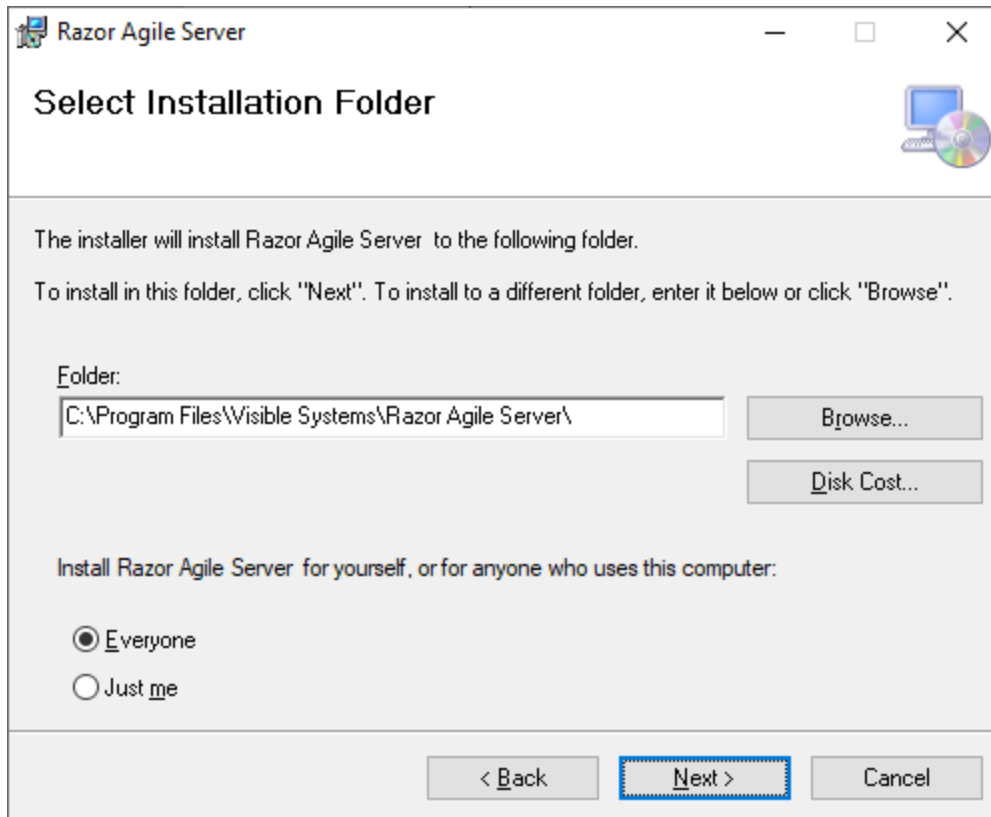
- 1) Navigate to where the files were downloaded to and double click on the setup.exe file



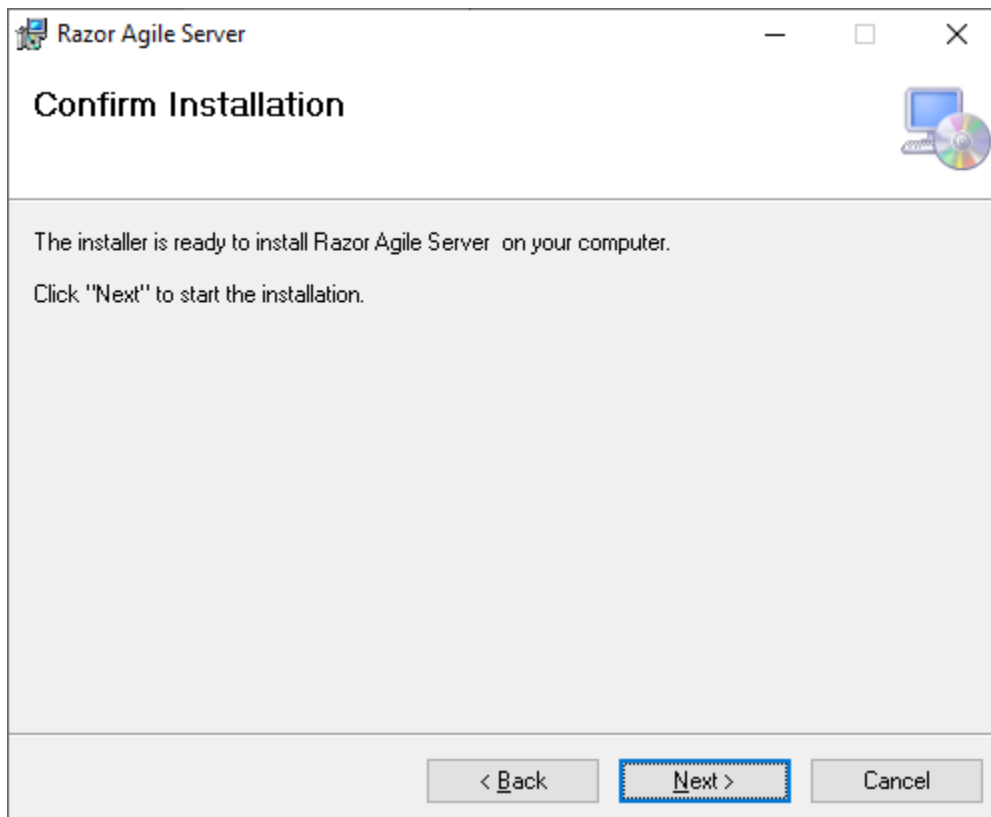
- 2) Click on Next



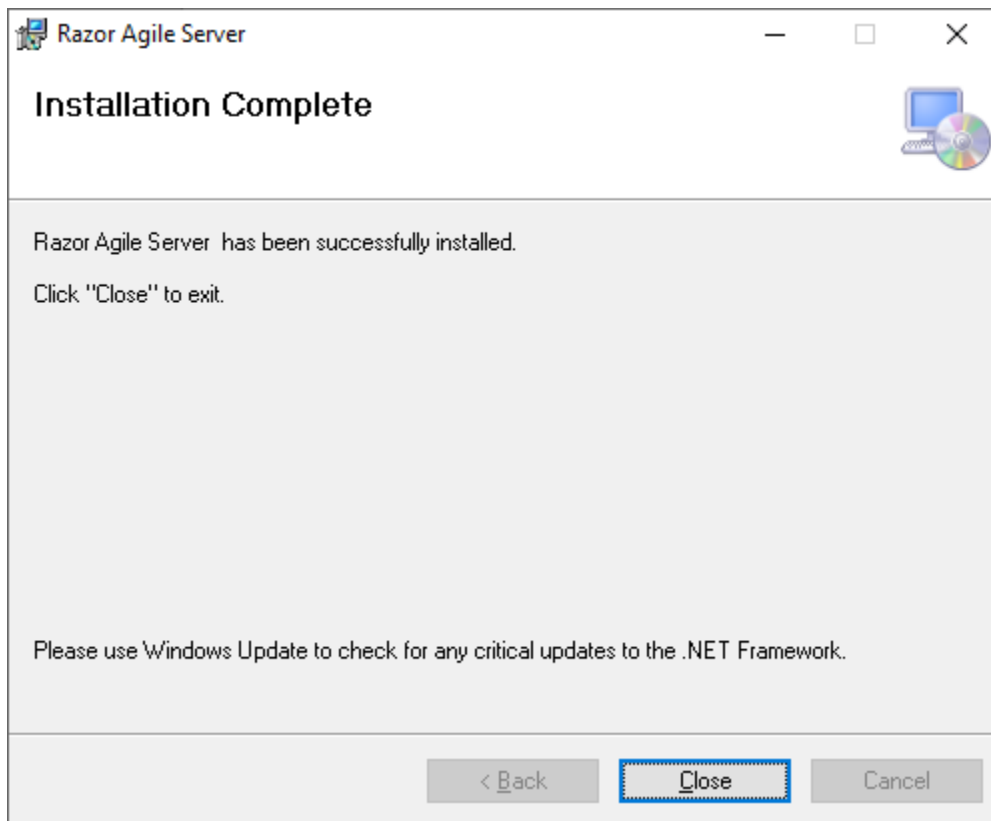
3) Read the License agreement select I Agree if you are OK with the license, Click Next



4) Select all the defaults. Note: DO NOT change the install directory. Click Next



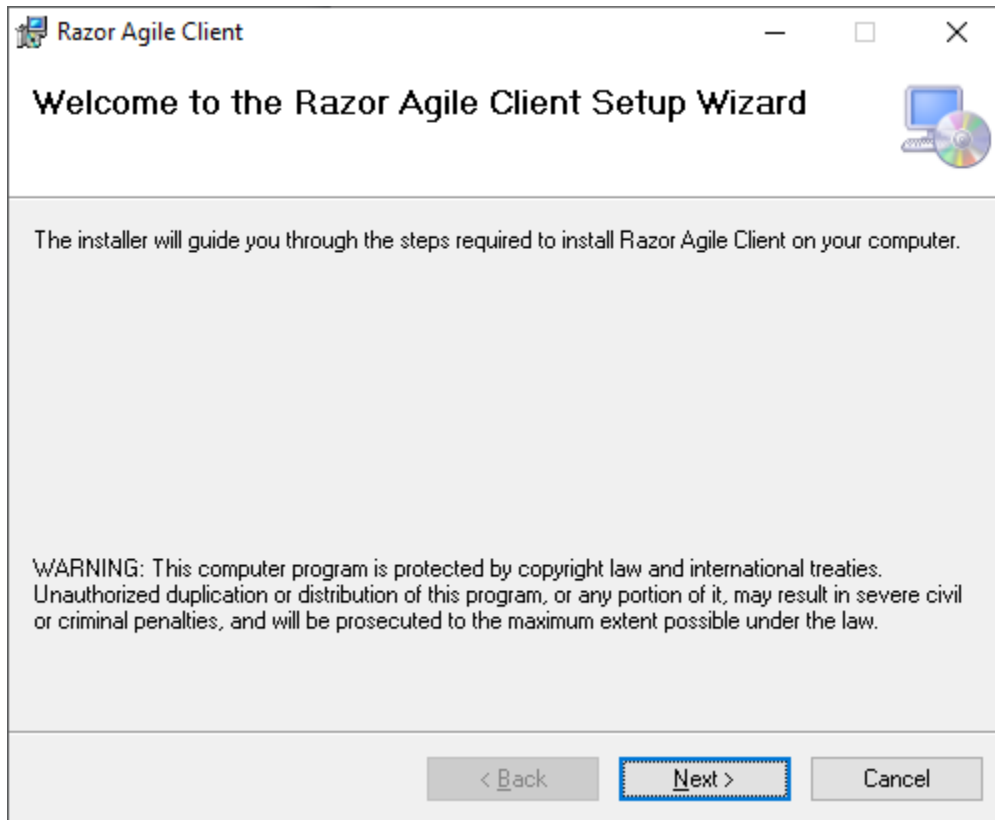
5) Click Next to begin the install process



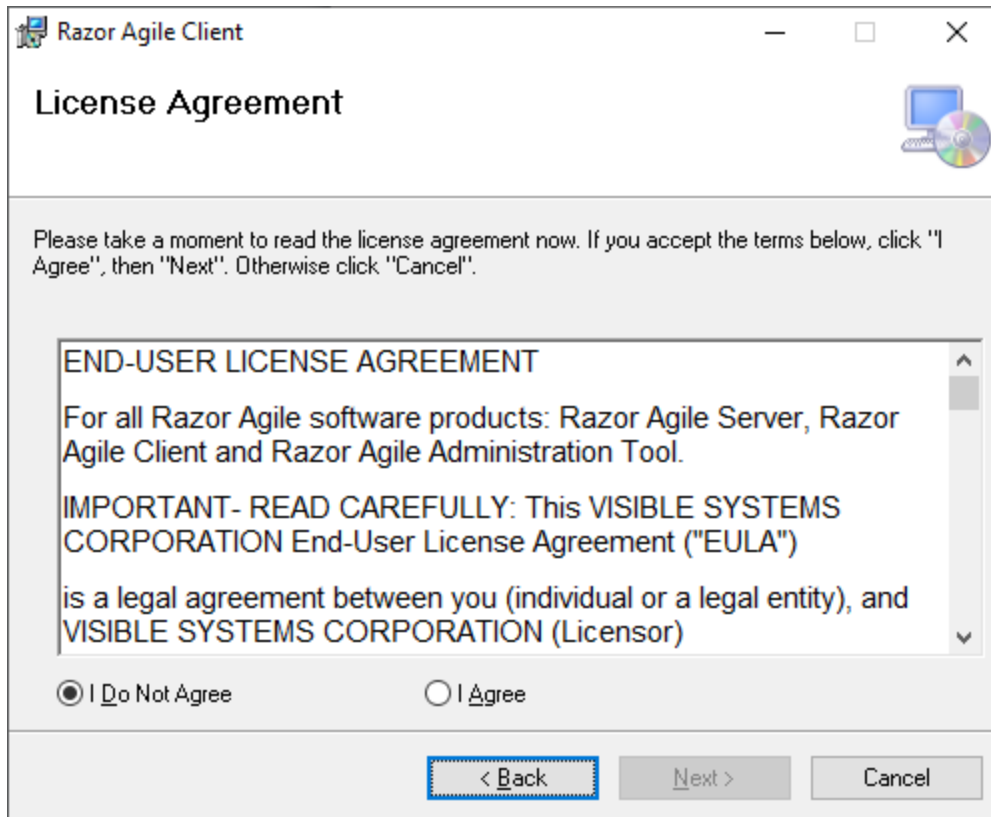
6) The server is now installed and started

Razor Client Installation

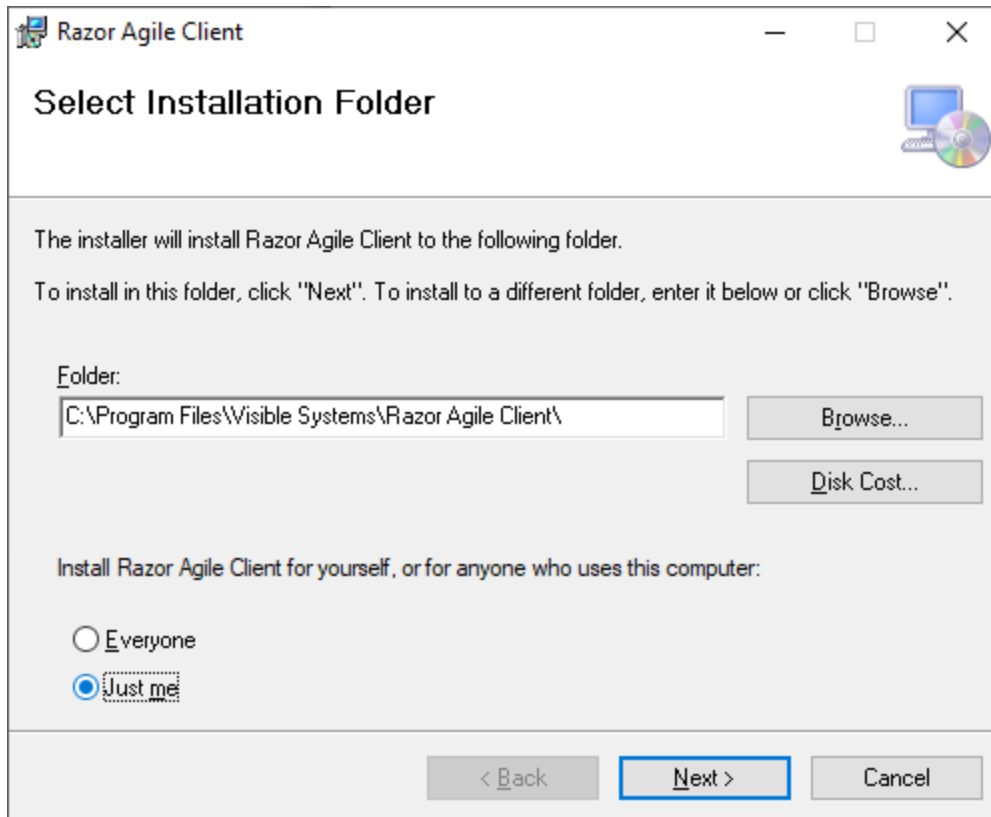
- 1) Navigate to where the files were downloaded to and double click on the setup.exe file



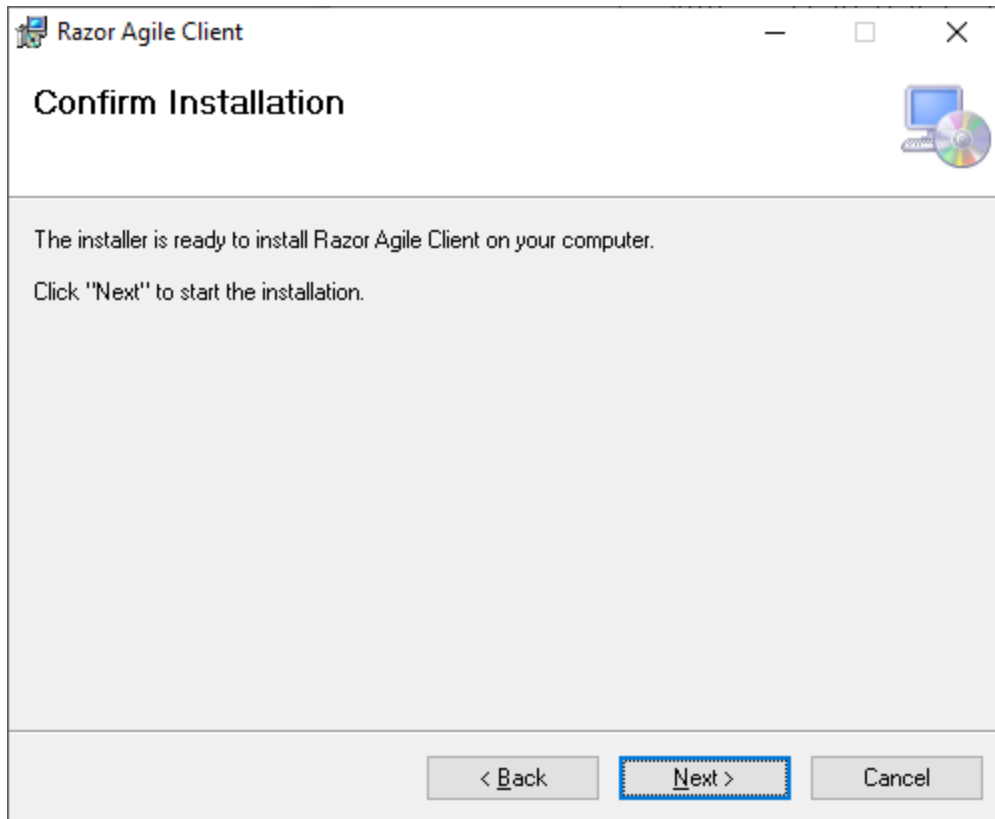
- 2) Click on Next



- 3) Read the License agreement select I Agree if you are OK with the license, Click Next



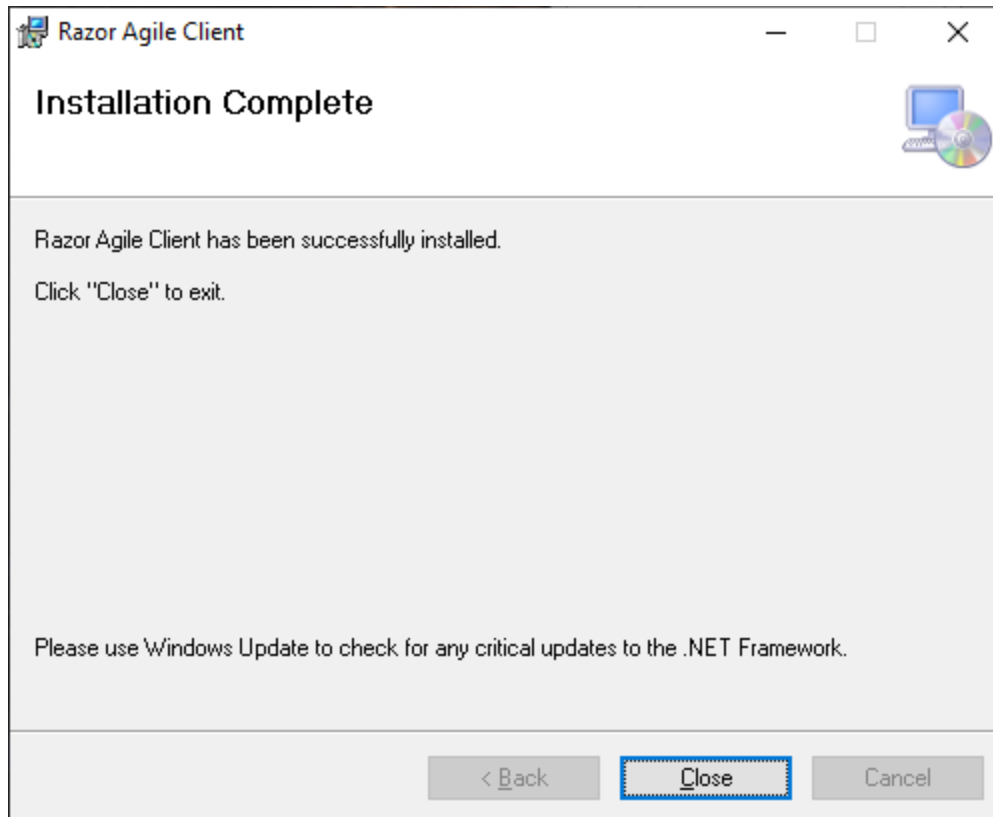
- 4) Select all the defaults. Note: DO NOT change the install directory. Click Next



5) Click Next to Continue

The screenshot shows a Windows-style dialog box titled "Razor Agile Client" with a subtitle "Razor6 Client Database Location". The dialog box contains a text prompt: "Enter the location of the Razor6 Database (or 127.0.0.1 if local) and the connect port number". Below this prompt are two text input fields. The first field is labeled "Database Server:" and contains the text "127.0.0.1". The second field is labeled "Database Connect Port Number:" and contains the text "3306". At the bottom of the dialog box are three buttons: "< Back", "Next >" (which is highlighted with a blue border), and "Cancel".

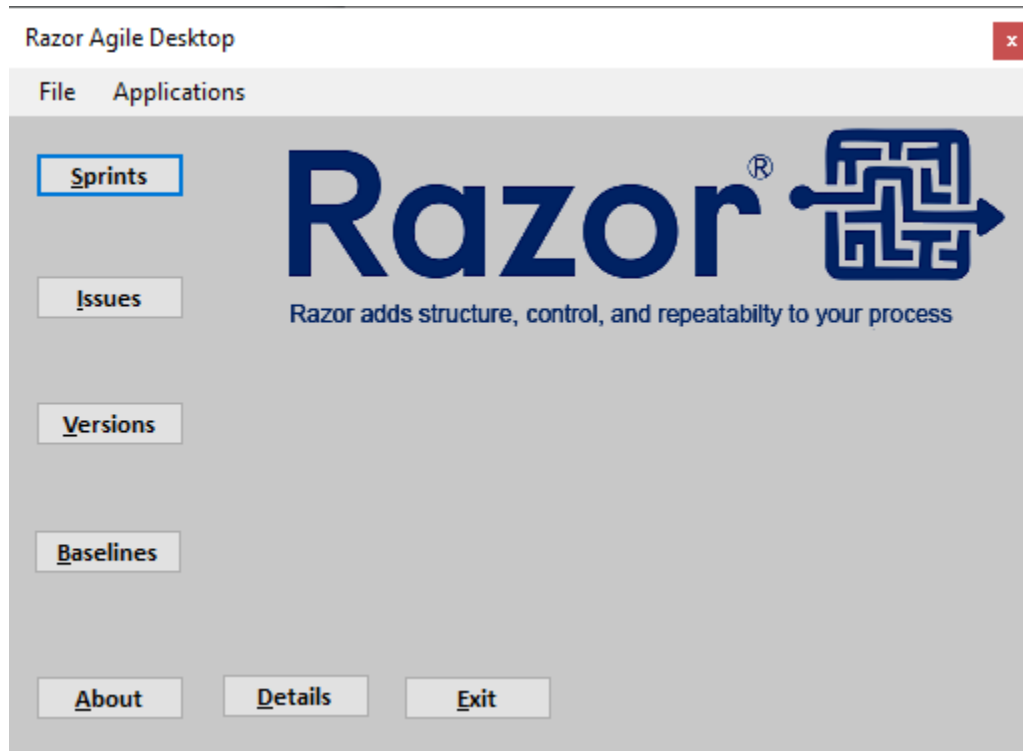
- 6) If you are not installing the client on the same server as the database as in the Test Drive configuration, then your Razor Administrator should have provided the information as to the Database server and Port number. Edit these fields as necessary and click Next.



7) Installation is now complete and there should be an Icon on your desktop for the client

Razor Hybrid Desktop

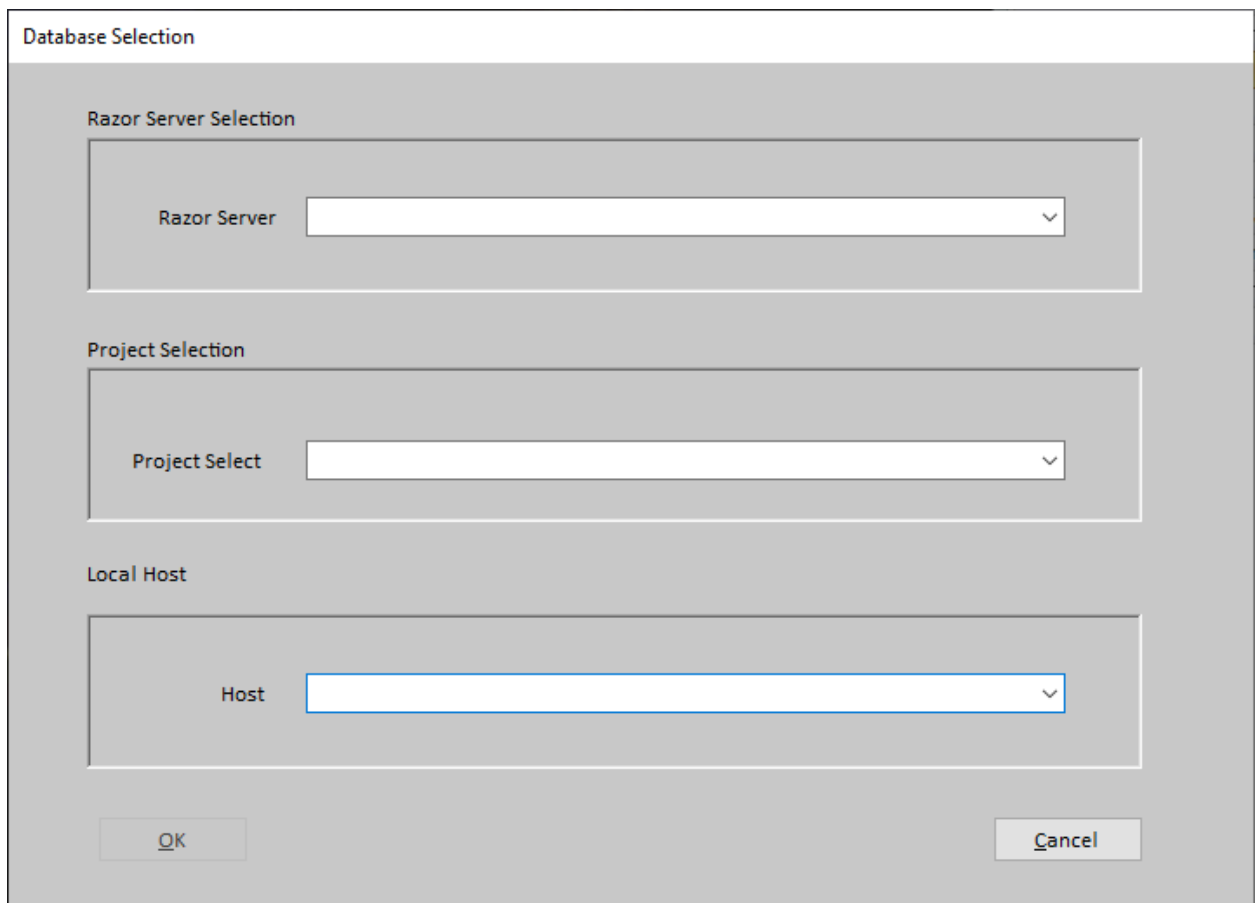
- 1) Double clicking on the Client Icon will bring up the Razor Hybrid Desktop



The Razor Hybrid Desktop is the gateway to all of the tools that are available to the user along the left hand side as well as the Applications pull down. The buttons along the bottom provide information on the release via the About button, details on the tools via the Details button and the Exit button exits the Desktop.

One can use the TAB key to navigate through the selections or press the ALT+ the Letter for the desired functions. Pressing the enter key will select the highlighted function.

- 1) Clicking on any of the tool's buttons will bring up the Database Selection form.



The screenshot shows a window titled "Database Selection". It contains three sections, each with a label and a dropdown menu:

- Razor Server Selection**: A dropdown menu labeled "Razor Server".
- Project Selection**: A dropdown menu labeled "Project Select".
- Local Host**: A dropdown menu labeled "Host".

At the bottom of the window, there are two buttons: "OK" on the left and "Cancel" on the right.

This tool allows for the selection of the Razor Server to contact in the event that your Razor administrator has set up more than one, the Project to connect to and presents all available local host IPs to use for commutations to the Razor Server. As before use of the TAB key and the arrow keys make the selections quick and easy.

2) Database is not available

In the case that the Client detects that the database is down or may have moved, the following screen may be displayed. Please read it carefully and contact your Razor Administrator to see if the Database is offline or has moved

Please Read the Following Carefully

The Razor Agile Client cannot contact the Database. This may be due to the Database having maintenance performed and is down or the Database has moved servers

OR

This is the first time you are running this Client. Then you must enter the Razor Database Server and Database Port number.

Please contact your Razor Administrator before Proceeding

Database Server (Fully qualified DNS Name)

Database Port Number

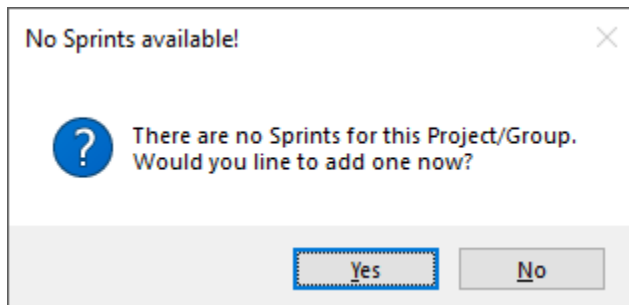
Sprints

1) Razor Hybrid Sprints

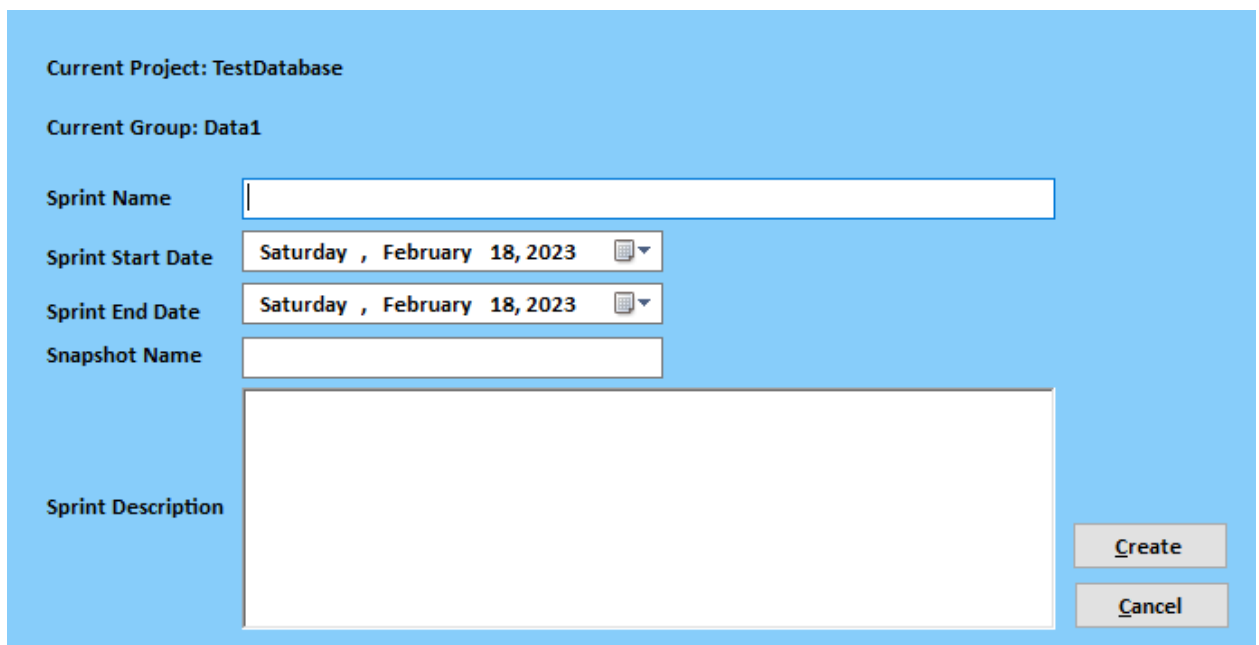
Razor Hybrid now has Superset to Issues that incorporates Hybrid Software development. This system uses Sprints and Stories to provide greater insight into the development process. It even incorporates rapid releases in the form when a story is completed, the Snapshot for that story is automatically updated with all the files worked on and associated to that story.

2) Start Working with Razor Hybrid Sprints

When selecting Sprints for the first time from the Razor Hybrid Desktop the following screen may be displayed



The default is yes and if yes is selected the following screen will be displayed



3) Creating your first Sprint

Current Project: TestDatabase

Current Group: Data1

Sprint Name

Sprint Start Date

Sprint End Date

Snapshot Name

Sprint Description

- a) Step one is to ensure that the correct Project and Group is displayed
- b) Next enter a Definitive Sprint Name. Note there is no checks on the Sprint name but it should be meaning full to the Project / Sprint
- c) Select a Sprint Start Date and End Date. Note the End Date has to be after the Start Date and is verified when the Create button is pressed
- d) The Snapshot Named field tiles this Sprint to and existing release or can create an entirely new release to the software. When clicked the following screen will appear

Copy, Use Existing or Create a New Snapshot

Existing Snapshots

New Snapshot

Snapshot Details

Purpose:

Snapshot State

☐ Development ☐ Integrated ☐ Tested

☐ Released

Brief Description (Optional)

4) Tying a Sprint to a Release

This part of the Sprint System ties any completed story in a sprint to a release in Razor Hybrid which then if desired can be automatically deployed via Scripting to production.

Copy, Use Existing or Create a New Snapshot

Existing Snapshots

New Snapshot

Copy Existing Snapshot

Use Existing Snapshot

Create New Snapshot

Snapshot Details

Purpose:

Snapshot State

☐ Development
☐ Integrated
☐ Tested
☐ Released

Brief Description (Optional)

Done

Cancel

- A) There are 3 choices that are available to the user
- i) Copy an existing snapshot. This system copies a snapshot that may be a full production release into a test release or a temporary release so the changes can be verified. It can then become the main line release if desired
 - ii) Use existing snapshot. This will update the current release snapshot of the product
 - iii) Create New Snapshot. This is used in the case where there are no snapshots to copy or use. Generally, this is a brand-new product.

Once all is complete the Main Sprint Page will look like this

Main Sprint Page

Current Project: BasicProject

Available Groups:

Testproject

Sprint Number	Sprint Status	Sprint Name	Sprint Start Date	Sprint End Date
1	New	Demo Sprint for Razor ...	6/13/2023 9:25:07 AM	7/17/2023 9:25:07 AM

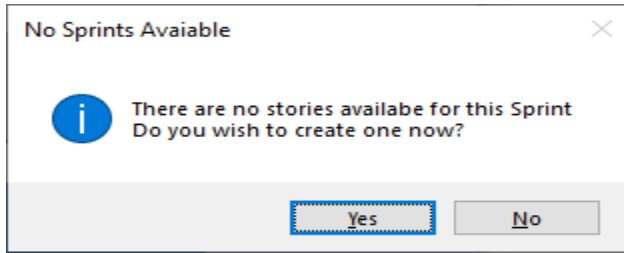
New Sprint

New Story

Exit

5) Creating the first Story

Double clicking on a new Sprint will bring up the following



Click on Yes and this screen will appear



Since Stories are Issues the Brief Description will be translated into the title of the Issue and the Detailed description will be translated into the Problem Statement

Once submitted the Story Board will show up and look a bit like this

Selected Sprint

Sprint Name: Demo Sprint for Razor Agile Manual

Sprint Start Date: 6/13/2023 9:25:07 AM

Sprint End Date: 7/17/2023 9:25:07 AM

Backlog	Story	Assigned	In Process	Completed
	Issue: Basic...2 First Story for Razor Manual Demo			

Maximize

Normal

Refresh

New Story

Exit

Double clicking on any story on the Story Board will bring up the associated Issue

Issue Number: Basic...-...2

☒ Review
 ☐ Modify
 Activity

Issue Overview

Title

Request Type
☐ Bug Report
 ☒ Enhancement
 ☐ Documentation

Priority
☐ Low
 ☐ Medium
 ☐ High
 ☒ Urgent

Submitters Information

Name	Mike C	Contact Number	() -
Email Address	mike@email.com	Extension	

Date Required

Issue State

☒ Submitted 2023/06/13,09:38:00 mikec122
☐ Assigned
☐ In-Process

Description of Problem

More Demo Testing but this time for Sprints

Actions Taken

At this time, Razor Hybrid does not have a role per say called a Scrum Master. However, one can be set up in the Roles and Permissions Tables which will be discussed in the Appendix on how to implement.

6) An Active Story Board Example

Selected Sprint

Sprint Name: Demo Sprint for Razor Agile Manual

Sprint Start Date: 6/13/2023 9:25:07 AM

Sprint End Date: 7/17/2023 9:25:07 AM

Backlog	Story	Assigned	In Process	Completed
	Issue: Basic...1 First Story for Razor Manual Demo			
			Issue: Basic...2 Second Story for Razor Manual Demo Assigned to: Mike C Due Date: Jun 15, 2023	
		Issue: Basic...3 Third Story for Razor Manual Demo Assigned to: Mike C Due Date: Jun 20, 2023		

Maximize

Normal

Refresh

New Story

Exit

Issues

1) Razor Hybrid Issues

Razor Hybrid's Issues functionality remains pretty much the same as Razor Classic's Issues. The main Issue form is completely redesigned to make selecting items quicker but the old methods for selecting items remain

Since Sprints are intergraded into Issues there is a selection to show Sprint based Issues only

The screenshot shows the 'Issues' application window. The title bar reads 'Issues Project: TestDatabase Group: Data1'. The menu bar includes 'File', 'View', 'Commands', 'Groups', 'Project Selection', and 'Help'. The main area is a table with columns: 'State', 'Priority', 'Attachments', 'Issue Number', and 'Title'. The table is currently empty. On the right side, there are several controls: 'Change Project' with a dropdown menu showing 'TestDatabase'; 'Change Group' with a dropdown menu showing 'Data1'; 'Display Issue' with a dropdown menu; a 'Display Issues' section with three radio buttons: 'All Issues' (selected), 'Assigned to me', and 'Sprints Only'; a checkbox for 'Show Grid Lines' which is unchecked; and three buttons: 'Refresh', 'New Issue', and 'Exit'.

2) Let's explore the Main Issue Display

These are descriptions of the righthand controls. Some of these like the Change Group and Change Project have the same functionality as the Groups and Project Selection on the Menu bar.

There are additional options with the View and Commands selections on the Menu Bar. However, the Command selections are only available if your Razor Administrator has set them up for you. These are usually scripts or other programs that provide additional functionality to the Issues System

The screenshot shows the 'Issues' application window. The title bar reads 'Issues Project: BasicProject Group: Testproject'. The menu bar includes 'File', 'View', 'Commands', 'Groups', 'Project Selection', and 'Help'. The main area contains a table with the following headers: 'State', 'Priority', 'Attachments', 'Issue Number', and 'Title'. The table is currently empty. To the right of the table, there are several green text annotations: 'You can change the project here ==>', 'You can switch to any group in a project here ==>', 'Display the selected Issue ==>', 'Select which Issues to display here ==>', 'Applies a grid to this display ==>', 'Refreshes the display. Good to use if up for long periods of time ==>', and 'Brings up a new Issue ==>'. The right-hand control panel includes a 'Change Project' dropdown menu set to 'BasicProject', a 'Change Group' dropdown menu set to 'Testproject', a 'Display Issue' dropdown menu, a 'Display Issues' section with three radio buttons ('All Issues' is selected, followed by 'Assigned to me' and 'Sprints Only'), a 'Show Grid Lines' checkbox, and three buttons: 'Refresh', 'New Issue', and 'Exit'.

When an Issue is submitted the form will look like this:

Issues Project: BasicProject Group: Testproject

File View Commands Groups Project Selection Help

State	Priority	Attachments	Issue Number	Title
Submitted	High	None	Basic...1	Demo Test 1 for Razor Agile Manual

Change Project
BasicProject

Change Group
Testproject

Display Issue
-

Display Issues
☒ All Issues
☐ Assigned to me
☐ Sprints Only

☐ Show Grid Lines

Refresh

New Issue

Exit

The State field shows that the Issue is Submitted with its Priority set to High. There are no attachments at this time and the full title is displayed.

3) The Issue Form

The Issue form that is included with Razor Hybrid is very basic but it is a good starting point for new users.

New Issue

☐ Review ☒ Modify

Print

Issue Overview

Title

Request Type

☐ Bug Report ☐ Enhancement ☐ Documentation

Priority

☐ Low ☐ Medium ☐ High ☐ Urgent

Submitters Information

Name

John Nash

Contact Number

() - -

Email Address

jnash@visible.com

Extension

Date Required

Monday , June 12, 2023

Issue State

☒ Submitted ☐ Rejected ☐ Approved ☐ Closed

Issue Assignment Information

Assigned To

Email Address

Scope of Work

☐ Software Update ☐ Add New Software Feature ☐ Software Quality Assurance

☐ Hardware Update ☐ Add New Hardware Feature ☐ Hardware Cerification

Description of Problem

Actions Taken

Submit

Apply Changes Only

Reset

Cancel

Help

4) State Driven Form Guide

The Issues form features what Visible Systems calls a state driven guide to filling out the form. If you notice in the form above, several fields are grayed out, but already filled out and some are open and have a yellow background color.

There is a table in the database that drives the field guide that is called the Insensitive table. This table along with the roles and permissions tables can have profound effects on how a user is guided in filling out the form. This can be used to ensure a consistent and repeatable process for describing a problem to the actual fix that was applied to the problem.

What this means as the states (Submitted ☐ Approved, or Submitted ☐ to Rejected) and so forth are changed, different areas of the form are opened up or closed off as needed by your process.

The setting up of these tables will be discussed in the appendix of this manual.

Here is an example of a submitted Issue set to the Approved State

Issue Number: Basic...-..1

☐ Review

☒ **Modify**

Activity

Attach

Print

Issue Overview

Title

Demo Test 1 for Razor Agile Manual

Request Type

☐ Bug Report

☒ Enhancement

☐ Documentation

Priority

☐ Low

☐ Medium

☒ High

☐ Urgent

Submitters Information

Name

John Nash

Contact Number

(123) 456-7890

Email Address

jnash@visible.com

Extension

Date Required

Tuesday , June 13, 2023

Issue State

☐ Submitted

2023/06/13,08:08:32 jnash122

☐ Rejected

☒ **Approved**

☐ Closed

Issue Assignment Information

Assigned To

Email Address

Scope of Work

☐ Software Update

☐ Add New Software Feature

☐ Software Quality Assurance

☐ Hardware Update

☐ Add New Hardware Feature

☐ Hardware Certification

Description of Problem

Demo for the Manual

Actions Taken

Submit

Apply Changes Only

Reset

Cancel

Help

67

Notice that new fields have been opened up and highlighted in Yellow. Also, the Activity and Attach buttons are now available and will be discussed below.

Once the form is filled out the submit button is selected and the database is then updated with the new information and the main display is refreshed.

5) Activity Button

When an Issue is set up to be required to interconnected with Versions and or Baselines the Activity button will provide insight as to what the issue was used for.

Activity for Issue: Basic....-1

File Operation	Developer	File Version	Date	Group	File Name
Introduce	John Nash	1.1	6/13/2023 11:12:23 AM	Testproject	/SampleAPIStuff/activityTest.sh
Introduce	John Nash	1.1	6/13/2023 11:12:23 AM	Testproject	/SampleAPIStuff/checkInFileTest.sh
Introduce	John Nash	1.1	6/13/2023 11:12:23 AM	Testproject	/SampleAPIStuff/checkOutFileTest.sh
Introduce	John Nash	1.1	6/13/2023 11:12:23 AM	Testproject	/SampleAPIStuff/checkOutReadOnlyTest.sh
Introduce	John Nash	1.1	6/13/2023 11:12:23 AM	Testproject	/SampleAPIStuff/compareSnapshots.sh
Introduce	John Nash	1.1	6/13/2023 11:12:24 AM	Testproject	/SampleAPIStuff/fileUnCheck.sh
Introduce	John Nash	1.1	6/13/2023 11:12:24 AM	Testproject	/SampleAPIStuff/getFieldData.sh
Introduce	John Nash	1.1	6/13/2023 11:12:24 AM	Testproject	/SampleAPIStuff/getFilesTest.sh
Introduce	John Nash	1.1	6/13/2023 11:12:24 AM	Testproject	/SampleAPIStuff/getFoldersTest.sh
Introduce	John Nash	1.1	6/13/2023 11:12:24 AM	Testproject	/SampleAPIStuff/issueSubmit.sh
Introduce	John Nash	1.1	6/13/2023 11:12:24 AM	Testproject	/SampleAPIStuff/newSnapshot.sh
Introduce	John Nash	1.1	6/13/2023 11:12:24 AM	Testproject	/SampleAPIStuff/promoteTest.sh
Introduce	John Nash	1.1	6/13/2023 11:12:25 AM	Testproject	/SampleAPIStuff/Prose.txt
Introduce	John Nash	1.1	6/13/2023 11:12:25 AM	Testproject	/SampleAPIStuff/testIssue.txt
Introduce	John Nash	1.1	6/13/2023 11:12:25 AM	Testproject	/SampleAPIStuff/updateExistingSnapshot.sh

Close

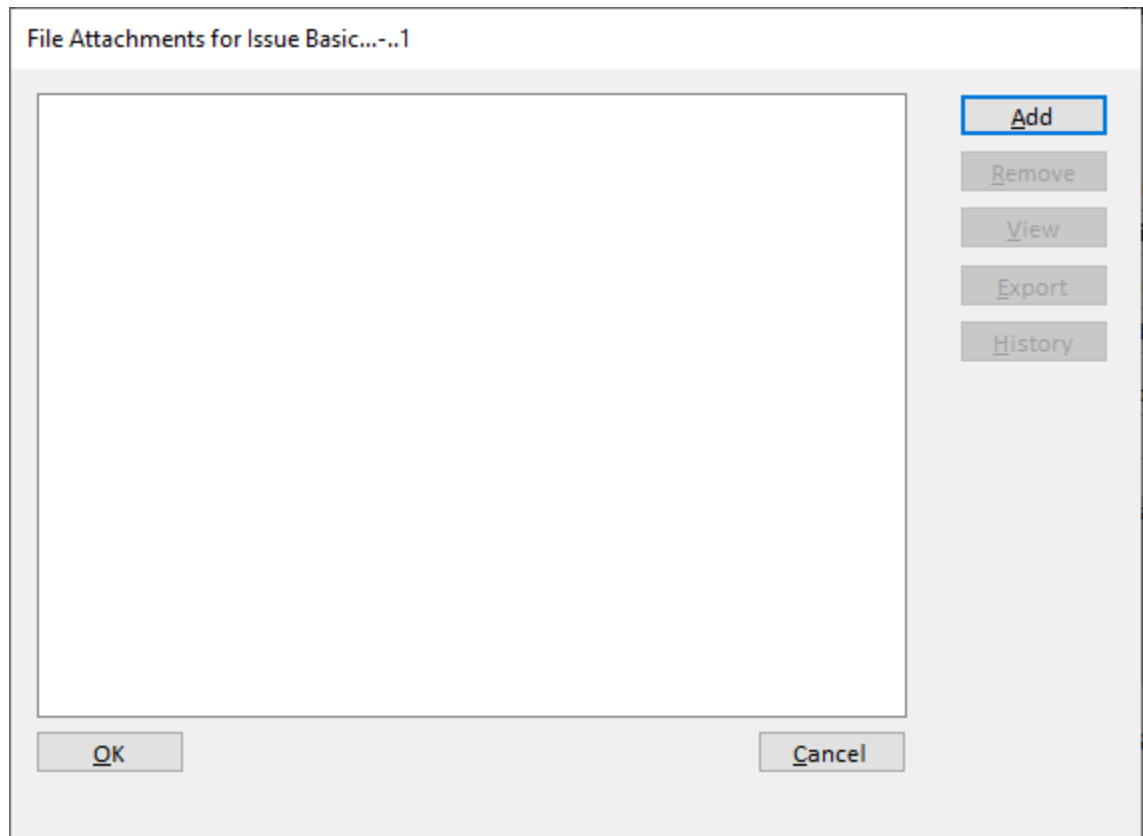
In this case this issue was used to Introduce files into the Versions system. It provides additional information as to Who and when the files were introduced and which files were introduced.

6) Adding Attachments

Sometimes a problem that an Issue must address is too complicated to be expressed in the Description of problem section or some other supporting documentation is required for final approval that all fixes have been properly applied before releasing the code to the public is approved. This is where attachments come in very handy.

- The Issue must be in the Approved State to add an Attachment

- b) Bring up the Issue and select the Modify Button at the top of the Issue and then select the Attach button
- c) The following screen is displayed



- d) Click on the add button and navigate to the file to add. It can be any file type for instance Word Doc, Excel spread sheet

File Attachments for Issue Basic...-..1

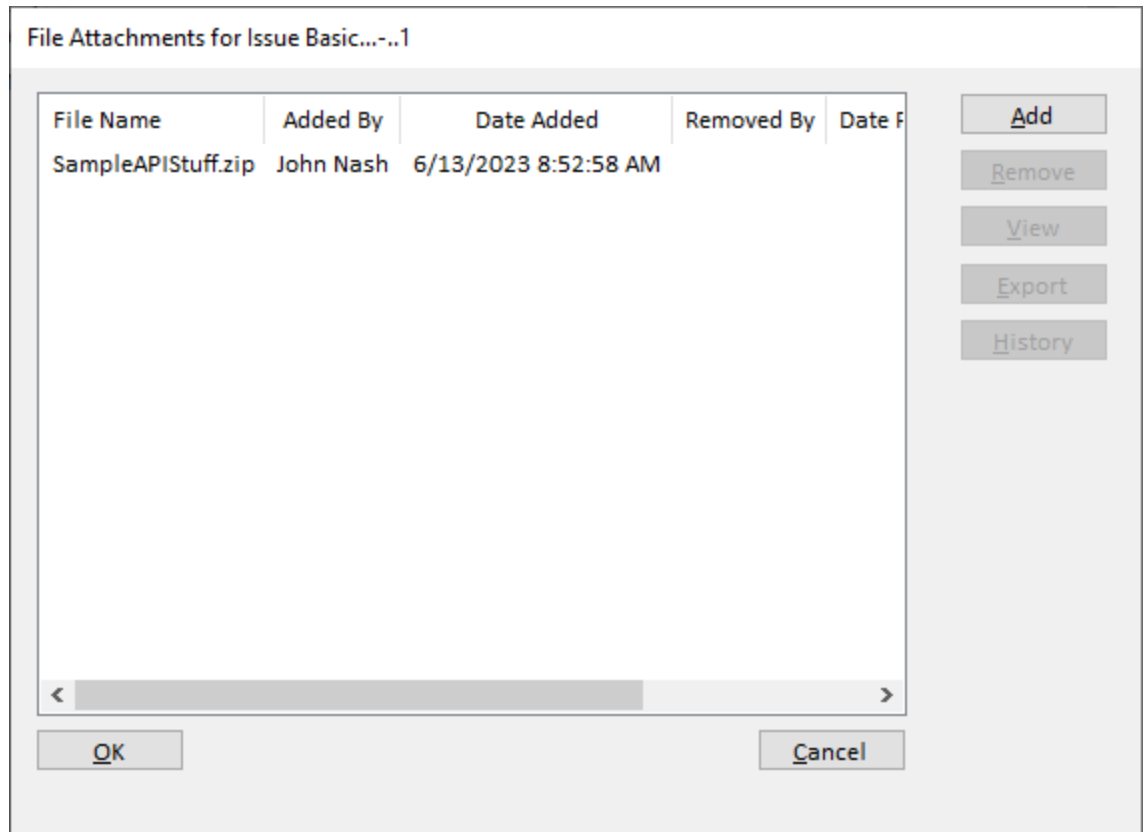
File Name	Added By	Date Added	Removed By	Date Rem
X:\SampleAPIStuff.zip	John Nash	*** PENDING ***		

< >

OK Cancel

Add
Remove
View
Export
History

- e) The state will remain pending until the OK button is selected. This is to allow for multiple files to be added.
- f) Once the OK button is pressed, the file is added



- g) Once the file is added clicking on the file will open up the options that will allow viewing the file, if viewable, removing the file, exporting the file or show the history of the attachment.

Note: Removing a file will not remove it from the display. It will be highlighted as removed but it's history will remain but can no longer be downloaded or viewed.

File Attachments for Issue Basic...-..1

File Name	Added By	Date Added	Removed By	Date F
SampleAPIStuff.zip	John Nash	6/13/2023 8:52:58 AM		

<

>

OK

Cancel

Add

Remove

View

Export

History

Versions

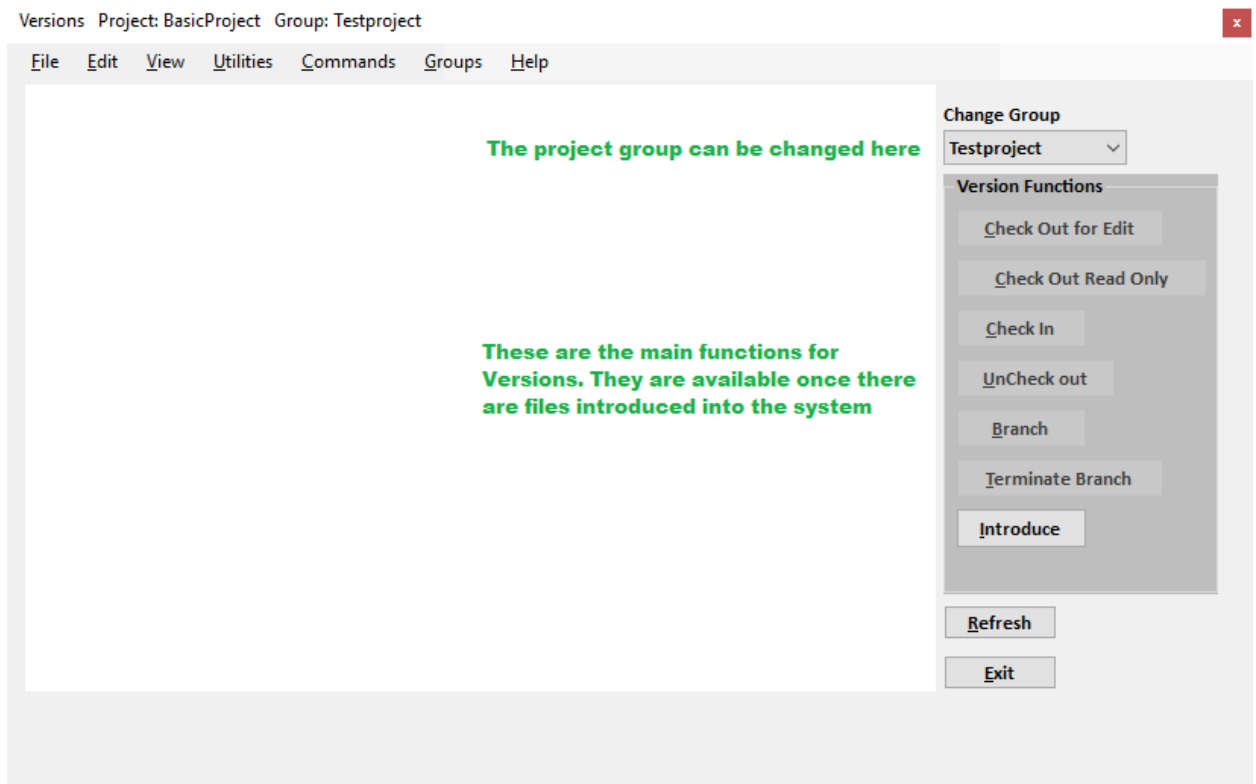
1) Razor Hybrid Versions

The versions program is the backbone of the configuration management aspect of the Razor tool suite. Versions provides a straightforward mechanism for controlling, monitoring, and archiving changes made to files in your development environment.

2) The Main Display of Versions

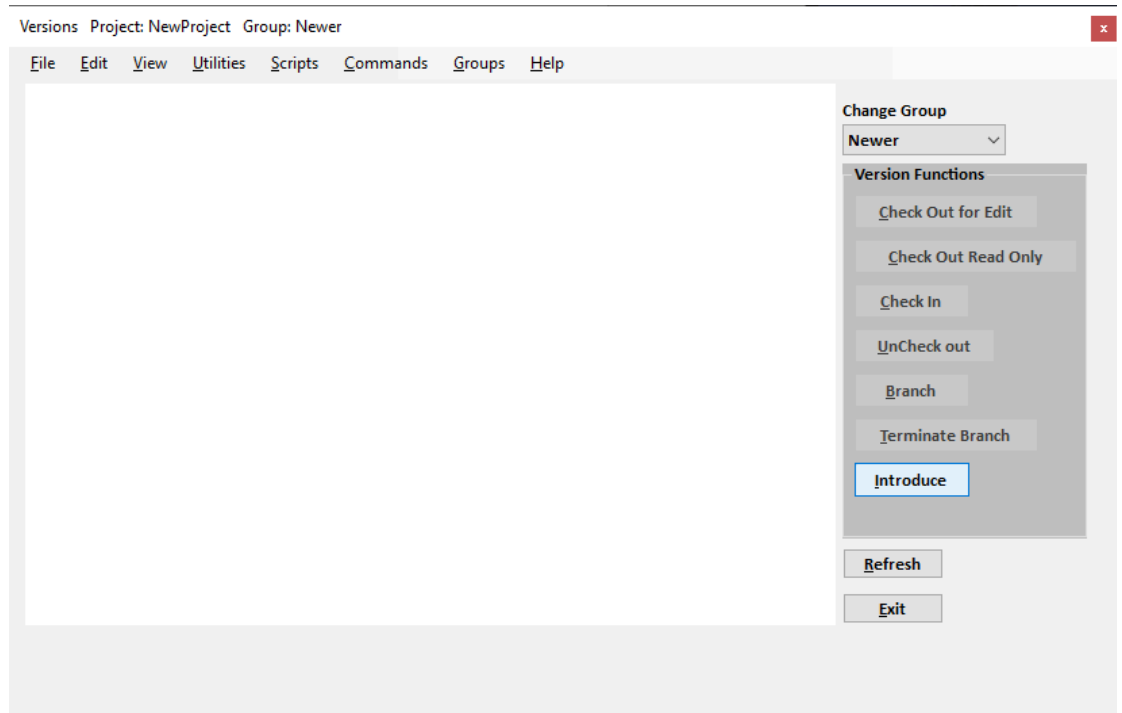
These are descriptions of the righthand controls. Some of these like the Change Group have the same functionality as the Groups Selection on the Menu bar.

There are additional options with the View and Commands selections on the Menu Bar. However, the Command selections are only available if your Razor Administrator has set them up for you. These are usually scripts or other programs that provide additional functionality to the Versions System

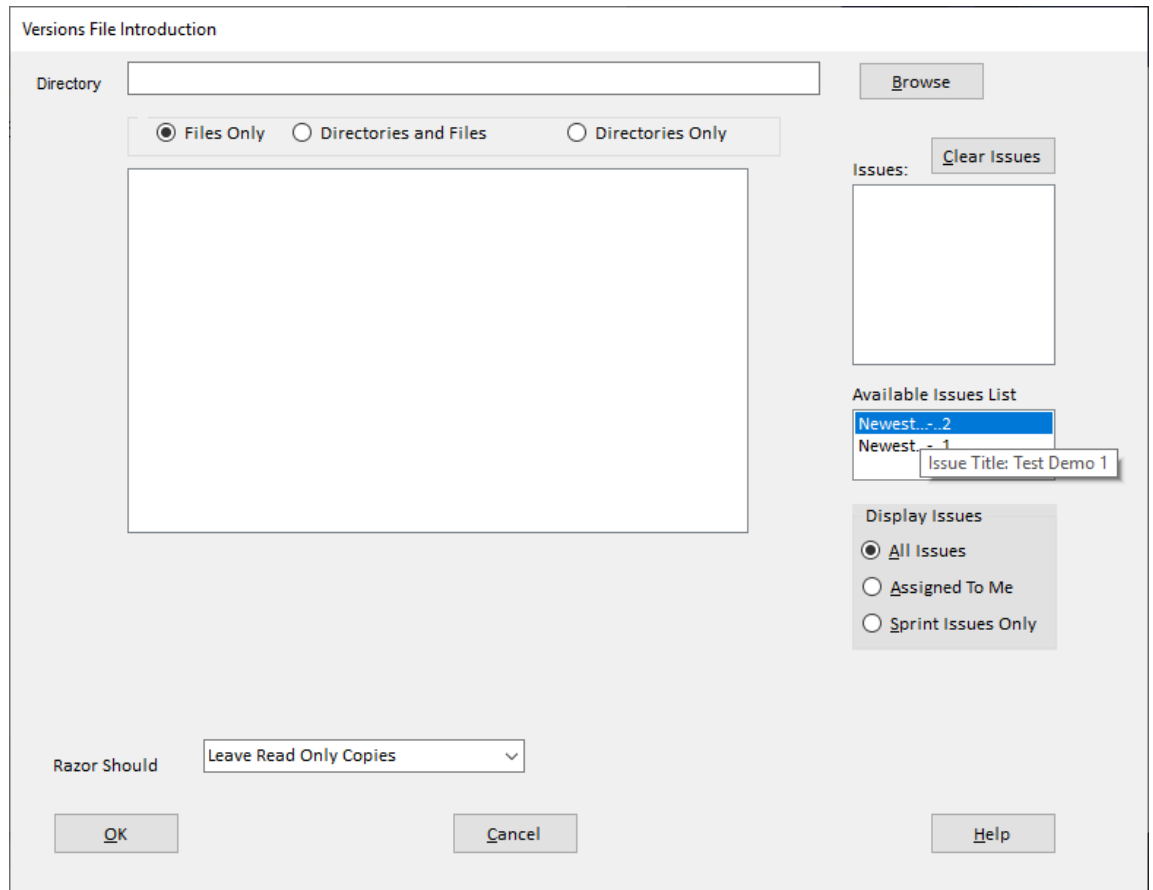


3) File Introduction

- A) There are 2 methods for file introduction: Razor Classic and Drag and Drop
- B) Both have similar Interfaces which have been designed to be easier to use
 - a. For Classic File Introduction select the Introduce button from the Versions Window



- b. On the right side of the Introduce window, in the box labeled available issues list you will have a list of the current Issues. By left clicking on an Issue the Issue title will appear



- c. By right clicking on the selected Issue, it is transferred to the Issues list to be used for this file introduction. One can use the standard Windows feature of Alt clicking or CTL clicking to select one or many issues to be included

Versions File Introduction

Directory

☒ Files Only ☐ Directories and Files ☐ Directories Only

Issues: Newest...2

Available Issues List
Newest...2
Newest...1

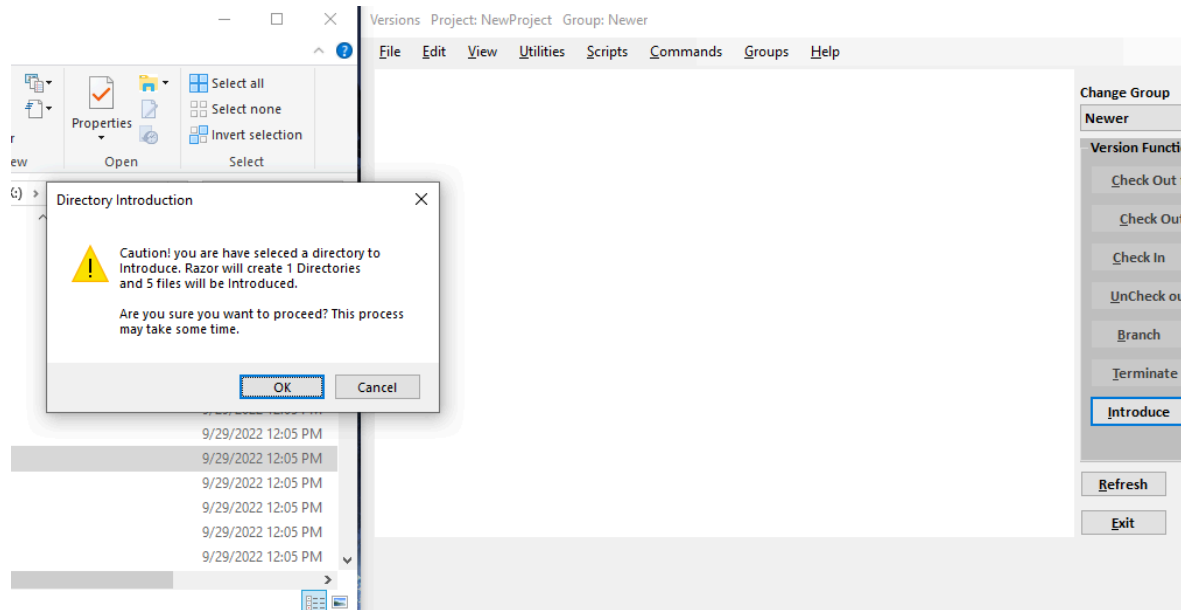
Display Issues
☒ All Issues
☐ Assigned To Me
☐ Sprint Issues Only

Razor Should

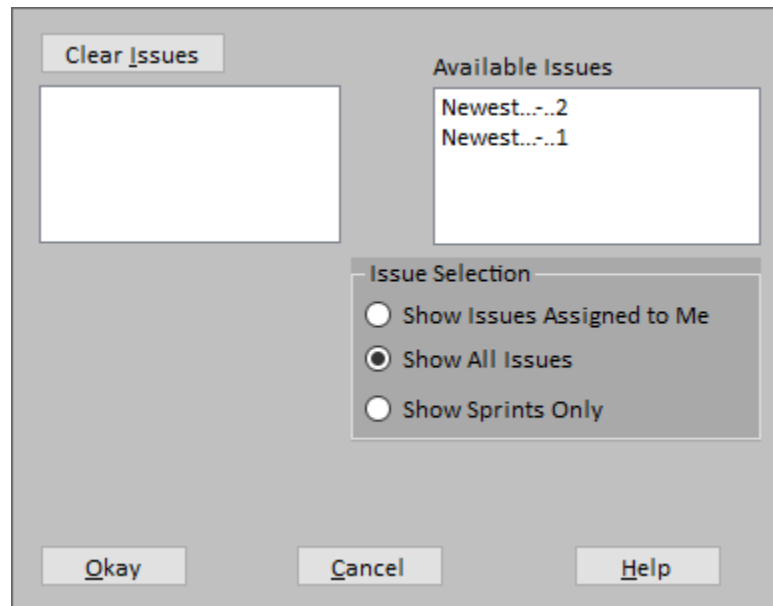
d. You can now proceed to complete the file introduction as needed

2) Drag and Drop File introduction

- a. Is basically the same as the standard file introduction
- b. Simply drag the file or folder that you wish to introduce onto the main Versions Window

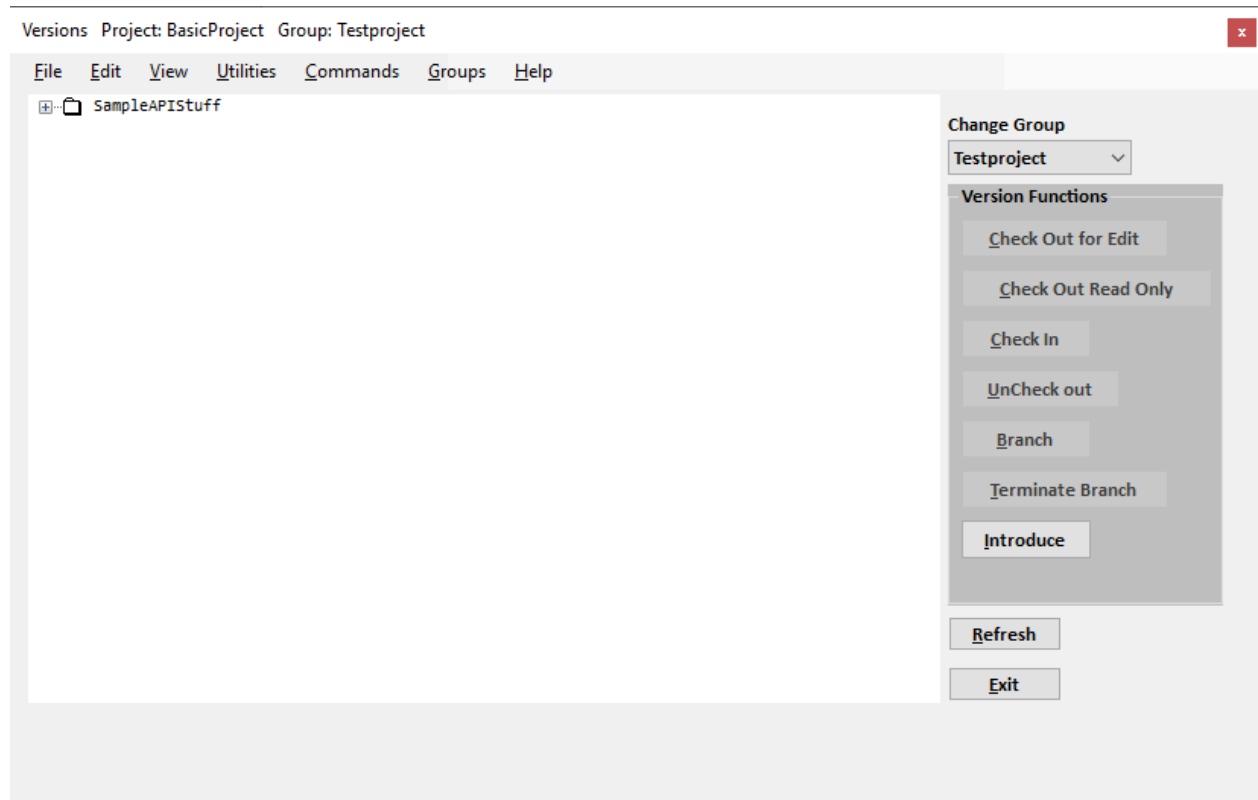


- c. Click on the OK button and the Issues selection Window will pop up

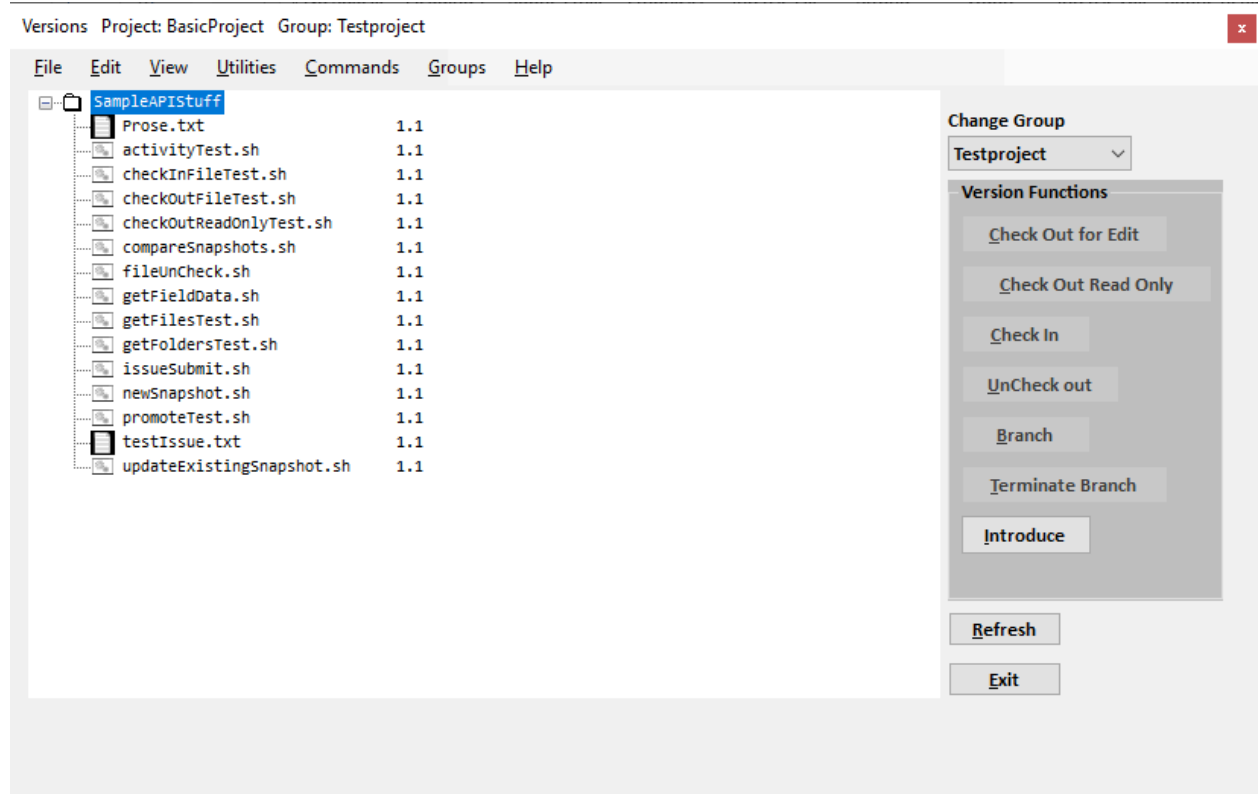


- d. Select the issues as needed and right click to include them and Click Okay

4) The Versions Screen After Introducing Files



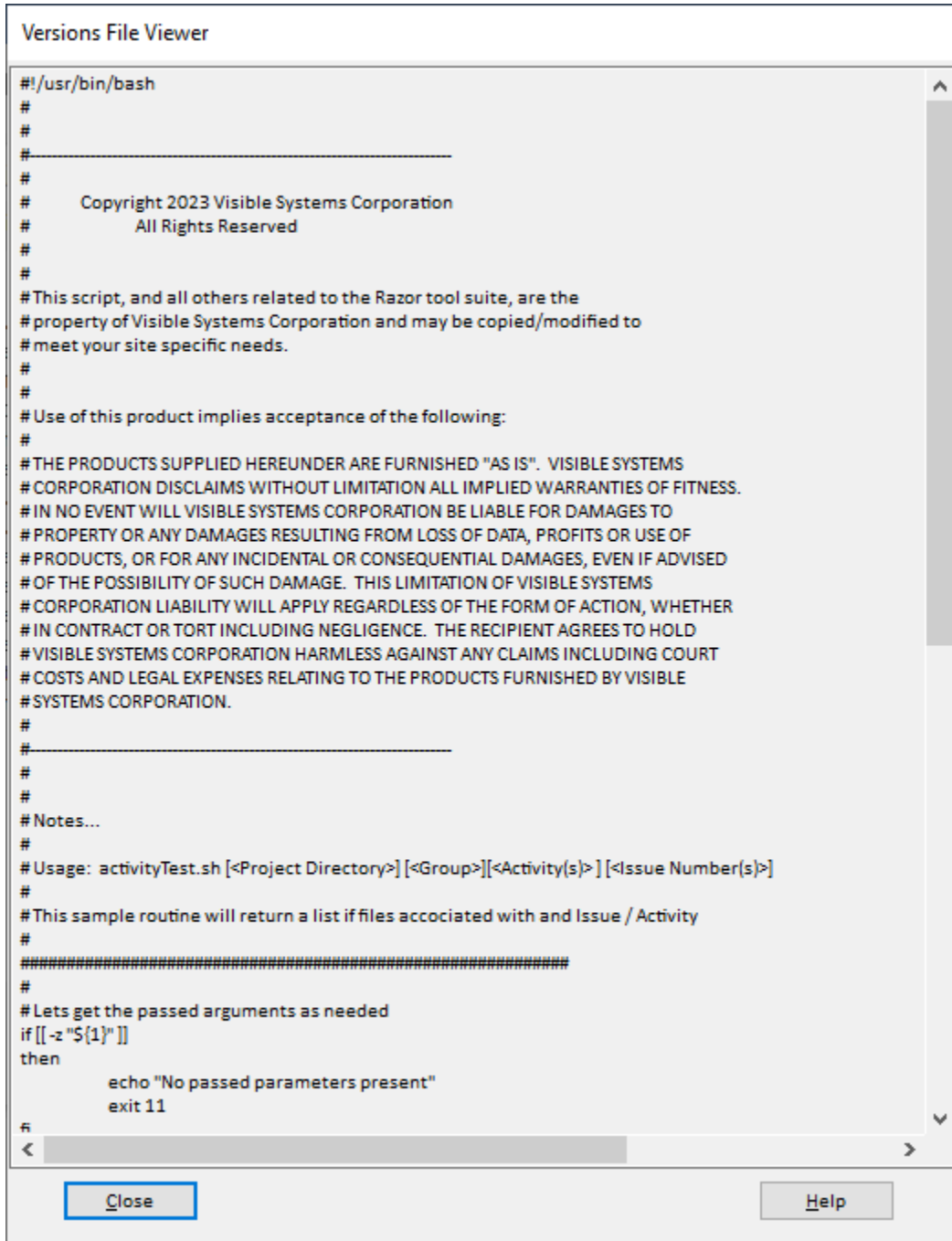
As one can observe, the directory is now showing up on the Versions Screen. By Clicking on the plus sign, the directory is opened and all the files are present.



Razor attempts to learn the file icon that is associated with the file type. In most cases it represents the file type quite accurately. One will also note that each of the introduced files are assigned a revision code 1.1. Even though the file is “introduced” into the Git repository, Razor Hybrid tracks the file revision / hash code for each file for a more human readable format. The importance of this will become more apparent as you read on.

5) Basic File Operations after Introduction

By double clicking on any text-based file a read-only display is presented of that file's contents. Note if the file is a binary file a warning message is displayed, but not the file contents.



```
#!/usr/bin/bash
#
#
#-----
#
#       Copyright 2023 Visible Systems Corporation
#             All Rights Reserved
#
#
# This script, and all others related to the Razor tool suite, are the
# property of Visible Systems Corporation and may be copied/modified to
# meet your site specific needs.
#
#
# Use of this product implies acceptance of the following:
#
# THE PRODUCTS SUPPLIED HEREUNDER ARE FURNISHED "AS IS". VISIBLE SYSTEMS
# CORPORATION DISCLAIMS WITHOUT LIMITATION ALL IMPLIED WARRANTIES OF FITNESS.
# IN NO EVENT WILL VISIBLE SYSTEMS CORPORATION BE LIABLE FOR DAMAGES TO
# PROPERTY OR ANY DAMAGES RESULTING FROM LOSS OF DATA, PROFITS OR USE OF
# PRODUCTS, OR FOR ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED
# OF THE POSSIBILITY OF SUCH DAMAGE. THIS LIMITATION OF VISIBLE SYSTEMS
# CORPORATION LIABILITY WILL APPLY REGARDLESS OF THE FORM OF ACTION, WHETHER
# IN CONTRACT OR TORT INCLUDING NEGLIGENCE. THE RECIPIENT AGREES TO HOLD
# VISIBLE SYSTEMS CORPORATION HARMLESS AGAINST ANY CLAIMS INCLUDING COURT
# COSTS AND LEGAL EXPENSES RELATING TO THE PRODUCTS FURNISHED BY VISIBLE
# SYSTEMS CORPORATION.
#
#-----
#
# Notes...
#
# Usage: activityTest.sh [<Project Directory>] [<Group>][<Activity(s)>] [<Issue Number(s)>]
#
# This sample routine will return a list if files accociated with and Issue / Activity
#
#-----
#
# Lets get the passed arguments as needed
if [[ -z "${1}" ]]
then
    echo "No passed parameters present"
    exit 11
fi
```

Also, by selecting a file and then selecting Utilities ☐ Info the following is displayed

File Information for: SampleAPIStuff/activityTest.sh

Current Status

Latest Version: 1.1
Last Modified: 6/13/2023 11:12:23 AM
Time Introduced: 6/13/2023 11:12:23 AM

File History

Issue Number: Basic...-1
Group Name: Testproject
File Operation: Introduce
Introduce to: SampleAPIStuff/activityTest.sh
By Who: John Nash
File Version: 1.1
File State: Released
Date: 6/13/2023 11:12:23 AM

Cancel

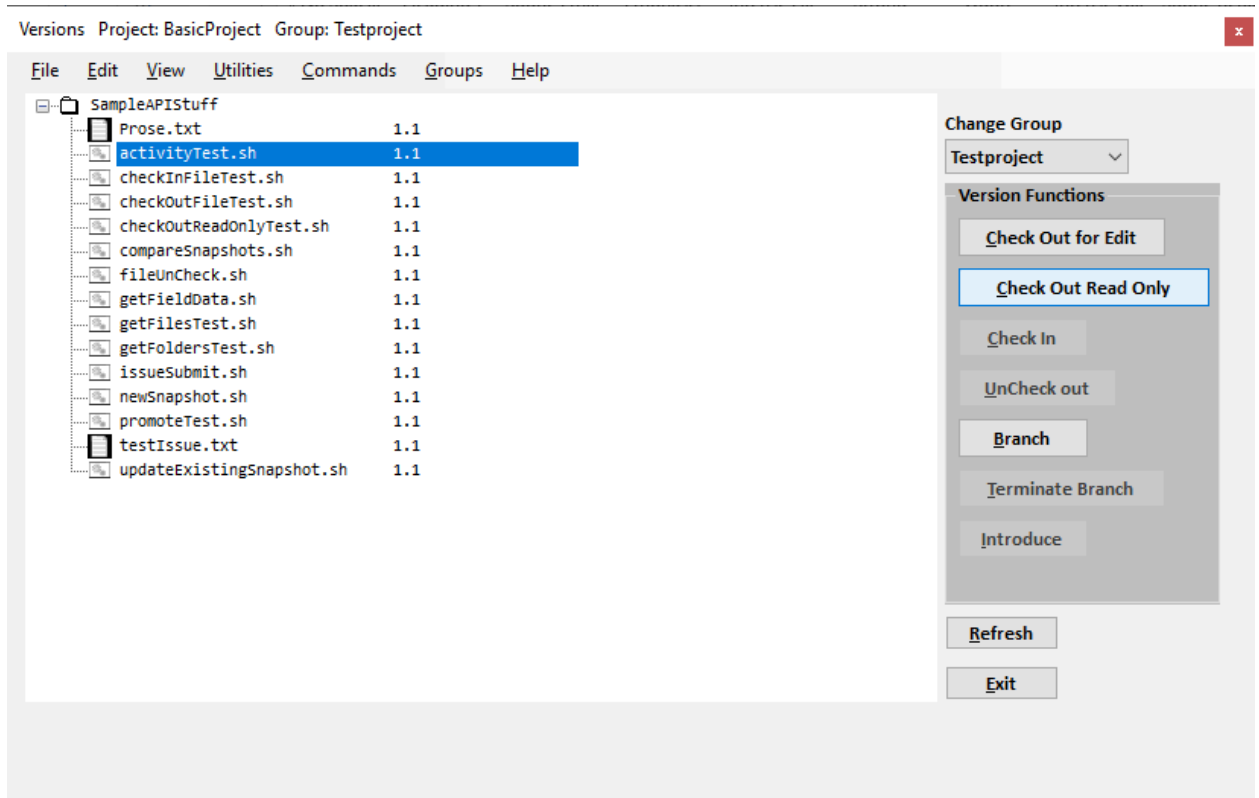
The file history will show in detail every action that has taken place on that file. When and who checked the file out for edit or read-only etc.

The Utilities ☐ File Diff will be discussed below.

4) File Check Out Read Only

There are times where a developer may need to get a copy of a file for use in another project or just want to get a print out a better view of the file.

Simply select the file and then select the Check Out Rad Only button



And the following screen is displayed

Checkout ReadOnly

Files for Checkout Read Only:

SampleAPIStuff/activityTest.sh

1 File(s) in List

☐ Path is absolute Browse

Check Out to: M:\

Available Versions: ▼

OK Cancel

The file name and directory are displayed as shown. The browse button is used to select where the file is to be written. In this case M:\

If the Path is absolute is checked the directory structure for the file is removed. The file location is in this case M:\ActivityTest.sh

The Available Versions pull down is unique. If left as show above, the latest version of the file is received from the Git repository, However, if there are many revisions on the file i.e., 1.1, 1.2, 1.3 then selecting one of those revisions will get that file version from the Git repository.

5) Check Out File for Edit

Now the work begins. An Issue or Story from a Sprit has been assigned to you. Time to get the files needed to begin work. The process is started by selecting the file or files needed and clicking on Check Out for Edit Button

File Check-Out

File(s) to check out:

SampleAPIStuff/activityTest.sh, <== Files selected

Issues: [Clear Issues](#)

Available Issues List

Basic...2
Basic...1

File(s) in List: 1

One can narrow down the available Issues list here

Display Issues

☒ Show All Issues
☐ Assigned to Me
☐ Sprint Issues Only

☐ Path is absolute [Browse](#)

Check-Out To:

Title: This field is mandatory

Description: This field is optional

[OK](#) [Clear](#) [Cancel](#) [Help](#)

- Left clicking on any Issue in the Available Issues List will display the Title for that Issue
- Right Clicking on any Issue in the Available Issue List will post that Issue into the Issues filed and associate that Issue to the file(s) Check out

- c) The Display Issue area allows for the filtering down of the Issues list. The default is to show all Issues. Selecting the Assigned to Me button will show all the Issues that are assigned to this person. Sprint Issues Only will provide a pop up that will allow the selection of the Sprint, then display Issues (Stories) assigned to that Sprint
- d) Path is absolute check box when selected will remove the entire directory structure from the files and deposit them directly to the checked out to destination
- e) The title field is mandatory and should be a brief description of what is being done
- f) Description field is optional. More details can be placed here

Once the form is filled out as shown below click on the OK button and the file will be checked out and assigned to the person doing the check out

File Check-Out

File(s) to check out:

SampleAPIStuff/activityTest.sh

Issues: [Clear Issues](#)

Basic...-...1

Available Issues List

Basic...-...2
Basic...-...1

File(s) in List: 1

Display Issues

☒ Show All Issues
☐ Assigned to Me
☐ Sprint Issues Only

☐ Path is absolute [Browse](#)

Check-Out To: M:\

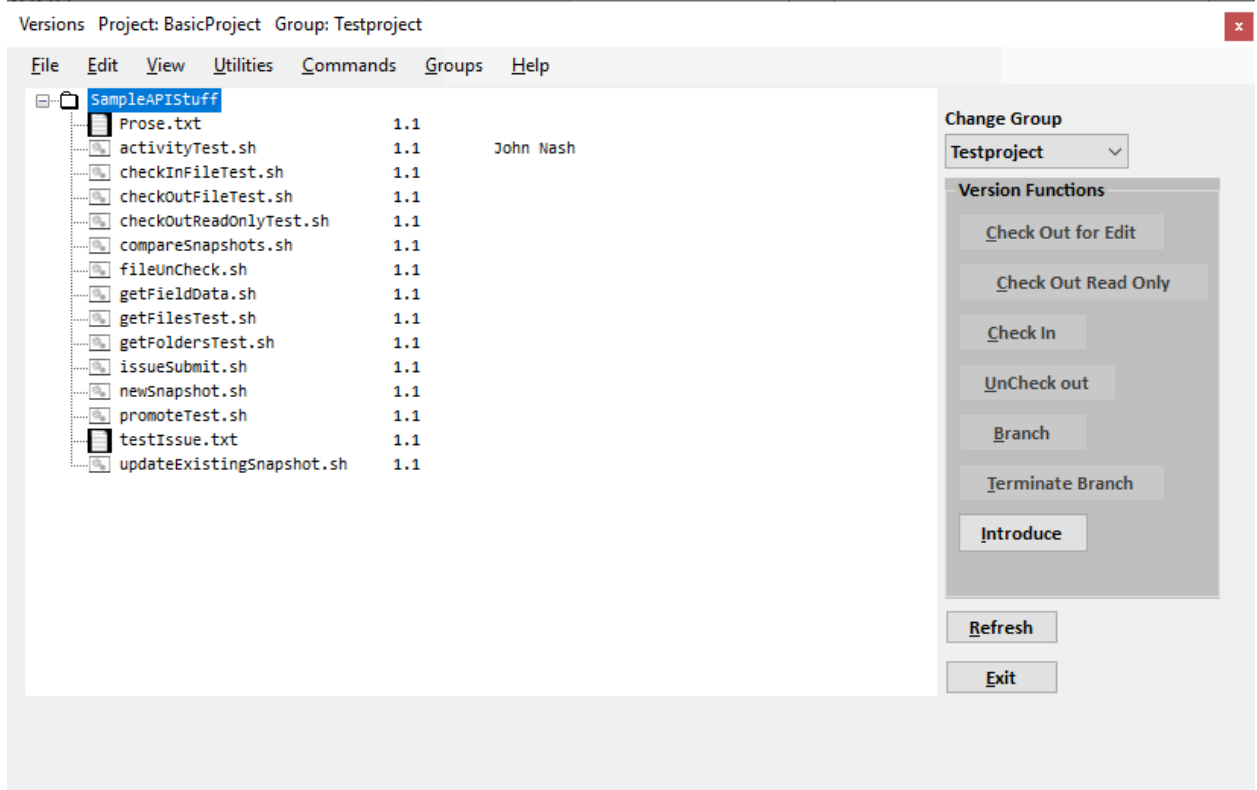
Title: Demo Test for File Checkout

Description:

Nothing new to report

[OK](#) [Clear](#) [Cancel](#) [Help](#)

Once the file is checked out the person who has the file checked out is displayed on the main versions screen



Now if you remember the Activity file, it now looks like this:

Activity for Issue: Basic...-1

File Operation	Developer	File Version	Date	Group	File Name
Introduce	John Nash	1.1	6/13/2023 11:12:23 AM	Testproject	/SampleAPIStuff/activityTest.sh
Introduce	John Nash	1.1	6/13/2023 11:12:23 AM	Testproject	/SampleAPIStuff/checkInFileTest.sh
Introduce	John Nash	1.1	6/13/2023 11:12:23 AM	Testproject	/SampleAPIStuff/checkOutFileTest.sh
Introduce	John Nash	1.1	6/13/2023 11:12:23 AM	Testproject	/SampleAPIStuff/checkOutReadOnlyTest.sh
Introduce	John Nash	1.1	6/13/2023 11:12:23 AM	Testproject	/SampleAPIStuff/compareSnapshots.sh
Introduce	John Nash	1.1	6/13/2023 11:12:24 AM	Testproject	/SampleAPIStuff/fileUnCheck.sh
Introduce	John Nash	1.1	6/13/2023 11:12:24 AM	Testproject	/SampleAPIStuff/getFieldData.sh
Introduce	John Nash	1.1	6/13/2023 11:12:24 AM	Testproject	/SampleAPIStuff/getFilesTest.sh
Introduce	John Nash	1.1	6/13/2023 11:12:24 AM	Testproject	/SampleAPIStuff/getFoldersTest.sh
Introduce	John Nash	1.1	6/13/2023 11:12:24 AM	Testproject	/SampleAPIStuff/issueSubmit.sh
Introduce	John Nash	1.1	6/13/2023 11:12:24 AM	Testproject	/SampleAPIStuff/newSnapshot.sh
Introduce	John Nash	1.1	6/13/2023 11:12:24 AM	Testproject	/SampleAPIStuff/promoteTest.sh
Introduce	John Nash	1.1	6/13/2023 11:12:25 AM	Testproject	/SampleAPIStuff/Prose.txt
Introduce	John Nash	1.1	6/13/2023 11:12:25 AM	Testproject	/SampleAPIStuff/testIssue.txt
Introduce	John Nash	1.1	6/13/2023 11:12:25 AM	Testproject	/SampleAPIStuff/updateExistingSnapshot.sh
Checked-Out	John Nash	1.1	6/15/2023 7:30:54 AM	Testproject	/SampleAPIStuff/activityTest.sh

Close

It shows that the file is checked out, by who and when.

Also remember the Info file, it now looks like this:

File Information for: SampleAPIStuff/activityTest.sh

Current Status

Latest Version: 1.1
Last Modified: 6/15/2023 7:30:54 AM
Time Introduced: 6/13/2023 11:12:23 AM

File History

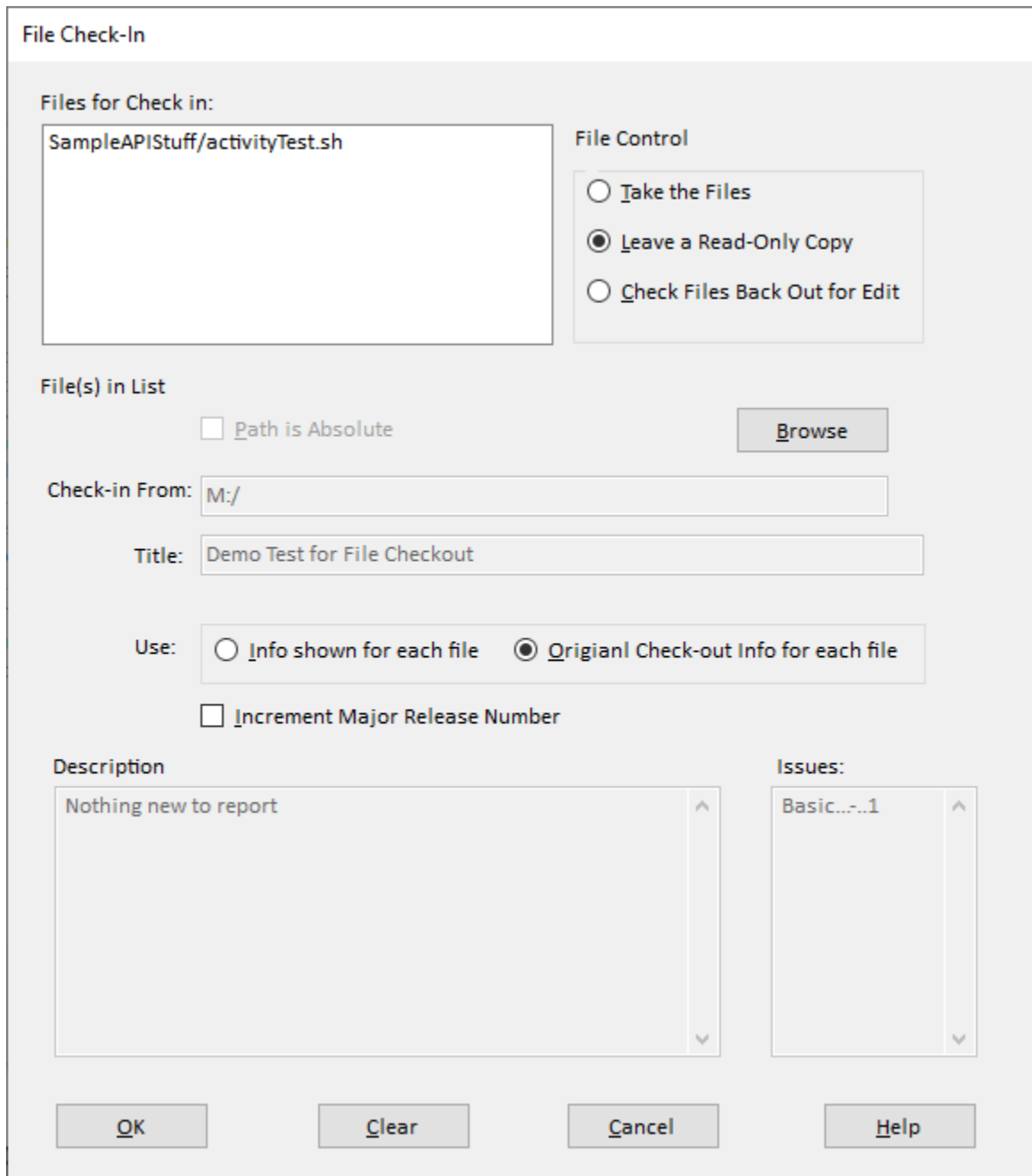
File Version: 1.1
File State: Released
Date: 6/13/2023 11:12:23 AM

Issue Number: Basic....1
Group Name: Testproject
File Operation: Checked-Out
Checked-Out to: SampleAPIStuff/activityTest.sh
By Who: John Nash
File Version: 1.1
File State: Released
Date: 6/15/2023 7:30:54 AM

Cancel

7) File Check In

After files have been modified, they must be re-entered into the database. To do so, the user first selects the relevant files from the main display of the versions program and selects the Check In button.



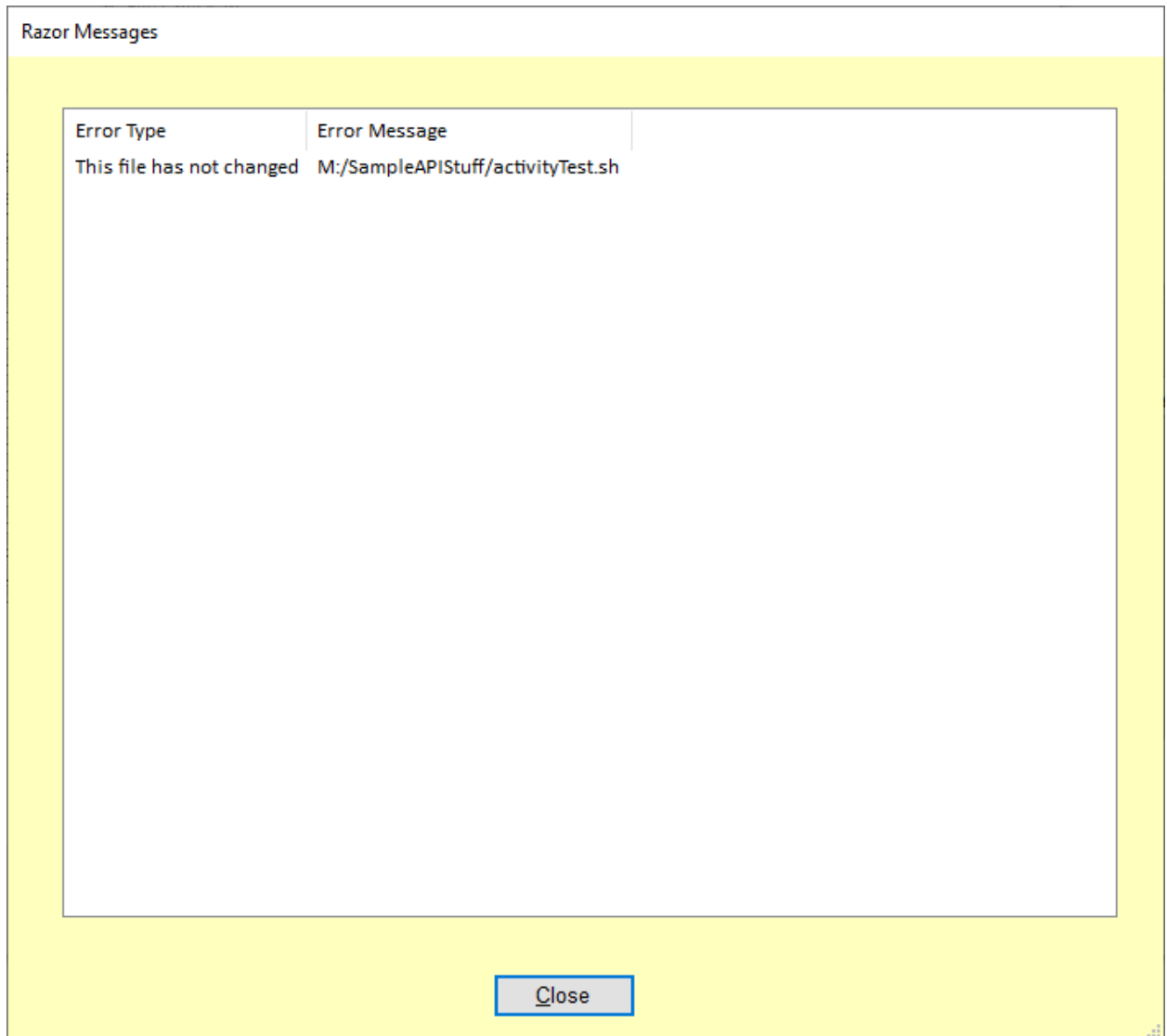
The dialog box is titled "File Check-In". It contains several sections:

- Files for Check in:** A text box containing "SampleAPIStuff/activityTest.sh".
- File Control:** A group box containing three radio buttons:
 - ☐ Take the Files
 - ☒ Leave a Read-Only Copy
 - ☐ Check Files Back Out for Edit
- File(s) in List:** A checkbox labeled "Path is Absolute" and a "Browse" button.
- Check-in From:** A text box containing "M:/".
- Title:** A text box containing "Demo Test for File Checkout".
- Use:** A group box containing two radio buttons:
 - ☐ Info shown for each file
 - ☒ Original Check-out Info for each file
- ☐ Increment Major Release Number
- Description:** A text box containing "Nothing new to report".
- Issues:** A text box containing "Basic...-1".
- Buttons:** "OK", "Clear", "Cancel", and "Help" at the bottom.

Under File Control one can select what is to be done with the file. The default is to Leave a read-only copy in the location of where the file was Checked out to and is the

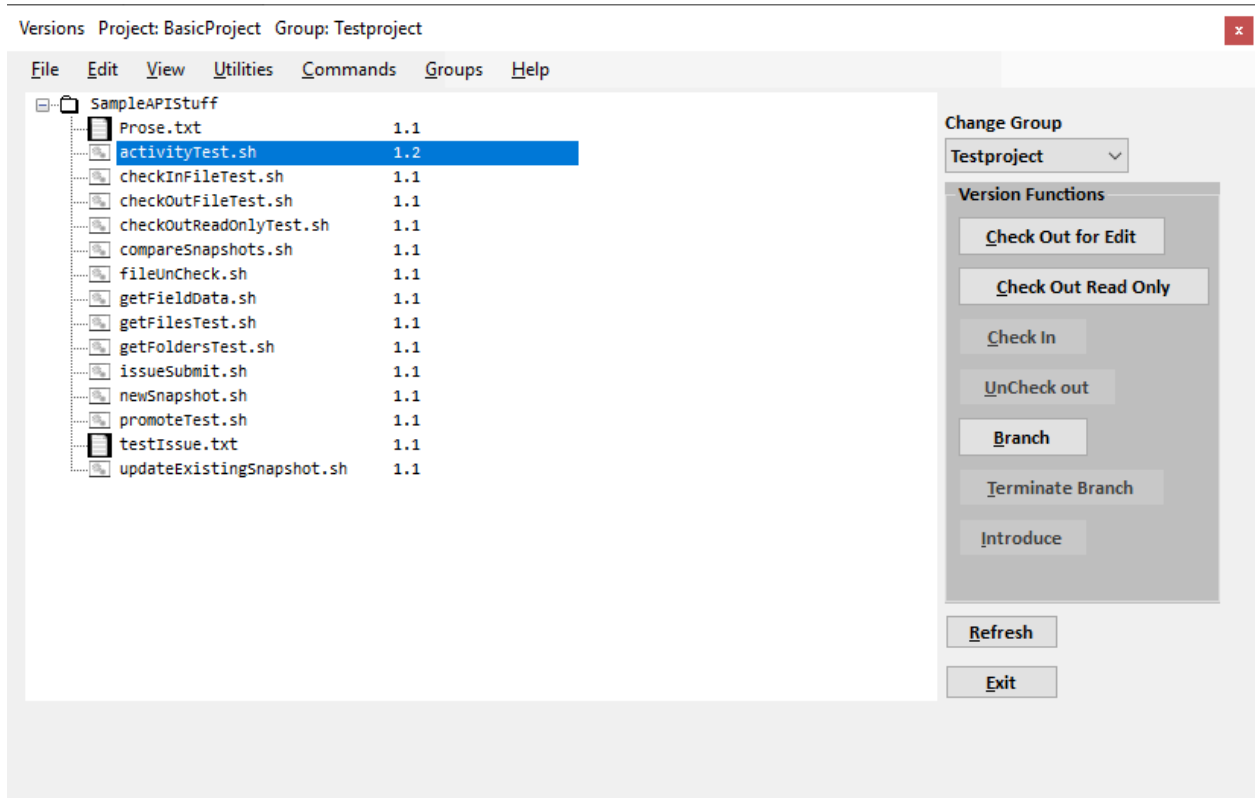
recommended action. Take the files will bring the files back into the Git repository, then delete the files from the location where they were checked out to. Check the files back out for edit is not currently implemented at this writing.

Now if the Razor Administrator has required that the file must change for a check in and the file was not edited, the following screen will appear



And nothing has been done, the file remains checked out

If the file was changed and checked back in the Versions main display will look like this

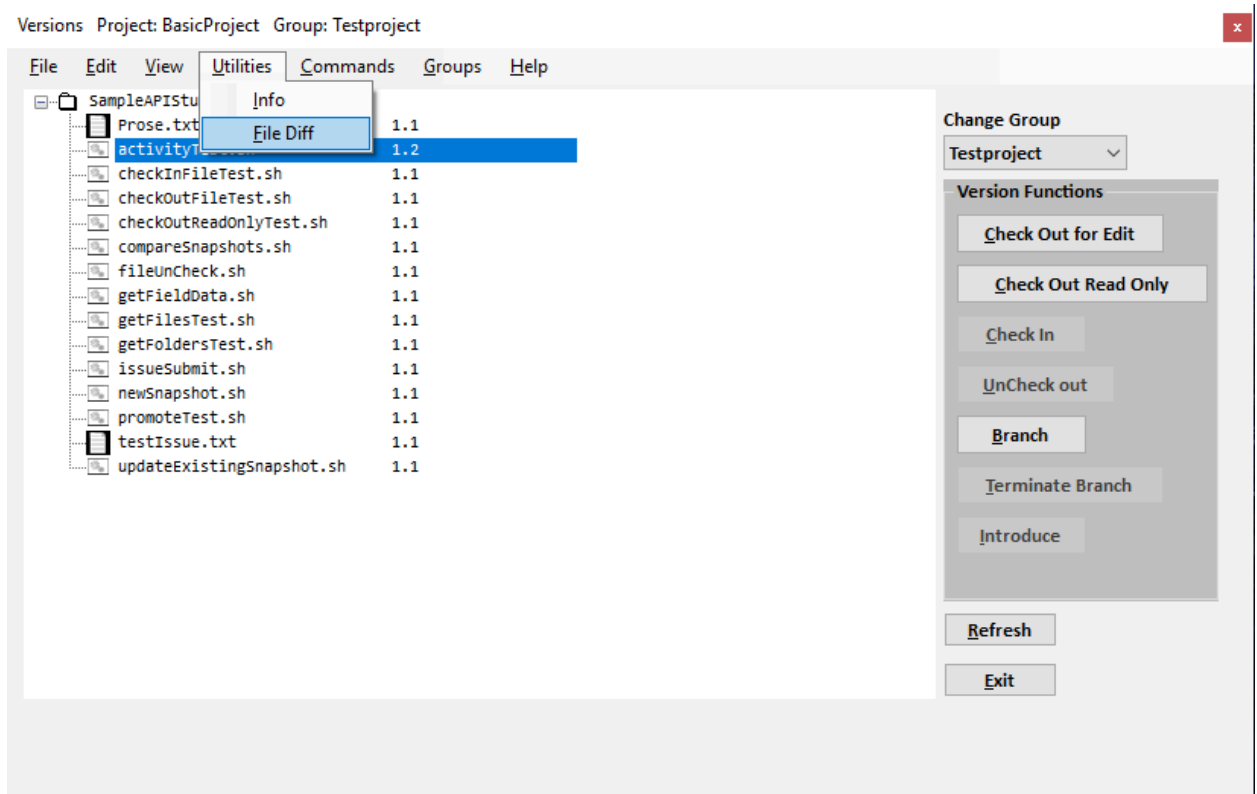


Note that the file revision has now been incremented to 1.2. The changes to this file are reflected in the Activities and Info displays respectively.

8) File Diffs using GUIFFY

Razor does not come prepackaged with the Guiffy Diff tool but it is pre-licensed for the tool. This is a major change from Razor Classic which allows the user to upgrade the Guiffy tools as needed and not have to wait on a new release from Visible Systems.

Here is the process to execute a diff on the file we just updated. First Select the file, then Utilities ☐ File Diff



The Main Diff form will show up. Here is an example with some info

Diff for file: SampleAPIStuff/activityTest.sh

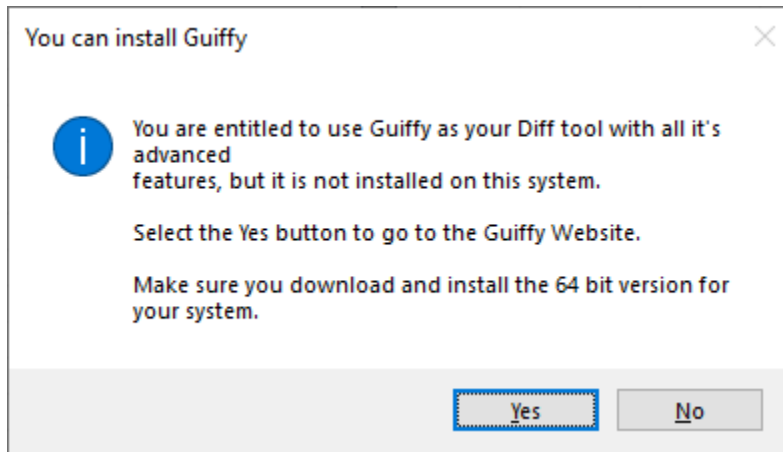
Compare Version: 1.1 ▾	Options Don't Show Identical Lines ▾	With This Version 1.1 ▾
Compare With a file outside Razor ▾ Different Versions of this File With a file outside Razor		<== Files contained in Razor <== Compare with a file not under Razor Control
Apply		Cancel

Diff Pane 1

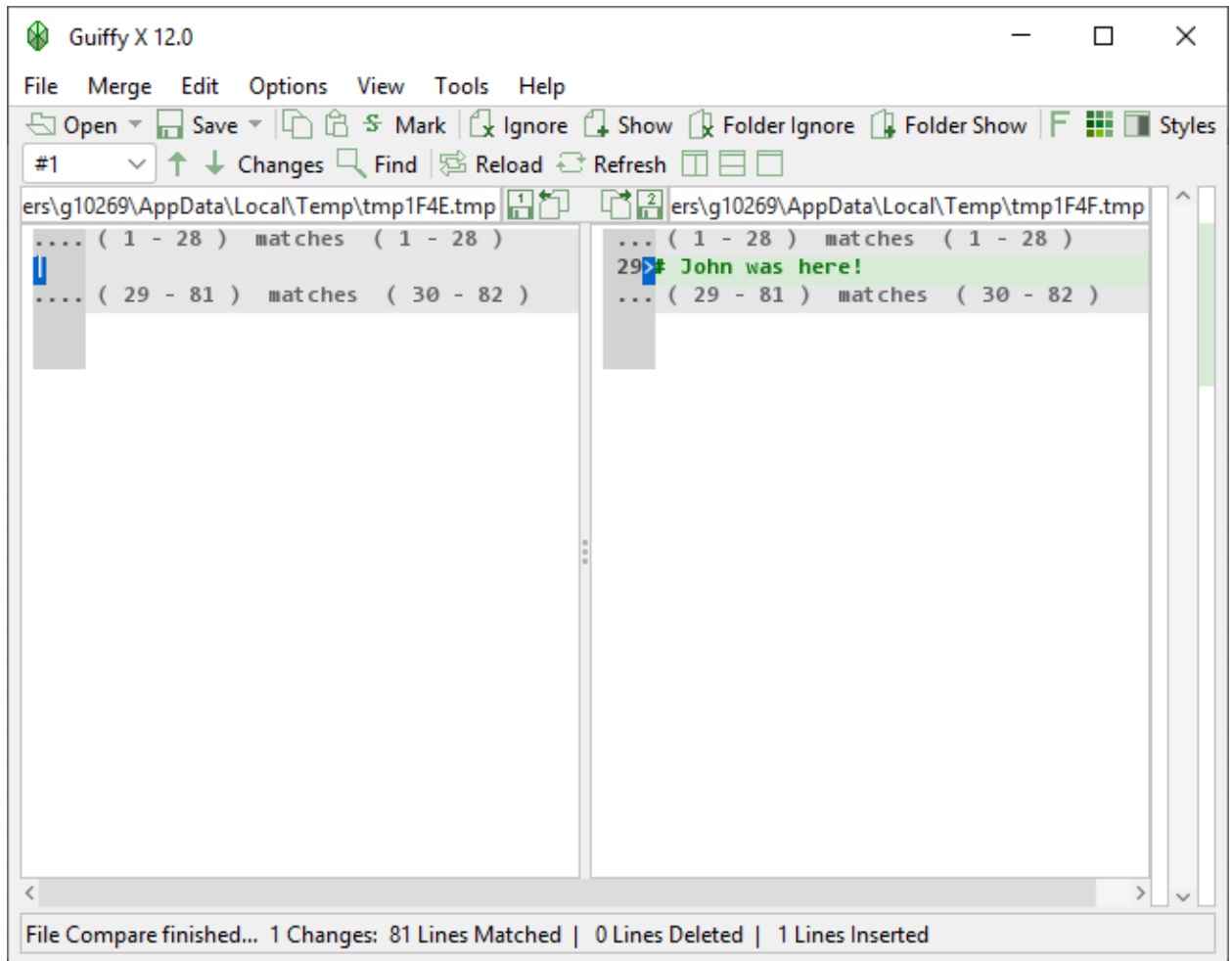
Diff Pane 2

Once the selections are made for what is to be shown and which versions to compare click on Apply

This screen will appear if you have not yet installed the Guiffy tool on this computer



If you select No, a very basic diff tool is used to display the file differences



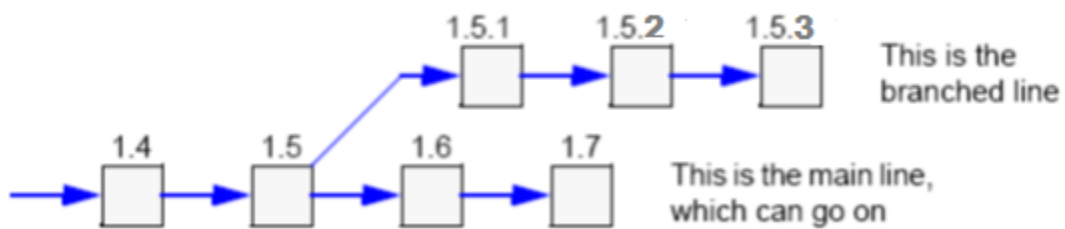
All options on the tool are available to the user as needed.

9) File Branching

Branches are deviations from the main development line for a file. They are a convenient mechanism for allowing two or more people to be working on the same file at the same time (perhaps for different goals).

A common scenario is having one person working to add new features to the product, while a second is doing bug fixes on prior versions

The version numbers of branches can be a little confusing, and warrant a quick discussion. Version numbers on the main development line have only two parts; a major and minor number.



Branches have 3 parts to their numbering scheme. The first two parts represent the point at which the branch split off the main line. The third number indicated the branches revision level.

Looking at the main Versions page, we discover that Mike C has the file we were tasked to work on already checked out. We need to Branch this file. We begin by highlighting the file then select the Branch Button

Branching the File: SampleAPIStuff/activityTest.sh

Select Source Revision: 1.1

Reason for this Branch

Description

Clear

Available Issues

- Basic...3
- Basic...2
- Basic...1

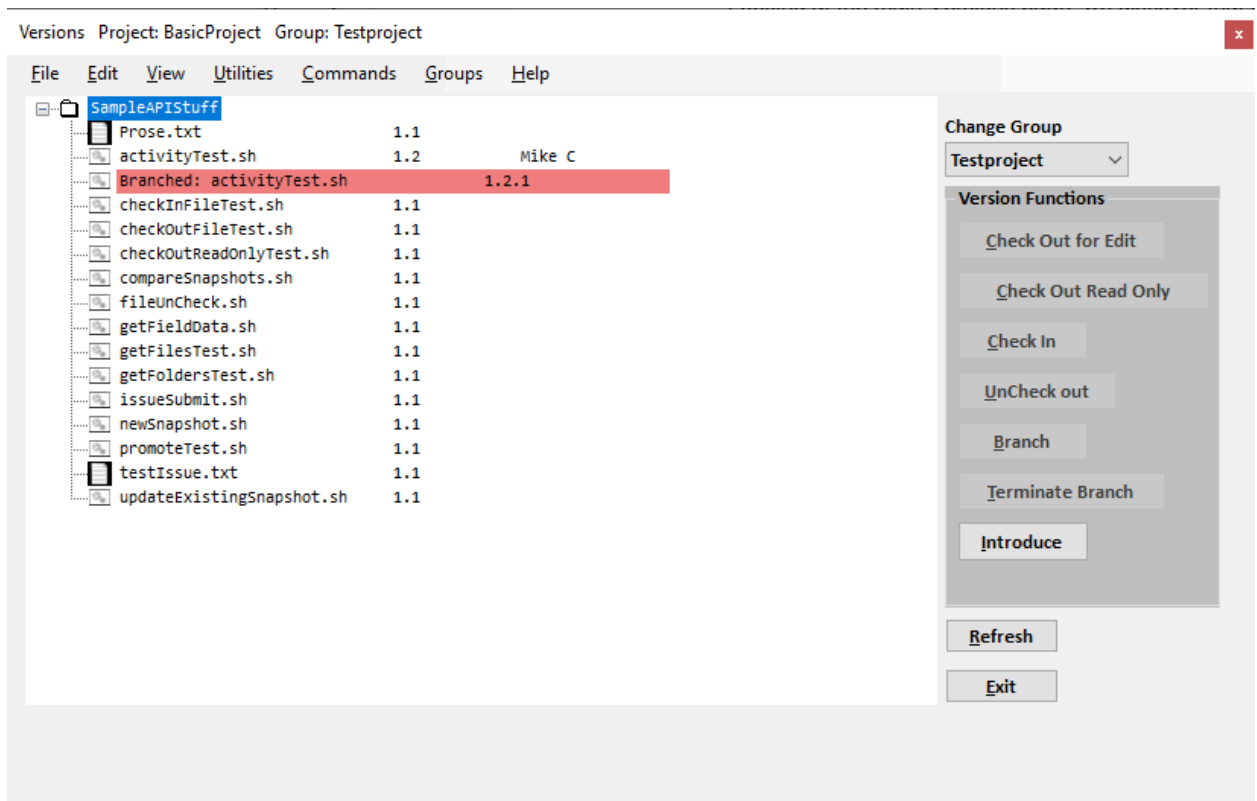
Issue Selection

- ☐ Show Issues Assigned to Me
- ☒ Show All Issues
- ☐ Show Sprints Only

OK Cancel

The Branch form is pretty strait forward. It requires the selection of the file version to branch, a reason for the branch and a description. Then assign an Issue for traceability and then select OK.

The main Versions Page now looks like this

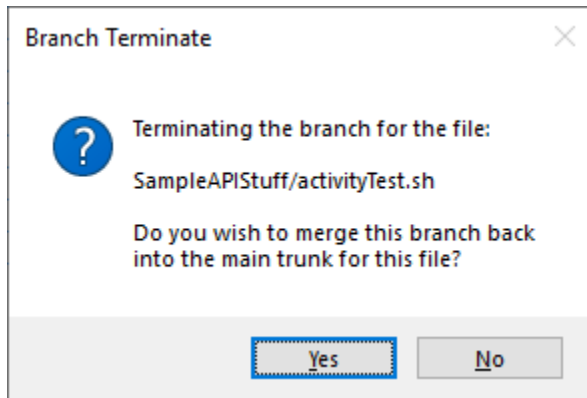


The Branch can now be checked out edited checked in checked out again to continue editing just like any other file on the main line. Likewise, the file that Mike C has checked out can be checked in and checked out again so there are 2 independent development paths for one file.

It is expected that branches are used as a temporary means of allowing concurrent development on a single file. Sooner or later, the edits made to the branched line will be incorporated into the main evolutionary line for the file.

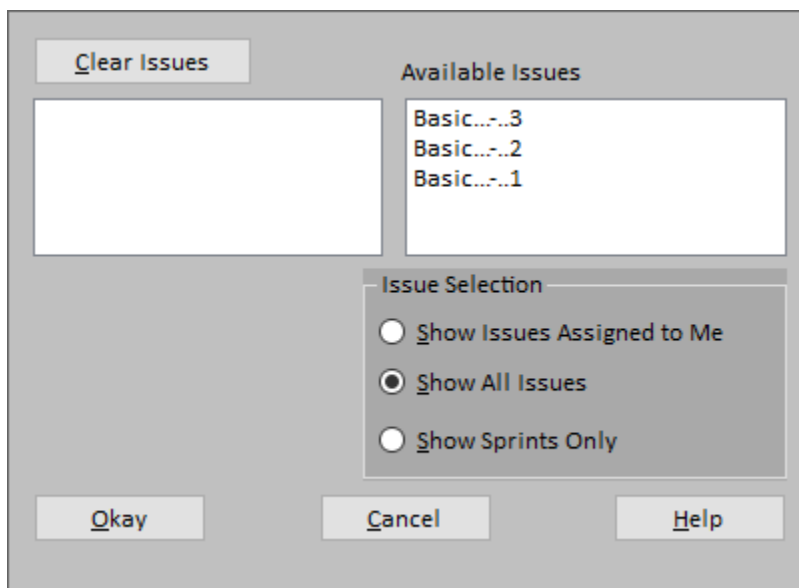
To begin the process of Terminating a Branch, the main line file and the Branched file must both be checked back into the Razor Database.

To signal the end of the utility of a branch, highlight the branched file in the main scrolling list and select the Terminate Branch button and the following will appear

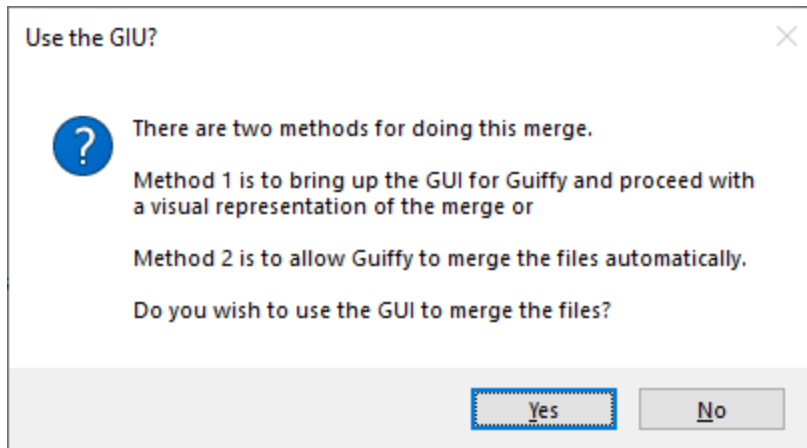


The default is Yes to merge back into the main trunk. If No is selected, then the Branch is terminated and the code changes are lost.

Selecting Yes will bring up the following for Traceability for the Merge

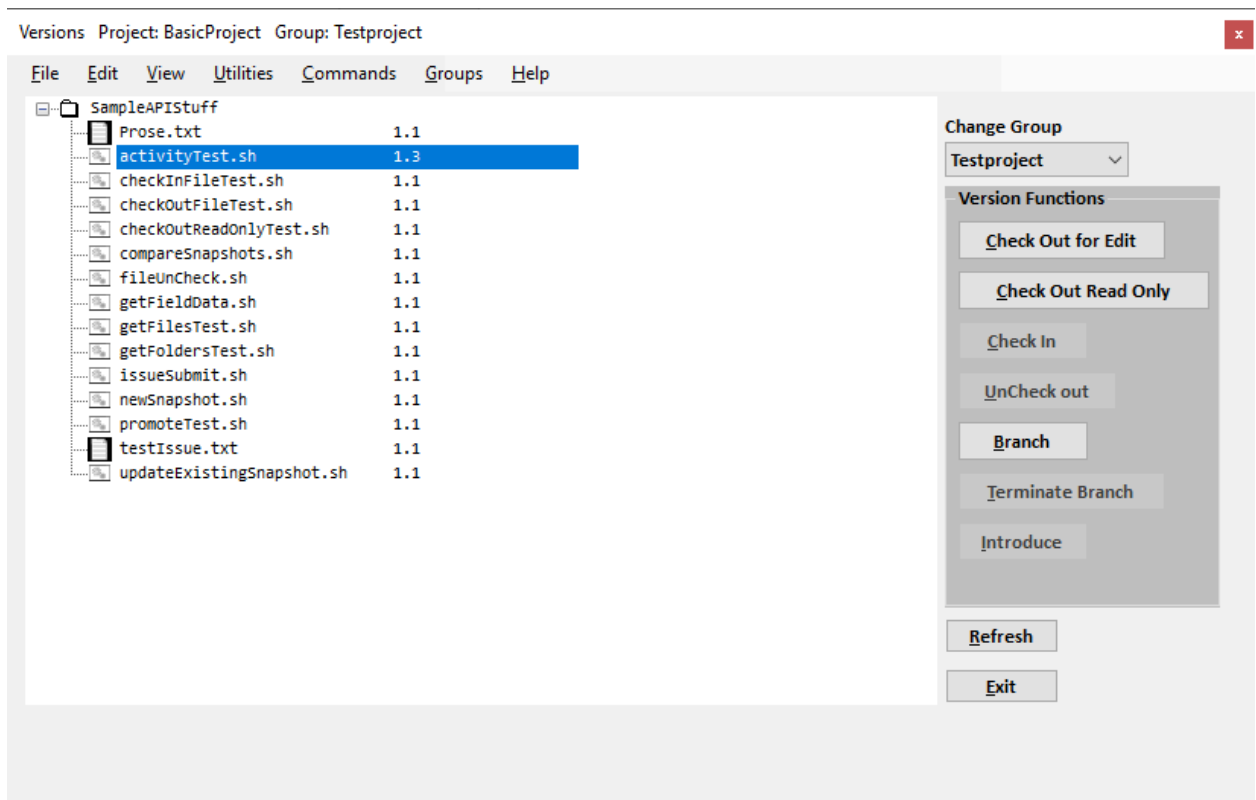


Select an Issue to this Merge and Select Okay



There are 2 options for the merge. Method 1 brings up the Guiffy GUI, if installed, to help visualize the code that will be merged into the main trunk. The user has the ability to accept or delete lines that may or may not be needed etc. prior to the merge completion.

Method 2 is allowing the Guiffy tool to automatically resolve the merge of the 2 files and create an amalgam of the two. In either case once the merge is complete the main trunk is automatically checked out the newly merged file is then checked in and the revision of the main truck is bumped up one



Baselines

1) Razor Hybrid Baselines

The Baselines program is the final piece Razor Hybrid. It provides a mechanism for defining which unique combination of files and versions "belong together" for the sake of a release or product build. This section discusses how to use the Baselines program from a user's perspective.

Within Razor Hybrid, a "group" is an organizational model. The Versions program allows you to define which files belong within which groups. All files in the same group are assigned a similar set of attributes, although the attribute collection may vary from group to group.

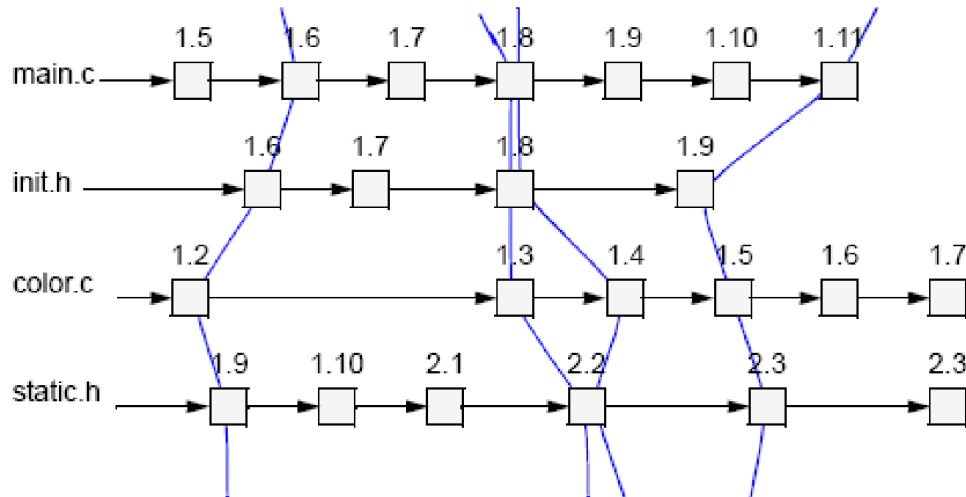
Although the Versions program provides a nice vehicle for monitoring changes on a file-by-file basis, it's often necessary to monitor changes to your effort from a broader perspective. Your product at any point in time is defined as the union of a unique combination of files, at a unique mix of their relative versions.

A Baseline or Snapshot can be thought of as a collection of specific files that have been grouped together for a special reason. One reason to group files together into a Snapshot is for release/build management. As new releases of a product are developed, the files (implying specific versions of those files) are gathered together, tested as a whole, and ultimately released as a whole.

The exact state of each file, as they relate to the Snapshot, can be reconstructed. A Snapshot is what forms the relationship among specific versions of files. Files are no longer thought of as individual, autonomous entities, but rather as what the combination becomes. An individual source file that performs mathematical calculations by itself is not worth much until it is packaged within the context of a system.

2) The Theory behind Baselines

If we imagine a simple group containing only four files, then we could diagram how each file has evolved from version to version, as well as how Snapshots come into play.



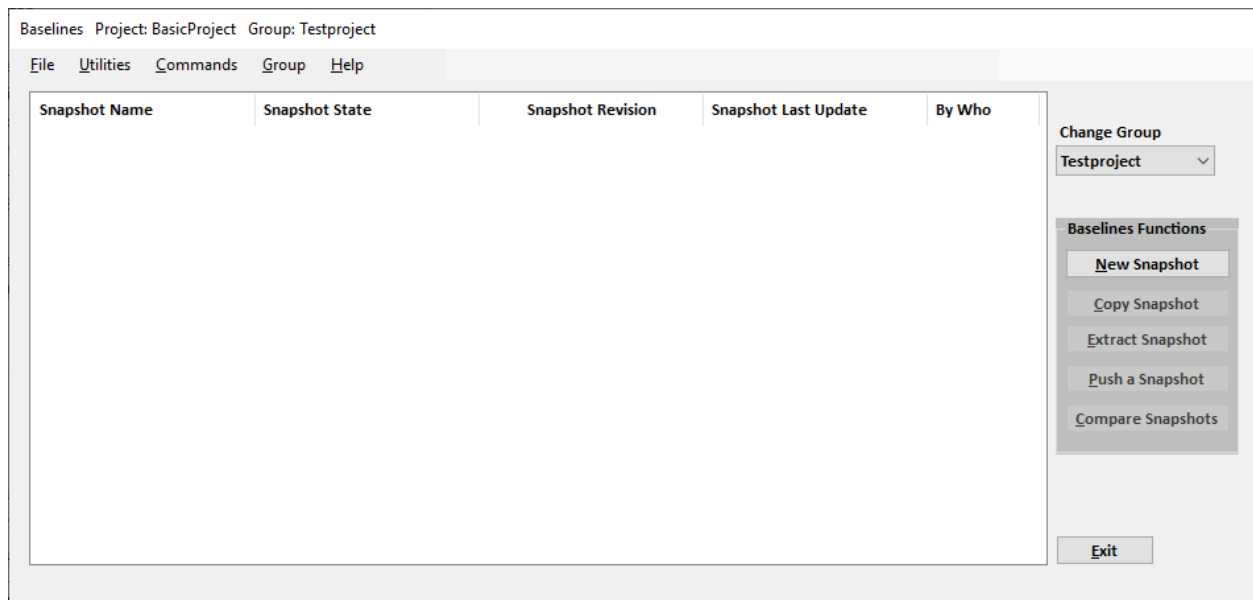
Each row of the above diagram shows how a single file within the group evolves from version to version. The vertical (blue) lines which flow behind the rows represent threads. Each thread indicates a relationship of which file/version combinations belong together. For example, the left most Snapshot on the diagram brings together main.c:1.6, init.h:1.6, color.c:1.2, and static.h:1.9. The only difference between the second and third Snapshots on the diagram is that they reference different versions of the file color.c.

The meaning and utility of a Snapshot is completely up to the user. Each Snapshot may be given a unique name, and may evolve over time. The diagram above may be represented as four different named Snapshots, or a single Snapshot that has gone through several changes itself. Often, each Razor group will have a Snapshot which defines the file/version combinations constituting a product shipment or release. This special Snapshot evolves over time to reflect the contents of each release. Separately, Snapshots may be defined to remember which combinations were compiled for the sake of lab testing, or to create a test mix of files for quick experiments. The scope and utility are completely up to the user.

3) The Main Display of Baselines

These are descriptions of the righthand controls. Some of these like the Change Group have the same functionality as the Groups Selection on the Menu bar.

There are additional options with the View and Commands selections on the Menu Bar. However, the Command selections are only available if your Razor Administrator has set them up for you. These are usually scripts or other programs that provide additional functionality to the Baselines System



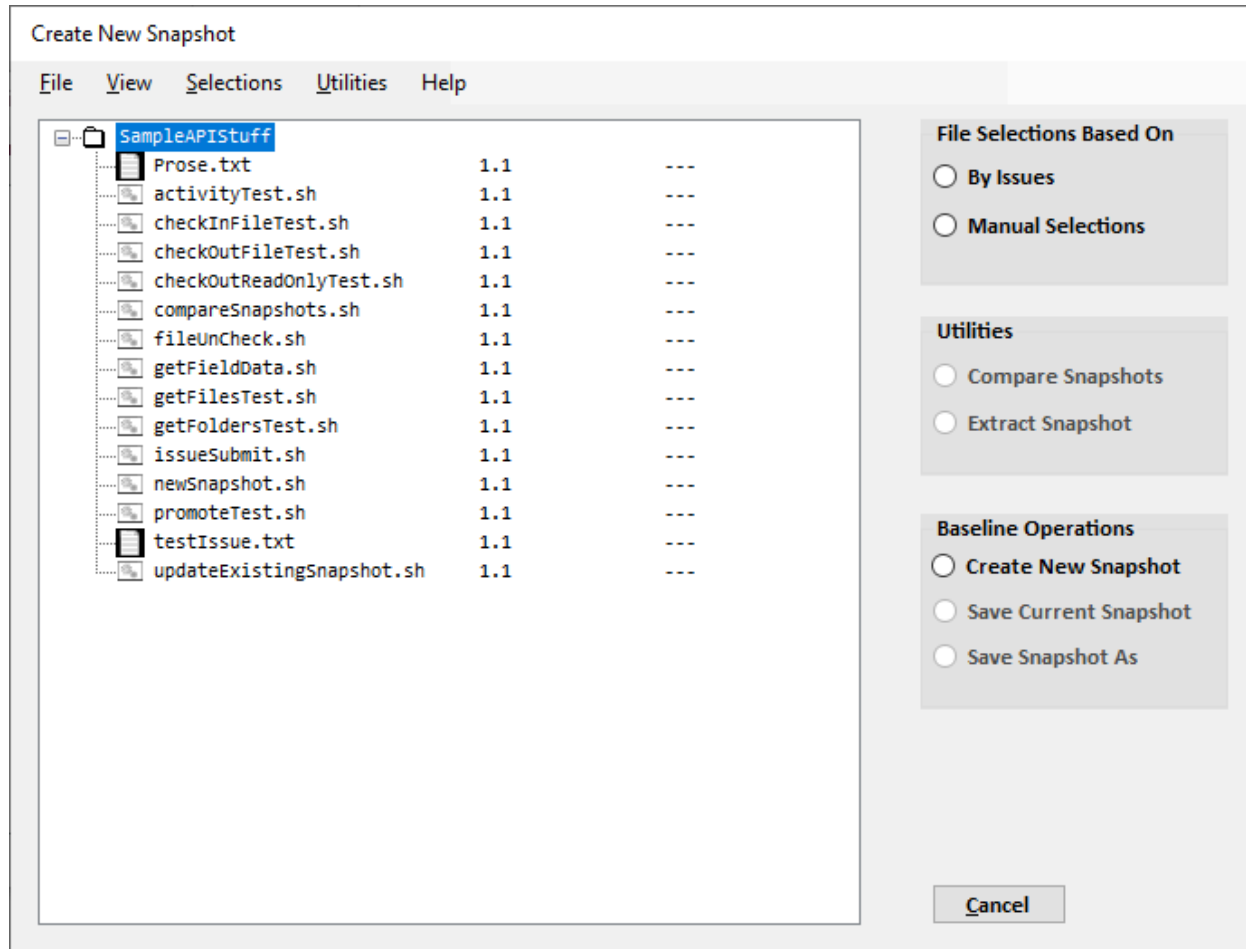
4) Creating your first Snapshot in Baselines

Once files for the project are introduced in Razor's Versions, and some editing has occurred along with some basic testing it is time to create the first Snapshot of the current state of the code.

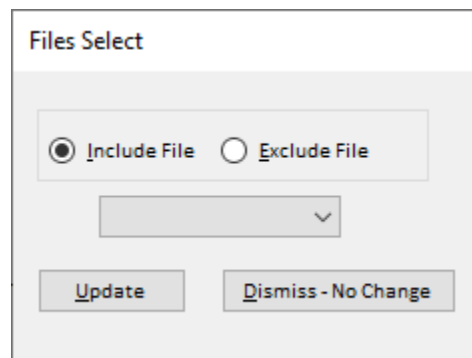
There can be up to 3 options, based on what your Razor Administrator allows. We will go through each now.

a) File by file selections

It begins by clicking on the New Snapshot button on the main Baselines form.

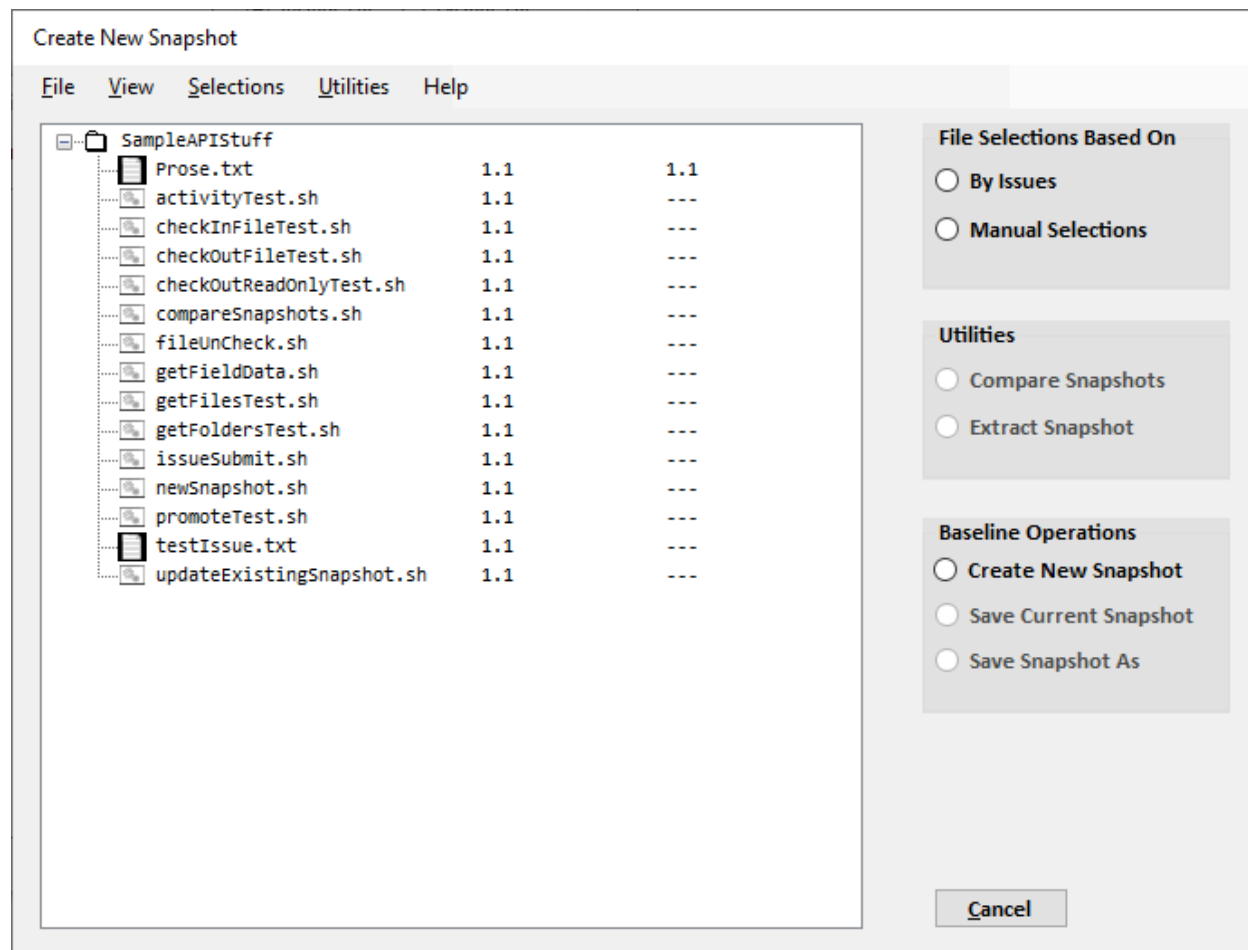


Click on each file. A new pop up will appear allowing the selection of the version of the file to include in this Snapshot



The 'Files Select' dialog box contains two radio buttons: 'Include File' (selected) and 'Exclude File'. Below them is a dropdown menu with a downward arrow. At the bottom are two buttons: 'Update' and 'Dismiss - No Change'.

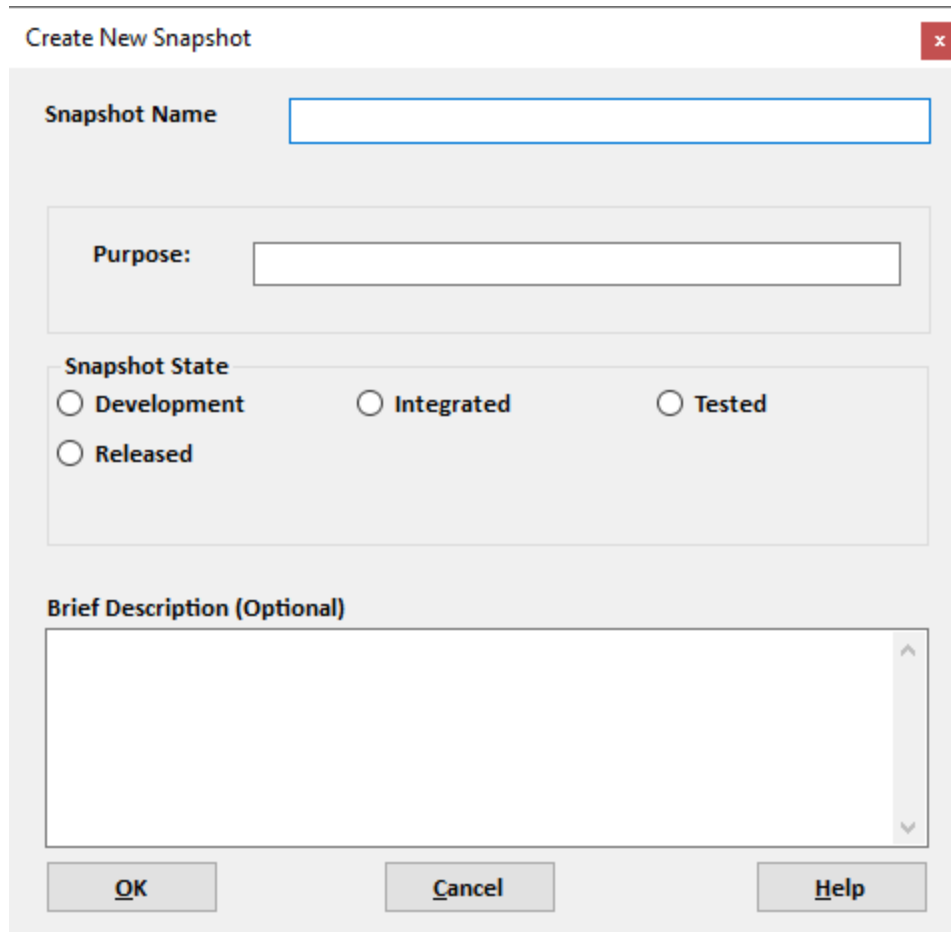
By selecting the version and then Update the file will be included in the Snapshot



The 'Create New Snapshot' dialog box features a menu bar with 'File', 'View', 'Selections', 'Utilities', and 'Help'. The main area displays a tree view of 'SampleAPIStuff' containing a list of files and their versions. The files are: Prose.txt (1.1, 1.1), activityTest.sh (1.1, ---), checkInFileTest.sh (1.1, ---), checkOutFileTest.sh (1.1, ---), checkOutReadOnlyTest.sh (1.1, ---), compareSnapshots.sh (1.1, ---), fileUnCheck.sh (1.1, ---), getFieldData.sh (1.1, ---), getFilesTest.sh (1.1, ---), getFoldersTest.sh (1.1, ---), issueSubmit.sh (1.1, ---), newSnapshot.sh (1.1, ---), promoteTest.sh (1.1, ---), testIssue.txt (1.1, ---), and updateExistingSnapshot.sh (1.1, ---). On the right, there are three sections: 'File Selections Based On' with 'By Issues' and 'Manual Selections' (selected); 'Utilities' with 'Compare Snapshots' and 'Extract Snapshot'; and 'Baseline Operations' with 'Create New Snapshot' (selected), 'Save Current Snapshot', and 'Save Snapshot As'. A 'Cancel' button is at the bottom right.

File	Version 1	Version 2
Prose.txt	1.1	1.1
activityTest.sh	1.1	---
checkInFileTest.sh	1.1	---
checkOutFileTest.sh	1.1	---
checkOutReadOnlyTest.sh	1.1	---
compareSnapshots.sh	1.1	---
fileUnCheck.sh	1.1	---
getFieldData.sh	1.1	---
getFilesTest.sh	1.1	---
getFoldersTest.sh	1.1	---
issueSubmit.sh	1.1	---
newSnapshot.sh	1.1	---
promoteTest.sh	1.1	---
testIssue.txt	1.1	---
updateExistingSnapshot.sh	1.1	---

Once you have all the files selected then Click on the Create New Snapshot button



The image shows a 'Create New Snapshot' dialog box. It has a title bar with the text 'Create New Snapshot' and a close button (X) on the right. The main area contains several input fields and a group of radio buttons. At the top is a text field labeled 'Snapshot Name'. Below it is a text field labeled 'Purpose:'. Underneath that is a section titled 'Snapshot State' containing four radio buttons: 'Development', 'Integrated', 'Tested', and 'Released'. Below the radio buttons is a large text area labeled 'Brief Description (Optional)'. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Create New Snapshot

Snapshot Name

Purpose:

Snapshot State

☐ Development ☐ Integrated ☐ Tested ☐ Released

Brief Description (Optional)

OK Cancel Help

It's strait forward from here, Create a Snapshot name – no spaces please, then the purpose of the Snapshot and so on. The click the OK button

Create New Snapshot x

Snapshot Name

Purpose:

Snapshot State

☒ Development
 ☐ Integrated
 ☐ Tested

☐ Released

Brief Description (Optional)

The main screen will now show your Snapshot

Baselines Project: BasicProject Group: Testproject

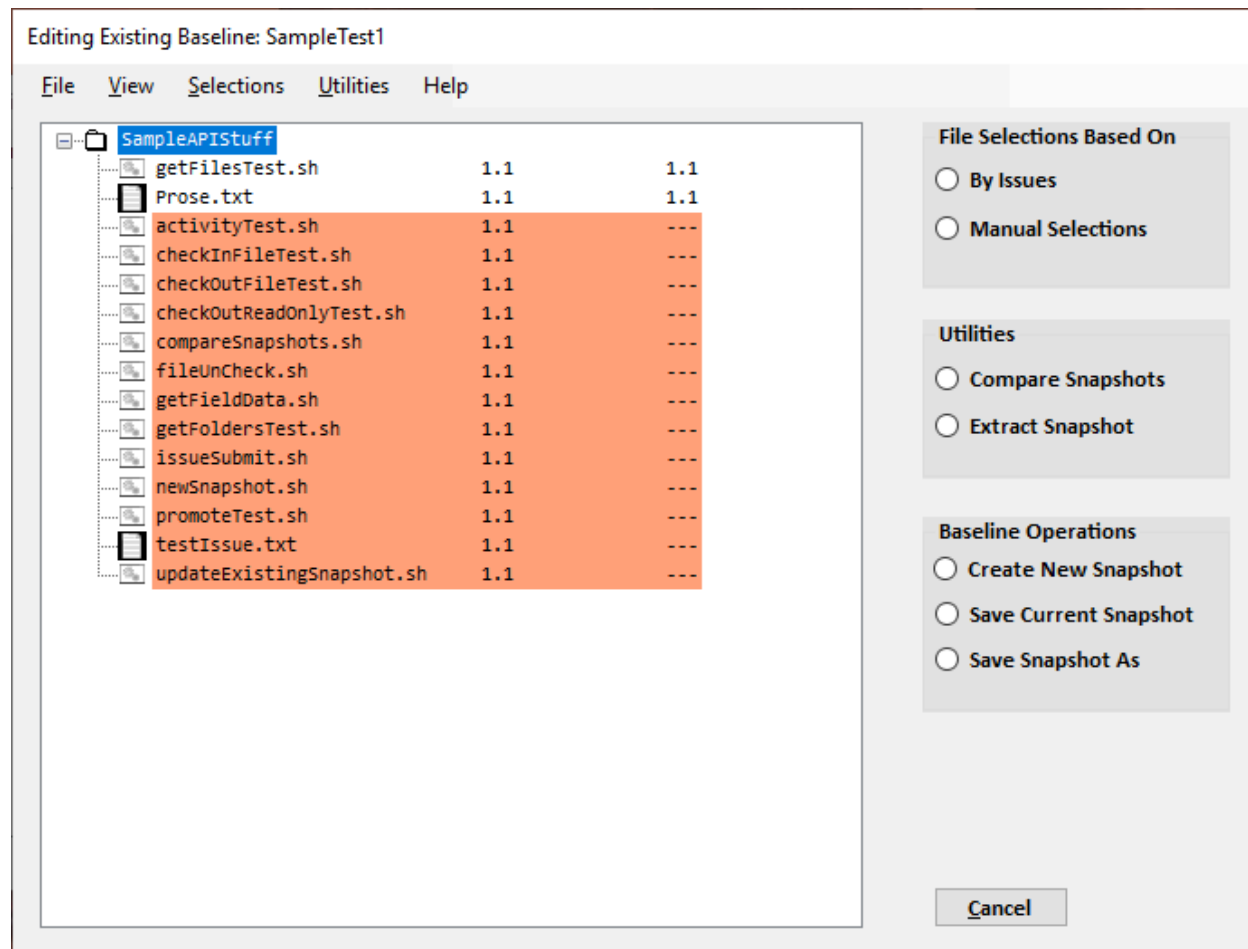
File Utilities Commands Group Help

Snapshot Name	Snapshot State	Snapshot Revision	Snapshot Last Update	By Who
SampleTest1	Development	1.1	7/10/2023 8:45:10 AM	John Nash

Change Group
Testproject ▼

Baselines Functions

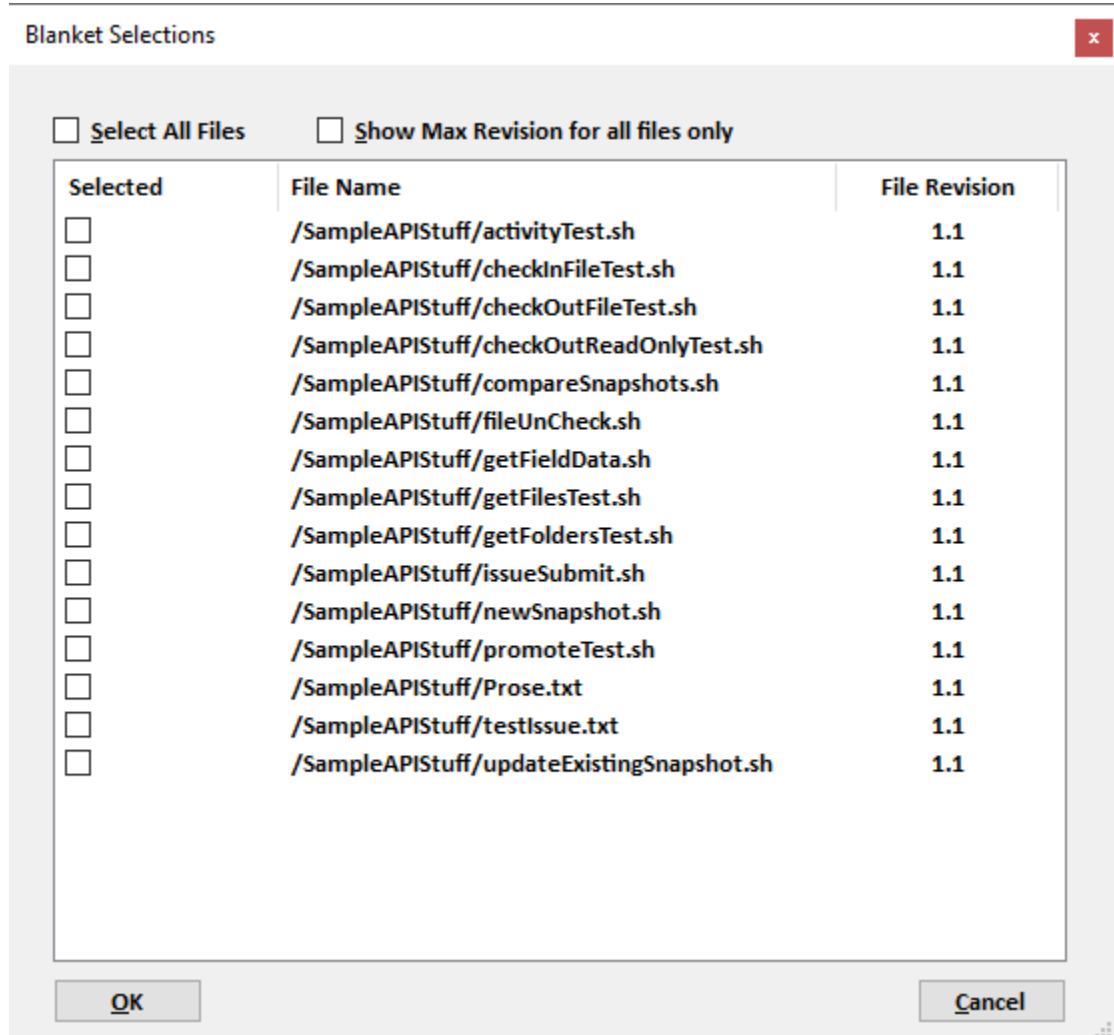
If you double click on the Snapshot the display will show the files that are included and those files which are not



This can become tedious if there are a lot of files to include so let explore the other options.

b) Manual Selections

The second method begins by selecting New Snapshot then selecting Manual Selections.



At this time one can select the desired files for this Snapshot by clicking the check box next to each file

Blanket Selections

☐ Select All Files☐ Show Max Revision for all files only

Selected	File Name	File Revision
☒	/SampleAPIStuff/activityTest.sh	1.1
☐	/SampleAPIStuff/checkInFileTest.sh	1.1
☒	/SampleAPIStuff/checkOutFileTest.sh	1.1
☒	/SampleAPIStuff/checkOutReadOnlyTest.sh	1.1
☐	/SampleAPIStuff/compareSnapshots.sh	1.1
☐	/SampleAPIStuff/fileUnCheck.sh	1.1
☒	/SampleAPIStuff/getFieldData.sh	1.1
☒	/SampleAPIStuff/getFilesTest.sh	1.1
☒	/SampleAPIStuff/getFoldersTest.sh	1.1
☐	/SampleAPIStuff/issueSubmit.sh	1.1
☐	/SampleAPIStuff/newSnapshot.sh	1.1
☐	/SampleAPIStuff/promoteTest.sh	1.1
☐	/SampleAPIStuff/Prose.txt	1.1
☐	/SampleAPIStuff/testIssue.txt	1.1
☐	/SampleAPIStuff/updateExistingSnapshot.sh	1.1

Or click on the Check Box for Select All files

Blanket Selections



☒ Select All Files

☐ Show Max Revision for all files only

Selected	File Name	File Revision
☒	/SampleAPIStuff/activityTest.sh	1.1
☒	/SampleAPIStuff/checkInFileTest.sh	1.1
☒	/SampleAPIStuff/checkOutFileTest.sh	1.1
☒	/SampleAPIStuff/checkOutReadOnlyTest.sh	1.1
☒	/SampleAPIStuff/compareSnapshots.sh	1.1
☒	/SampleAPIStuff/fileUnCheck.sh	1.1
☒	/SampleAPIStuff/getFieldData.sh	1.1
☒	/SampleAPIStuff/getFilesTest.sh	1.1
☒	/SampleAPIStuff/getFoldersTest.sh	1.1
☒	/SampleAPIStuff/issueSubmit.sh	1.1
☒	/SampleAPIStuff/newSnapshot.sh	1.1
☒	/SampleAPIStuff/promoteTest.sh	1.1
☒	/SampleAPIStuff/Prose.txt	1.1
☒	/SampleAPIStuff/testIssue.txt	1.1
☒	/SampleAPIStuff/updateExistingSnapshot.sh	1.1

OK

Cancel

Once the selections are complete, then click on the OK button and fill out the new Snapshot form as previously shown.

c) By Issues

The final method begins by selecting New Snapshot then selecting By Issues

The screenshot shows a software window with the following components:

- File State Selections** (disabled group):
 - ☐ File Introductions
 - ☐ Renamed Files
 - ☐ Files Checked-In
 - ☐ Moved Files
- Selected Issues**: An empty list box with up and down arrows.
- Available Issues**: A list box containing "Basic...-..1" with up and down arrows.
- Display Issues** (radio buttons):
 - ☒ All Issues
 - ☐ Assigned to Me
 - ☐ Sprint Issues Only
- Clear Issues**: A button located below the Selected Issues list.
- Files Selected Based on Choices Above**: A table with three columns: File Name, File Revision, and File Status. The table is currently empty.
- OK** and **Cancel**: Buttons at the bottom of the window.

This form has several areas that are interconnected. File State Selections are disabled until Issues are selected. The user must select at least one Issue but has the option to select several. The Issues are selected by right clicking an Issue. It is then moved into the Selected Issues field which then unlocks the File State Selections Group.

File State Selections

☐ File Introductions

☐ Renamed Files

☐ Files Checked-In

☐ Moved Files

Selected Issues

Basic...-..1

Available Issues

Basic...-..1

Display Issues

☒ All Issues

☐ Assigned to Me

☐ Sprint Issues Only

Clear Issues

Files Selected Based on Choices Above

File Name	File Revision	File Status
-----------	---------------	-------------

OK

Cancel

The user then selects which files are to be included in this new Snapshot by selecting the various states of the files.

File State Selections
☐ File Introductions
☐ Files Checked-In
☐ Renamed Files
☐ Moved Files

Selected Issues
Basic...-..1
^
v

Available Issues
Basic...-..1
^
v

Display Issues
☒ All Issues
☐ Assigned to Me
☐ Sprint Issues Only

Clear Issues

Files Selected Based on Choices Above

File Name	File Revision	File Status
-----------	---------------	-------------

OK

Cancel

File State Selections

☒ File Introductions
 ☐ Renamed Files

☐ Files Checked-In
 ☐ Moved Files

Selected Issues

Basic...-..1

Clear Issues

Available Issues

Basic...-..1

Display Issues

☒ All Issues
 ☐ Assigned to Me
 ☐ Sprint Issues Only

Files Selected Based on Choices Above

File Name	File Revision	File Status
/SampleAPIStuff/activityTest.sh	1.1	Introduce
/SampleAPIStuff/checkInFileTest.sh	1.1	Introduce
/SampleAPIStuff/checkOutFileTest.sh	1.1	Introduce
/SampleAPIStuff/checkOutReadOnl...	1.1	Introduce
/SampleAPIStuff/compareSnapshot...	1.1	Introduce
/SampleAPIStuff/fileUnCheck.sh	1.1	Introduce
/SampleAPIStuff/getFieldData.sh	1.1	Introduce
/SampleAPIStuff/getFilesTest.sh	1.1	Introduce
/SampleAPIStuff/getFoldersTest.sh	1.1	Introduce
/SampleAPIStuff/issueSubmit.sh	1.1	Introduce
/SampleAPIStuff/newSnapshot.sh	1.1	Introduce

OK

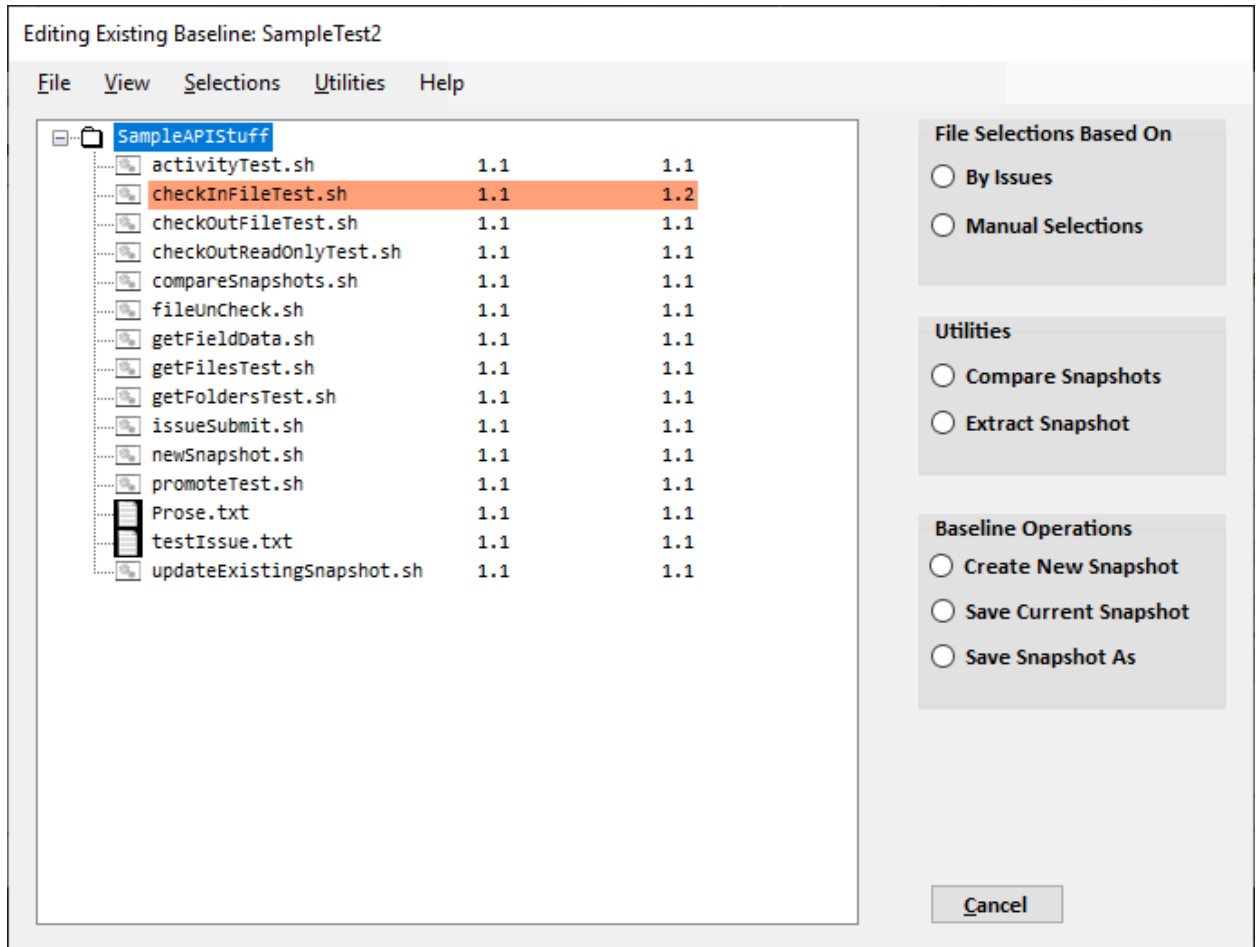
Cancel

Once the selections are complete, then click on the OK button and fill out the new Snapshot form as previously shown.

5) Update an Existing Snapshot

As development progresses, and new releases are ready for prime time, the attending Snapshot should be updated.

By double clicking on the Snapshot to be updated the user is presented a screen that will highlight all of the files that have has their revisions updated, that is been modified by the development staff.



By selecting the By Issues button, the following screen is displayed

File State Selections

☐ File Introductions ☐ Renamed Files

☒ Files Checked-In ☐ Moved Files

Selected Issues

Basic...-..1

Available Issues

Basic...-..1

Display Issues

☒ All Issues

☐ Assigned to Me

☐ Sprint Issues Only

Clear Issues

Files Selected Based on Choices Above

File Name	File Revision	File Status
/SampleAPIStuff/checkInFileTest.sh	1.2	CheckIn

OK **Cancel**

By selecting the Issues that were used for all updates, and then by selecting the desired file States a complete list of files to be included or updated in this selected Snapshot are presented. Next click on the OK button.

Updating the Baseline: SampleTest2

Snapshot Name: SampleTest2

Purpose: General Demo Update

Snapshot State

☐ Development ☐ Integrated ☐ Tested ☐ Released

Brief Description (Optional)

OK Cancel Help

Notice that the Snapshot Name is Greyed out – can't be changed. But the Purpose field is open and must be filled out. Also, the State area is opened so that the state of this Snapshot can be changed if needed.

Once everything is updated click on the OK button, then the cancel button.

The Snapshot is then Updated

Baselines Project: BasicProject Group: Testproject

File Utilities Commands Group Help

Snapshot Name	Snapshot State	Snapshot Revision	Snapshot Last Update	By Who
SampleTest1	Development	1.1	7/10/2023 8:45:10 AM	John Nash
SampleTest2	Tested	1.2	7/18/2023 9:10:57 AM	John Nash

Change Group
Testproject ▼

Baselines Functions

New Snapshot

Copy Snapshot

Extract Snapshot

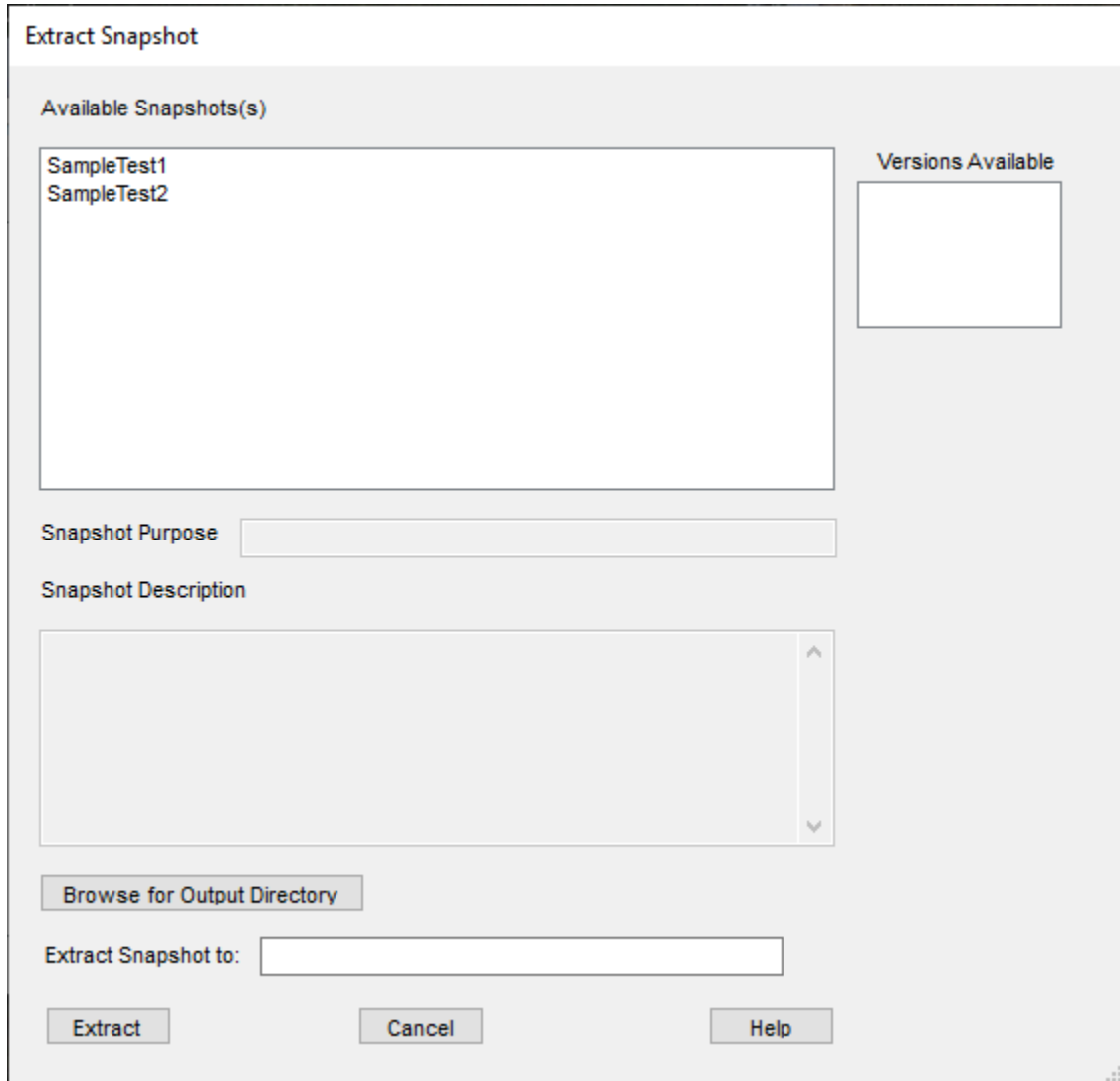
Push a Snapshot

Compare Snapshots

Exit

6) Extracting a Snapshot

From the main Baseline Panel, selecting the Extract Snapshot button the following panel shows up



The dialog box is titled "Extract Snapshot". It contains the following elements:

- Available Snapshots(s):** A list box on the left containing "SampleTest1" and "SampleTest2".
- Versions Available:** A list box on the right, currently empty.
- Snapshot Purpose:** A text input field.
- Snapshot Description:** A multi-line text area with a vertical scrollbar.
- Browse for Output Directory:** A button.
- Extract Snapshot to:** A text input field.
- Buttons:** "Extract", "Cancel", and "Help" at the bottom.

The top left corner of the panel contains a list of all the Snapshots for the indicated Razor group. As different Snapshots are highlighted on that list, the scrolling list in the top right of the panel will update to show various versions of the Snapshot. The purpose and Description fields will also update as necessary to show what was entered at the time of creation.

The Set output directory button allows the user to navigate to the directory where the files should be placed.

Extract Snapshot

Available Snapshots(s)

SampleTest1
SampleTest2

Versions Available

1.1
1.2

Snapshot Purpose: General Demo Update

Snapshot Description

More testing needed
John Nash

[Browse for Output Directory](#)

Extract Snapshot to: M:\

Once all of the selections are made, click on the extract button to get the files into the selected location

7) Pushing a Snapshot

NOTE: This is a preliminary set up procedure and is nowhere a complete procedure. This procedure will be updated as needed. At this time the button for this process is disabled

Setting up Razor Hybrid to have the ability to push a Snapshot to a production server requires some understanding of Git, the production Server and SSH. There are several steps necessary to get this accomplished so we will begin.

NOTE: For the purpose of these instructions, we are assuming a Windows Source and destination.

a) Downloading PuTTY to Get Started

Step one is to get the PuTTY toolset from the following URL and install it on both the Razor Server and the Production Server.

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

It is suggested to download the 64-bit MSI installer package. Once downloaded install the package on the Razor Server and the production system.

b) Creating the SSH Credentials

This section will be updated as needed to ensure that the proper Razor User for the Client/Server interaction is correct. We are determined to not have to create a key for every user but to provide a means within the Razor Client/Server to make this a seamless operation.

- 1) Launch the C:\PuttyTools\puttygen.exe tool.
- 2) Select the 'Generate' button.
- 3) Move the mouse about the GUI's blank area to add a unique, random element to the key generation process.
- 4) The key generation can take several minutes to complete.
- 5) When key generation is complete, the GUI will display the public key, key fingerprint, key comment, as well as fields for specifying and confirming the key 'passphrase'.
- 6) Specify a 'passphrase' that will be used to safeguard your key. The passphrase should be handled like an account password.
- 7) Select the 'Save public key' button to create a local copy of the key, e.g. C:\PuttyTools\puttyXpPublicKey.
- 8) Select the 'Save private key' button to create a local copy of the key, e.g. C:\PuttyTools\puttyXpPrivateKey.ppk

c) Setting Up the Server Repository

The first step is to navigate to the Baselines Repository for the group that this service is to be implemented on. It is located on the Drive/Project/BASELINES.Group that was setup by your Razor Administrator.

Next navigate to the .git repository that this service is to be installed on and open the config file in your favorite editor and add the following lines:

```
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true

[remote "live"]
    url = ssh://<FQDN of Destination>/Upper directory /<.Destination Git
Repository>/<Final Location>/
    fetch = +refs/heads/*:refs/remotes/live/*
```

Okay some notes here

The FQDN of Destination: This is the production server where the code is to be released too. Not the Razor Server.

Upper Directory: This is the location of where the Git repository is to reside.
Should be hidden (dotted) directory

Destination Git Repository: This is the directory where the program will eventually reside.

So, a URL may look like this

url=ssh://ProductionServer.someplace.com/.HiddenRepository/ReleaseFolder

d) Setting up the Production Server

First step to setting up the production Server is to install or ensure that git is installed. Next, if not already done is to create the production upper-level directory where the code executes from or in the case of a product that is available to the public to download, the download area.

Then a hidden git repository is created somewhere on the production server that is not within the release area using the command

```
git init --bare
```

Next you must edit the repository post-receive hook

Edit

```
.git/hooks/post-receive
```

Add

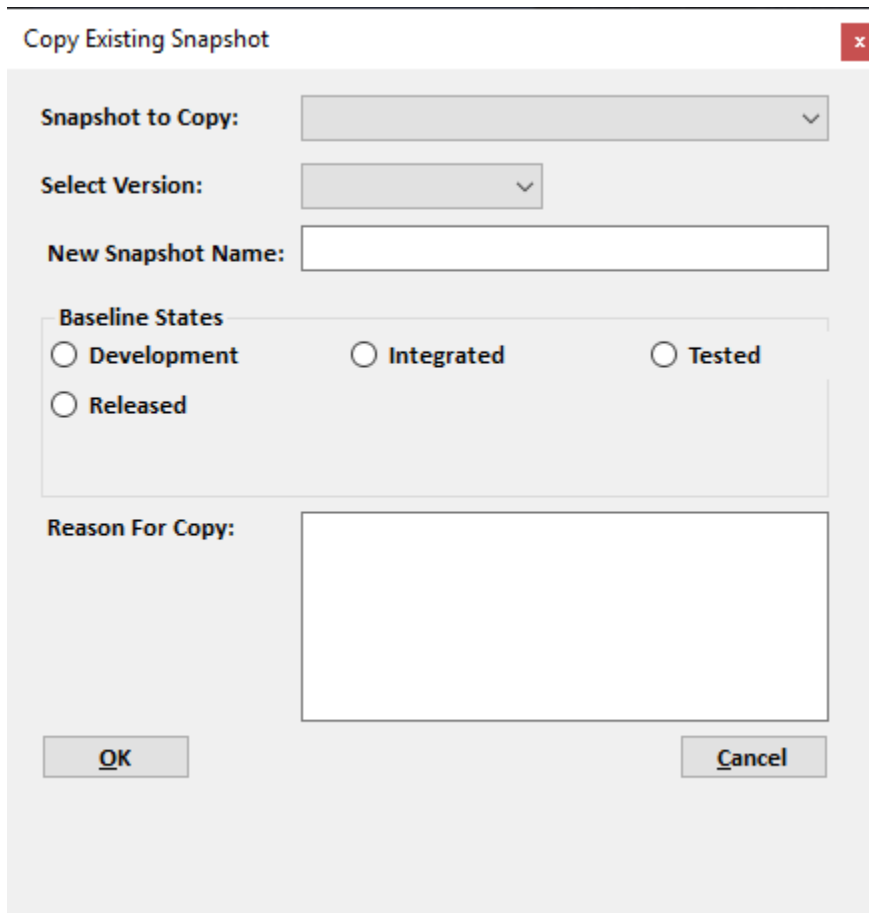
```
#!/bin/sh
```

```
git --work-tree=/var/www/html/apps/labor --git-dir=/var/.gitRepository/labor
```

```
checkout -f
```

8) Copying a Snapshot

A developer may at some point want to branch off the main development path and try something new without disturbing the current release of the code. To do this the developer selects Copy Snapshot from the main panel.

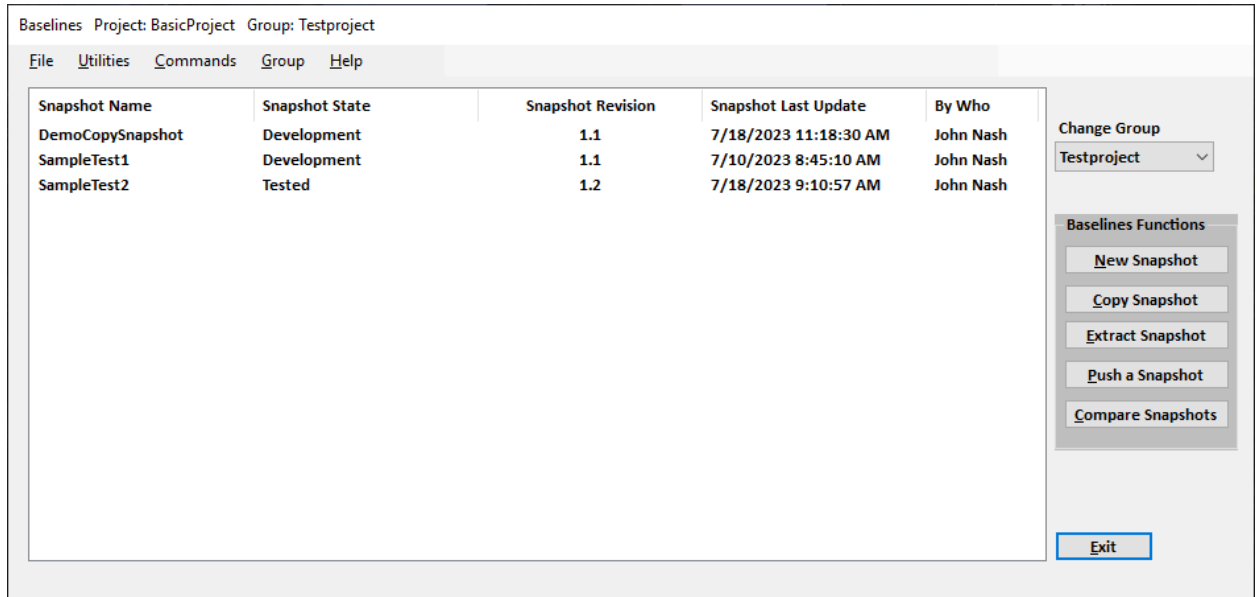


The image shows a dialog box titled "Copy Existing Snapshot" with a red close button in the top right corner. The dialog contains the following fields and controls:

- Snapshot to Copy:** A dropdown menu.
- Select Version:** A dropdown menu.
- New Snapshot Name:** A text input field.
- Baseline States:** A group box containing four radio buttons:
 - ☐ Development
 - ☐ Integrated
 - ☐ Tested
 - ☐ Released
- Reason For Copy:** A large text area.
- Buttons:** "OK" and "Cancel" buttons at the bottom.

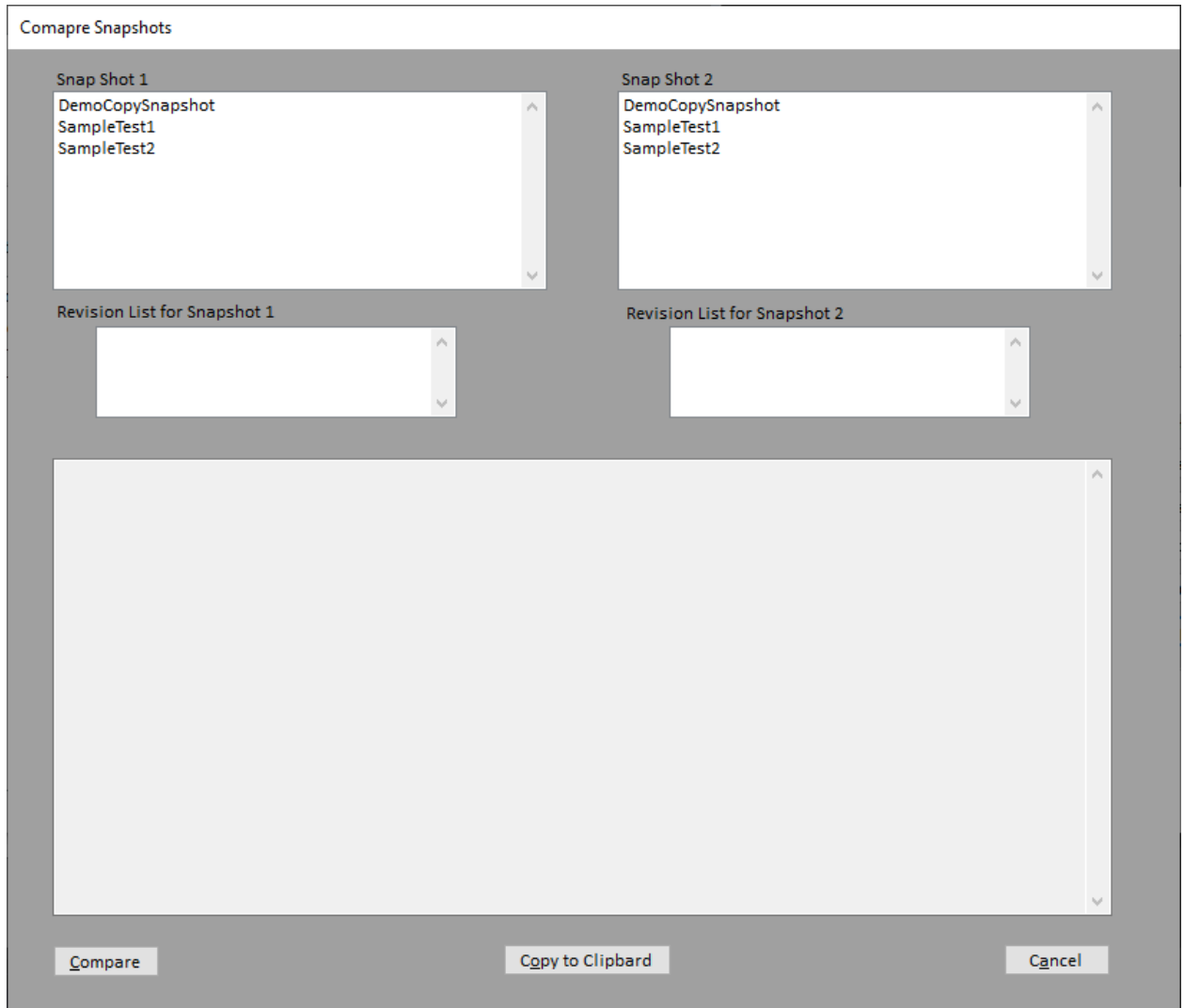
On this panel the developer selects the Snapshot to Copy and the Version of the Snapshot. Next, the developer must enter a New Snapshot Name, then select the State and the Reason for Copy, finally, click on the OK button

And the main panel is updated with the new Snapshot.



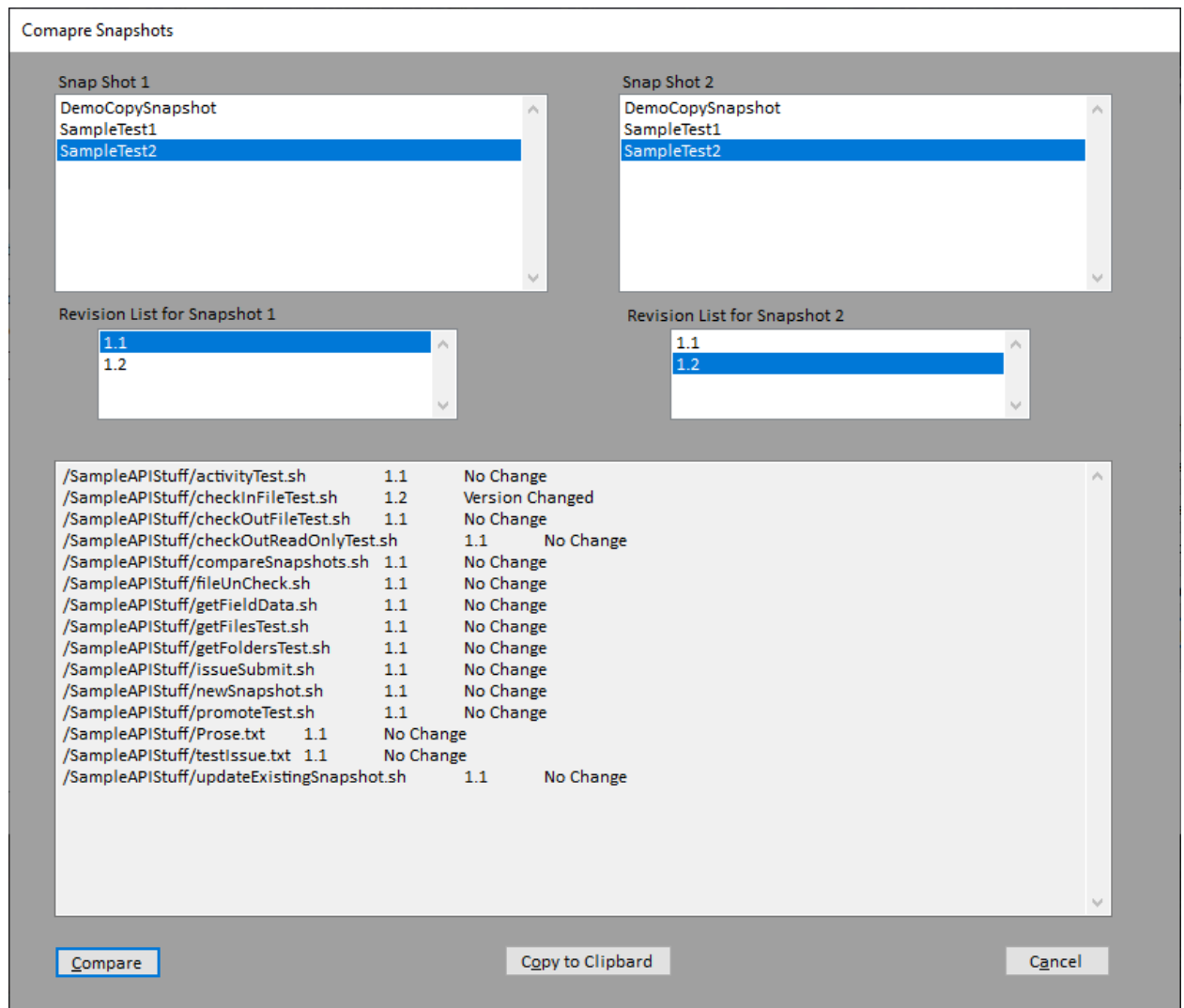
9) Comparing Snapshots

It is often important and useful to be able to see what differentiates one thread from another. Fortunately, there is a simple means of doing just that. By selecting the Compare Snapshots from the main display, the following panel will appear.



Using the mouse, the user identifies which Snapshot and version is to be on the left, and which is to be on the right. Selecting the Compare button at the bottom of the panel runs the analysis.

The output lists all files currently in the group along with each file version. The last column identifies the type of change as "No change," "Included," "Version Changed" or "Excluded."



The user has the option to Copy to Clipboard so the output can be applied to a report or into a file as needed.

Customizing Razor Hybrid to follow your Configuration Management Process.

Razor Hybrid has a very robust scripting facility and API. However, not every site has the means necessary to write and maintain scripts to support a configuration management process.

There are some built in systems in Razor Hybrid that can help with implementing a simple to middle level configuration management process that we will present below

1) Creating a project to support basic process flow

When creating a project, the Razor Administrator has the option to select some options for the project that will enforce some traceability to the project

Add New Project

Project Directory

Git Project Name

Group Name

Issue Type Select ☒ Standard ☐ Use Custom Format

Available Formats

Versions Configuration ☒ Files Must Change ☒ Must Attach Issues

Baselines Config. ☒ Select Files Based on Issues Only

☒ Use Scripts

Administrator's Email

Issue Prefix

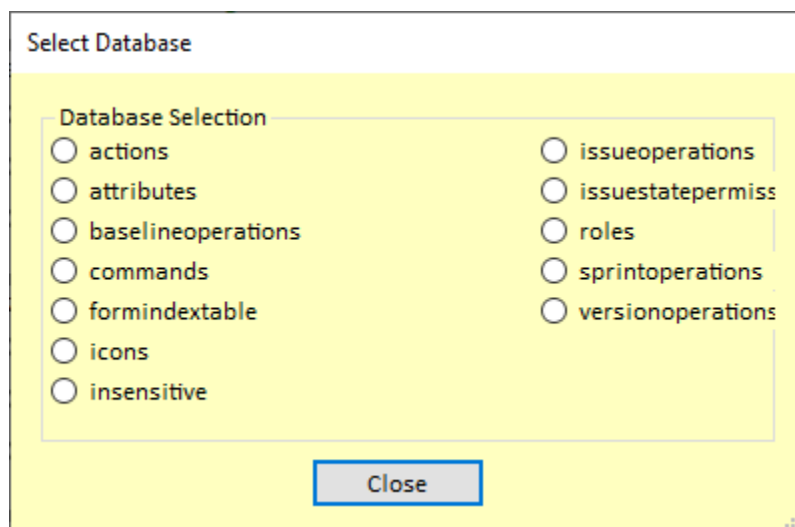
Let's take a look at the sections highlighted in green. The defaults for new projects are preset and represent some basic configuration management setting that would be handled in scripts.

- a) Files must change: When a file is checked out by a developer this setting requires that the developer must have made changes to the file prior to checking the file back into the Database. This prevent the file revision being bumped even if the file has not changed.
- b) Must Attach Issues: This begins the traceability of the change request to the work being done.
- c) Select Files Based on Issues Only: More on the traceability of the change request. This last step solidifies that the Issue that started this change is the one that creates the release.
- d) Use Scripts: Leave this checked. The before and after scripts that are executed do not impact the overall performance of Razor Hybrid but leaves the door open for some simple scripts to auto preload an Issue that we will discuss later

2) Let's discuss Roles in Razor Hybrid

Your Razor Administrator has the ability to set roles for use in your configuration management process. Roles are a defined set of users that can be used in to define who can to what to each process within Razor Hybrid; from creating an Issue, promoting an Issue to who can check out files or create Snapshots. It can get pretty involved without writing a single line of scripting!

- a) The Razor Administrator starts the Razor Admin tool and logs in
- b) Then the Edit Custom Configuration button is selected
- c) The tool will recommend to back up the database prior to making changes. This is very important to do. See the section on Database maintenance to see how this is done
- d) Next select the Roles button



- e) The following window will appear

Database Editor (Use with Caution)

	theProject	theGroup	theRole	thePlayers
*				

Close Add a Row Delete Row(s) Update Database

- f) This table in the database is very granular, that is roles and players can be set by project/group. Therefore, in a given project if there are several groups the roles can have different players. So, let's set some up
- g) First Click on the theProject cell

Database Editor (Use with Caution)

	theProject	theGroup	theRole	thePlayers
✎	razorTest1			
*				

Close Add a Row Delete Row(s) Update Database

Note that the system automatically adds in a row and we have entered the name of our project that we set up. See Razor Projects section for more details

h) Next let's add in a group for this project

Database Editor (Use with Caution)

	theProject	theGroup	theRole	thePlayers
...	razorTest1	TestGroup1		
*				

Close Add a Row Delete Row(s) Update Database

i) Now we will create Role:

Database Editor (Use with Caution)

	theProject	theGroup	theRole	thePlayers
...	razorTest1	TestGroup1	SWQA	
*				

Close Add a Row Delete Row(s) Update Database

Note: SWQA stands for Software quality Assurance

j) Now we will add in the members of this Role. The members that are listed must use their login ID or UID not their full name.

Database Editor (Use with Caution)

	theProject	theGroup	theRole	thePlayers
	razorTest1	TestGroup1	SWQA	mike123,john123,al1234
▶*				

Close Add a Row Delete Row(s) Update Database

- k) Once you are satisfied with the entries then click on Update Database and close.

3) Ok, I Created a Role Now What?

Now we get to work with several tables to implement our new role.

Select Database

Database Selection

- ☐ actions
- ☐ attributes
- ☒ baselineoperations
- ☐ commands
- ☐ formindextable
- ☐ icons
- ☐ insensitive
- ☐ issueoperations
- ☐ issuestatepermiss
- ☐ roles
- ☐ sprintoperations
- ☐ versionoperations

Close

There are 5 tables highlighted in the Database editor. Each has a specific function within Razor Hybrid that the table controls. So, let's look at one for an example Issue States

Database Editor (Use with Caution)

	formName	formIndex	IssueOrSprint	StateFrom	StateTo	ByWho	Email
	standard	1	0	Submitted	Submitted	*	
	standard	1	0	Submitted	Rejected	*	
	standard	1	0	Submitted	Approved	*	
	standard	1	0	Submitted	Closed	*	
	standard	1	0	Rejected	Submitted	*	
	standard	1	0	Rejected	Rejected	*	
	standard	1	0	Rejected	Approved	*	
	standard	1	0	Rejected	Closed	*	
	standard	1	0	Approved	Submitted	*	
	standard	1	0	Approved	Rejected	*	
	standard	1	0	Approved	Approved	*	
	standard	1	0	Approved	Closed	*	
	standard	1	0	Closed	Submitted	*	
	standard	1	0	Closed	Rejected	*	
	standard	1	0	Closed	Approved	*	
	standard	1	0	Closed	Closed	*	

This is the IssueStatePermission Table. Notice that a standard Issue has 4 states: Submitted, Rejected, Approved and Closed. Each state can translon from one state to another at any time in the default setup due to the "*" in the ByWho column. Now to enable the role we created SWQA to only allow members of this role to move an Issue from Approved to Closed we do the following:

Database Editor (Use with Caution)

	formName	formIndex	IssueOrSprint	StateFrom	StateTo	ByWho	Email
	standard	1	0	Submitted	Submitted	*	
	standard	1	0	Submitted	Rejected	*	
	standard	1	0	Submitted	Approved	*	
	standard	1	0	Submitted	Closed	*	
	standard	1	0	Rejected	Submitted	*	
	standard	1	0	Rejected	Rejected	*	
	standard	1	0	Rejected	Approved	*	
	standard	1	0	Rejected	Closed	*	
	standard	1	0	Approved	Submitted	*	
	standard	1	0	Approved	Rejected	*	
	standard	1	0	Approved	Approved	*	
	standard	1	0	Approved	Closed	@SWQA	
	standard	1	0	Closed	Submitted	*	
	standard	1	0	Closed	Rejected	*	
	standard	1	0	Closed	Approved	*	
	standard	1	0	Closed	Closed	*	
	standard	1	1	Submitted	Submitted	*	

Close Add a Row Delete Row(s) **Update Database**

We add in the SWQA role into the Approved to Closed Row as shown and then select update the database button.

Note: The role must have the “@” sign in front or else the server will try to find the user SWQA.

Now to allow only the razoradmin to move an Issue from Submitted to Reject one must add in the razoradmin to that transition field. This will prevent everyone else from trying to move a submitted state to rejected.

Database Editor (Use with Caution)

	formName	formIndex	IssueOrSprint	StateFrom	StateTo	ByWho	Email
	standard	1	0	Submitted	Submitted	*	
✎	standard	1	0	Submitted	Rejected	razoradmin	
	standard	1	0	Submitted	Approved	*	
	standard	1	0	Submitted	Closed	*	
	standard	1	0	Rejected	Submitted	*	
	standard	1	0	Rejected	Rejected	*	
	standard	1	0	Rejected	Approved	*	
	standard	1	0	Rejected	Closed	*	
	standard	1	0	Approved	Submitted	*	
	standard	1	0	Approved	Rejected	*	
	standard	1	0	Approved	Approved	*	
	standard	1	0	Approved	Closed	@SWQA	
	standard	1	0	Closed	Submitted	*	
	standard	1	0	Closed	Rejected	*	
	standard	1	0	Closed	Approved	*	
	standard	1	0	Closed	Closed	*	

Now should every state transition have a role or a user? Maybe. If left as it is above then all states that do not have an entry are open to everyone.

How to prevent a specific user from moving an Issue or maybe submitting and Issue is accomplished by using the "!" in from of their user login ID

Database Editor (Use with Caution)

	formName	formIndex	IssueOrSprint	StateFrom	StateTo	ByWho	Email
✎	standard	1	0	Submitted	Submitted	!bob123	
	standard	1	0	Submitted	Rejected	razoradmin	
	standard	1	0	Submitted	Approved	*	
	standard	1	0	Submitted	Closed	*	
	standard	1	0	Rejected	Submitted	*	
	standard	1	0	Rejected	Rejected	*	
	standard	1	0	Rejected	Approved	*	
	standard	1	0	Rejected	Closed	*	
	standard	1	0	Approved	Submitted	*	
	standard	1	0	Approved	Rejected	*	
	standard	1	0	Approved	Approved	*	
	standard	1	0	Approved	Closed	@SWQA	
	standard	1	0	Closed	Submitted	*	
	standard	1	0	Closed	Rejected	*	
	standard	1	0	Closed	Approved	*	
	standard	1	0	Closed	Closed	*	

In this example bob123 will not be able to submit an Issue, but everyone else can.

4) Operations Tables

These tables govern how each of the tools are controlled. Here is a typical operations table for Versions

Database Editor (Use with Caution)

	projectName	groupName	Operation	ByWho
	razorTest1	TestGruop1	Branch	*
	razorTest1	TestGruop1	CheckIn	*
	razorTest1	TestGruop1	CheckOut	*
	razorTest1	TestGruop1	Introduce	*
	razorTest1	TestGruop1	FileMerger	*
	razorTest1	TestGruop1	ReadOnly	*
	razorTest1	TestGruop1	Remove	*
	razorTest1	TestGruop1	Move	*
	razorTest1	TestGruop1	NewFolder	*
	razorTest1	TestGruop1	Rename	*
	razorTest1	TestGruop1	Revert	*
	razorTest1	TestGruop1	TerminateBranch	*
	razorTest1	TestGruop1	FileCommandsPulldown	*
	razorTest1	TestGruop1	FileViewer	*
	razorTest1	TestGruop1	UnCheckOut	*
*				

Close Add a Row Delete Row(s) Update Database

It functions much like the States Tables covered above with an additional option. This option is OWNER and can be used in several unique situations. Let's say bob123 has a file checked out. In the current setup anyone can come along and check in, un check out the file without bob123 knowing that this has been done. To fix this problem you can add in OWNER in these positions

Database Editor (Use with Caution)

	projectName	groupName	Operation	ByWho
	razorTest1	TestGruop1	Branch	*
	razorTest1	TestGruop1	CheckIn	OWNER
	razorTest1	TestGruop1	CheckOut	*
	razorTest1	TestGruop1	Introduce	*
	razorTest1	TestGruop1	FileMerger	*
	razorTest1	TestGruop1	ReadOnly	*
	razorTest1	TestGruop1	Remove	*
	razorTest1	TestGruop1	Move	*
	razorTest1	TestGruop1	NewFolder	*
	razorTest1	TestGruop1	Rename	*
	razorTest1	TestGruop1	Revert	*
	razorTest1	TestGruop1	TerminateBranch	*
	razorTest1	TestGruop1	FileCommandsPulldown	*
	razorTest1	TestGruop1	FileViewer	*
...	razorTest1	TestGruop1	UnCheckOut	OWNER
*				

Close Add a Row Delete Row(s) Update Database

Again, as before in the Roles section, the “!” before an UID or Role can be used to restrict or enable operations as necessary.

Scripting

As stated, before, Razor Hybrid has a very robust Scripting Interface and API. Visible Systems is always working to improve the API and welcomes suggestions for improvement or new API requests.

It is important to note that if a site has many projects but all need to use the same scripts the use of links in each project is encouraged so make a change only has to be completed one.

On the Linux servers this is strait forward and the standard commands work but on Windows it is a bit different. The command is mklink /J destination and then source directories.

1) Razor Hybrid Server Environmental Variables

These Environmental Variables are created by the server when a script request is detected by the server. They should not be changed at any time by any scripts unless noted below

RAZOR_HOME	Where the Binaries for the Server are located
RAZOR_PROJECT_DIR	Where the Project directory is located
RAZOR_GROUP	What group are we working in
RAZOR_SCRIPTS_DIR	Where the user scripts are located
RAZOR_PRELOAD	The file that contains Issue preloads
RAZOR_TMP	The Razor temp storage area
RAZOR_PID	Random PID number
RAZOR_USER	Logged in user FULL name
RAZOR_USER_EMAIL	Logged in User Email
USER	User login
TMP	Unix Temp Directory
RAZORHYBRID	Yep, we be running Razor Hybrid
PASSED_PARAMETERS	Any passed data from the server to the scripts
PASSED_FILE_TO_SCRIPTS	A file passed from the server to the scripts
PASSED_FILE_TO_SERVER	A file from the scripts to the server

2) Razor Hybrid API

The API for Razor Hybrid contains 2 types of commands. Those that do not involve the entire server engine and those that do not. The commands that require the server engine must include the keyword "console" as show in the following items

- a. Standard commands

To execute these standard commands in the console, first start a Windows Console and then enter:

cd c:\Program Files\Visible Systems\Razor Hybrid Server

Once there enter the commands as listed below

- | | |
|------------------------------------|-------------------------------------|
| i. Razor6Server.exe id | Returns the ID of the computer |
| ii. Razor6Server.exe license_info | Returns the License Status |
| iii. Razor6Server.exe version | Returns the release version |
| iv. Razor6Server.exe who | Returns who is logged in |
| v. razor.exe rz_time | Returns the Integer timestamp (now) |
| vi. Razor6Server.exe rz_fmt_time | Returns the formatted Date time |
| based on integer timestamp | |
| vii. Razor6Server.exe rz_time_val: | Returns the Integer timestamp |
| based on the input time string | |

b. Server Interactive Commands (more to follow)

- | | |
|-----------------------|--|
| i. Promote_issue | Promotes the issue to the passed state |
| ii. Issue_submit | Submits a new issue |
| iii. Modify_issue | Modifies or updates an existing issue |
| iv. Get_activity | Gets the activity associated to an Issue(s) |
| v. Checkout_read_only | Extracts the given file as read only to a specified location |
| vi. File_check_out | Check out a file for edit |
| vii. File_check_in | Checks in a file |
| viii. unCheck_out | Releases the lock on a file if it is checked out |
| ix. razor_files | Get the directory/files and revisions |
| x. razor_folders | Gets the folders in a given Project/Group |

3) Standard Command Usage

To execute these standard commands in the console, first start a Windows Console and then enter:

- g) cd c:\Program Files\Visible Systems\Razor Hybrid Server
- h) Enter the desired command in as follows
Razor6Server.exe console id
- i) The command will display the output on the console.
- j) To run the command via a script, see the next section on Interactive Commands

4) Scripted Command Usage

Running Server requests from a Script no matter what type the user must inform the server that it is to run the command in “console” mode. This is because the Razor Hybrid Server is running as a service and services cannot normally be accessed from a

command line. However, by using the `--past` parameter when calling a `RazorService` command the server will execute the command and return the results to the Script.

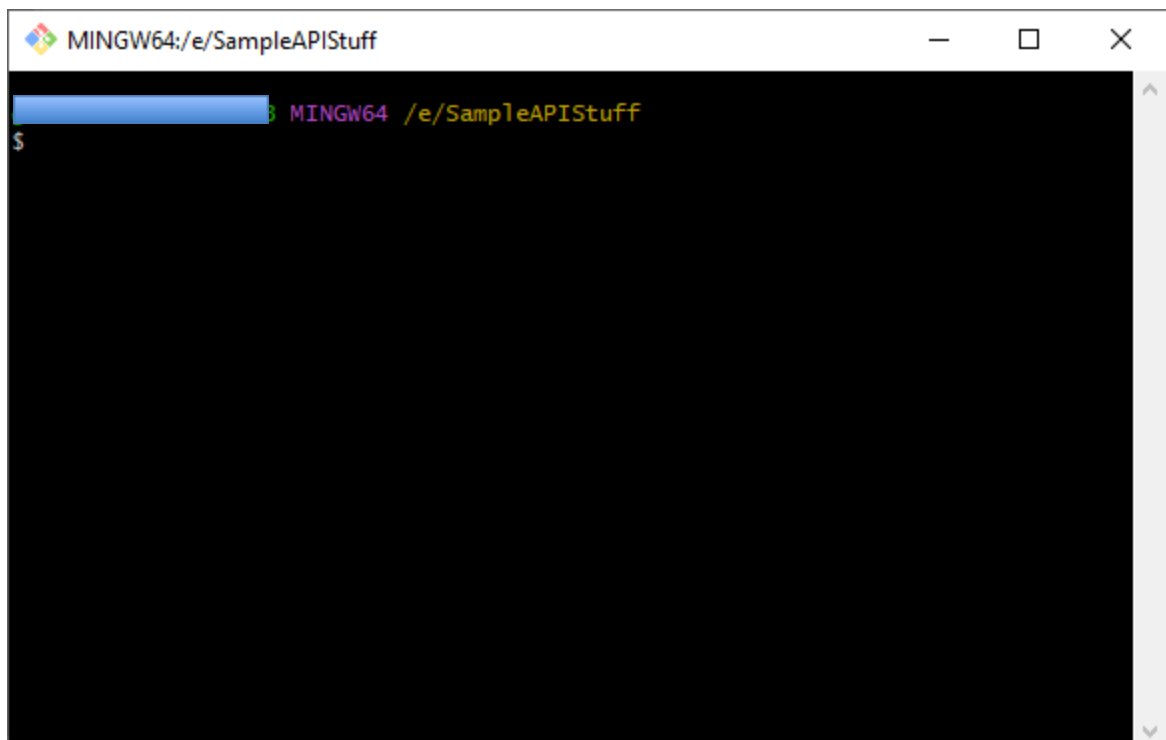
Something to note here is that the error paths return a robust parcel of information back to the script. It may make it more difficult to test for this information, however it will indeed speed up development of a script as the programmer will know what is amiss in their script.

One last thing to note. The `system_setup` script supplied by Visible Systems Corporation and installed when the server is installed is for the most part complete and tested. But, if while developing scripts and you are required to update any of these, please send the updates to razor_support@visiblesystems.com so we can include them in our next release. The install program will overwrite these locations.

Visible Systems is providing sample scripts for your use and or modification. Each script will demonstrate the proper calls and return values. These scripts are contained in a ZIP file which can be downloaded from this location:

<https://www.visiblesystemscorp.com/Products/Razor/scripts/index.htm>

Once downloaded, unzip the files in a desired location. Next if on Windows use the File Explorer to navigate to where you unzipped the files, then Right click and select Git Bash Here option. This will bring up a Git Bash Shell it will look something like this:



Note the directory format: /e/SampleAPIStuff. If you are going to specify file locations within a script this is the format to use. E:\SampleAPIStuff will not work at all.

Next you will need to set the RAZOR_HOME path. So, enter EXPORT=" C:\Program Files\Visible Systems" and press the enter key. At this point, open any of the scripts in WordPad or the editor of your choice, yes vi works here too and read the script's passed parameters requirements.

5) Running scripts from the Command Pull Down

Each of the tools in Razor Hybrid with the exception of Sprints has available to the user the Command pull down in the main tool bar. If your Administrator has made commands available to the user this is where they will be available from. Sprints do not have a Command Pull Down available as they are a super set of Issues and can be worked with within the Issues Tool.

Exit codes for Command scripts

There are three exit codes for these types of scripts:

0 = Normal Exit. No reporting back script results to the user

1 = Informative exit. A window will pop up to the user that will have a message about the script results.

11 = Error Output. Usually used while debugging a script

On the return code 1 to return a message to the user it would look something like this:

```
#!/usr/bin/bash
echo "This is a test
      And we are adding a second line to that test"
exit 1
```

Appendix

Usage:

Preload File Usage Example:

Note: The Attribute (Originator) and the Result (RAZOR_USER) are separated by a single tab

```
echo "Originator<TAB>$RAZOR_USER" >> $RAZOR_PRELOAD
```

Passed Parameters Usage Example:

This is for 2 passed parameters in a comma separated list.

(PassedParam1,PassedParam2)

```
if [[ -z "${1}" ]]
then
    # Do something based on nothing
    echo "Nothing passed in"
else
    # This will assume that only 2 parameters are passed.
    # It provides a simple method for getting the goodies
    passedVar1=`echo "${1}" | "${cutexec}" -d ',' -f 1`
    passedVar2=`echo "${1}" | "${cutexec}" -d ',' -f 2`
fi
```

Format for passed result is as follows:

\$updateThisAttribute,\$item1,\$item2,\$item3

Where \$updateThisAttribute is the item that is to be updated within Razor followed by a comma and the items values to be used for the update separated by commas

```
echo $result
```

Passed File from server to scripts usage Example

This file will contain a formatted Issue, full file from versions or Baseline output
It can be parsed within the external script as any normal file would be

\$1 is the file path to the passed file, in this case the Issue

```
# The fourth item in the file is the issue name which we need to  
# Get so this is one way to do it  
ISSUE=`"${sedexec}" -n '4p' $1`
```