HCL VersionVault Integration with Visual Studio Code

Visual Studio Code is a lightweight, powerful, desktop source code editor. For more information, please visit https://code.visualstudio.com

HCL VersionVault is a software configuration management (SCM) tool used for version control, which provides controlled access to source code, design documents, models, test plans and test results. It features secure version management and authoritative build auditing. For more information, please visit https://www.hcltechsw.com/versionvault

The HCL VersionVault integration with **Visual Studio Code IDE** provides the following VersionVault operations.

- Add Existing View
- Connect to WAN Server
- Activities - Create Activity, Set Existing Activity
- Add To Source Control
- Check Out
- Check In
- Undo Checkout
- Hijack
- Undo Hijack
- Rename
- Remove
- Refresh
- Update From Repository
- Disconnect from WAN Server

It supports **UCM** and **Base VersionVault**. It supports only the Windows platform with two view types: **Web** and **Automatic**.

Prerequisite
- VersionVault or the VersionVault Remote Client must be installed.
- Download and install Visual Studio Code from https://code.visualstudio.com
- A VersionVault view (Web/Automatic) must be created using VersionVault tools.

1. HCL VersionVault Extension installation in Visual Studio Code IDE
The **HCL VersionVault** extension is available in the Visual Studio Code Marketplace. In the Visual Studio Code IDE, click the **Extensions** icon from left panel and search for "**HCL VersionVault**". To install the extension, click on the **"Install"** button.
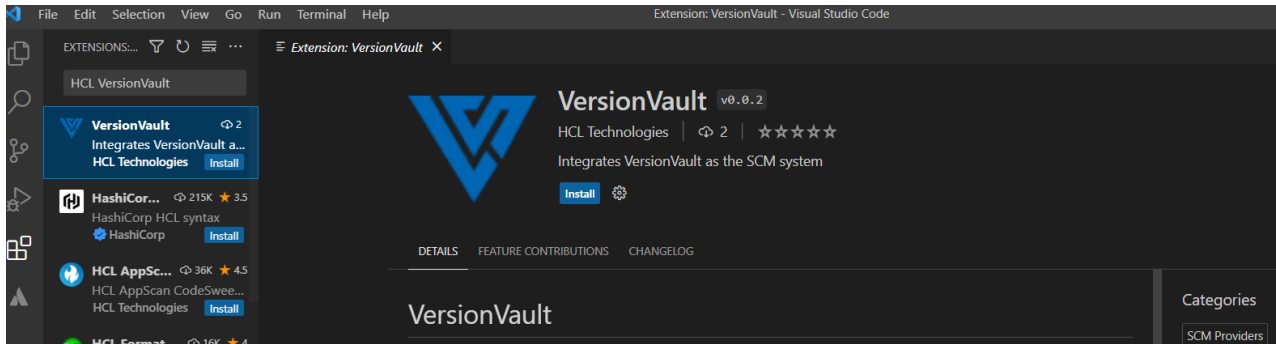
Figure 1 HCL VersionVault extension installation

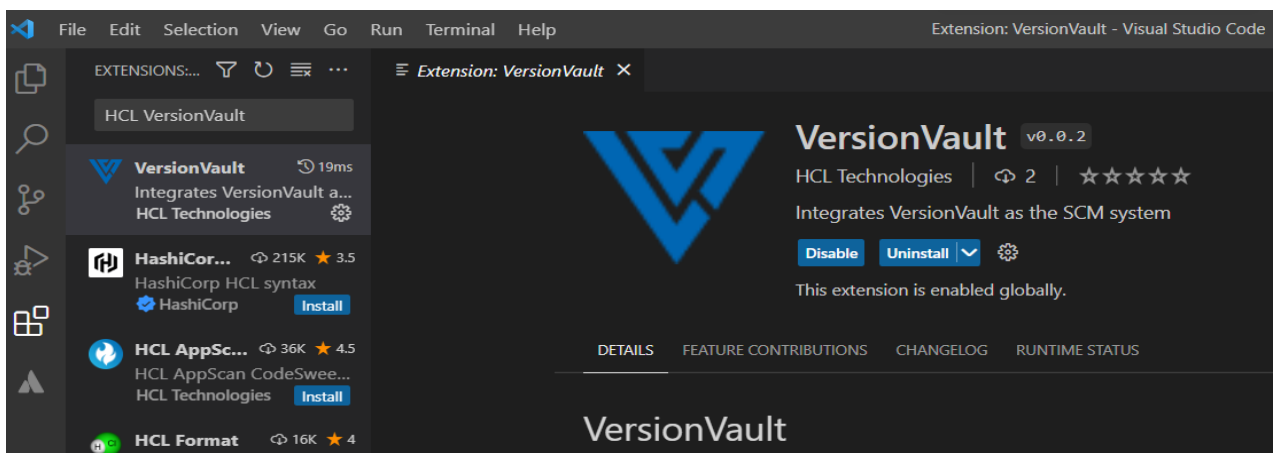Post HCL VersionVault extension installation:

Figure 2 HCL VersionVault extension installed

2. VersionVault Settings Page

You can configure options related to VersionVault in the Settings page. To open the Settings page, navigate to **File->Preferences->Settings->Extensions->VersionVault.**
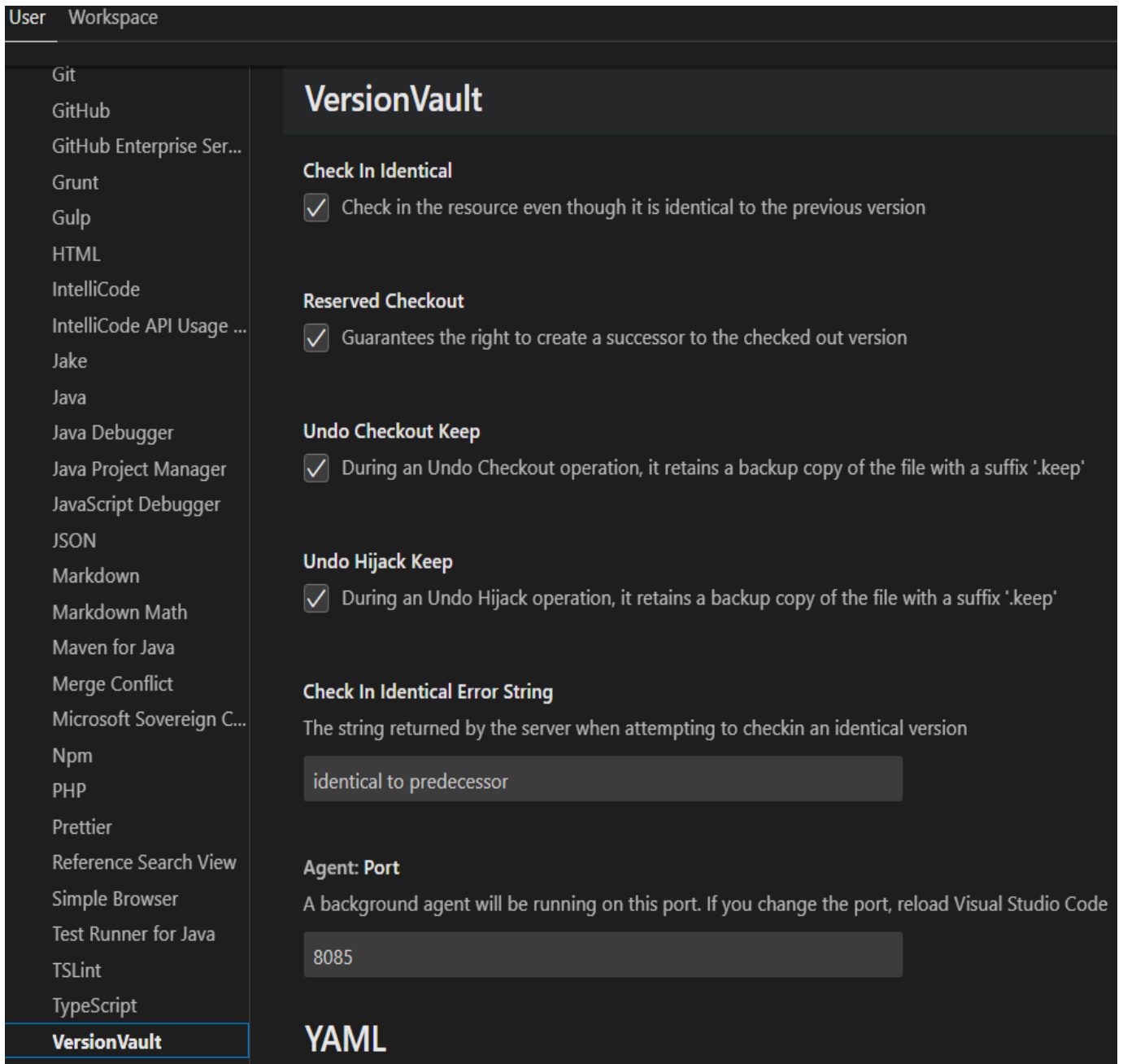
Figure 3 VersionVault Setting page

Check In Identical

If you want to check in the file even though it is identical to the previous version enable this option; otherwise, disable it.

Check In Identical Error String

To see a user-friendly error message, configure the error or partial error string returned by the server when attempting to checkin an identical version. If you do not configure or provide a different error string, you will see the default English error message returned by the server.

Agent Port

The **HCL VersionVault** extension uses a local background agent that will be running on this port. If you change the default port, reload the Visual Studio Code IDE. If the new port is available, the agent will start on it; otherwise, you are prompted to enter another available port.

Reserved Checkout

During a **Check Out** operation, if you want to do reserved checkouts, enable this option; otherwise, disable it.

Undo Checkout Keep

During an **Undo Checkout** operation, if you want to retain a backup copy of the file with a suffix '.keep', enable this option; otherwise, disable it.

Undo Hijack Keep

During an **Undo Hijack** operation, if you want to retain a backup copy of the file with a suffix '**.keep**', enable this option; otherwise, disable it.

3. VersionVault Operations

3.1 Add Existing View

VersionVault views must be created using VersionVault tools. Adding or importing an existing view can be done in two ways.

a. Using the "**Add Existing View**" Command Palette:

Click **View->Command Palette** menu option to open the command palette.
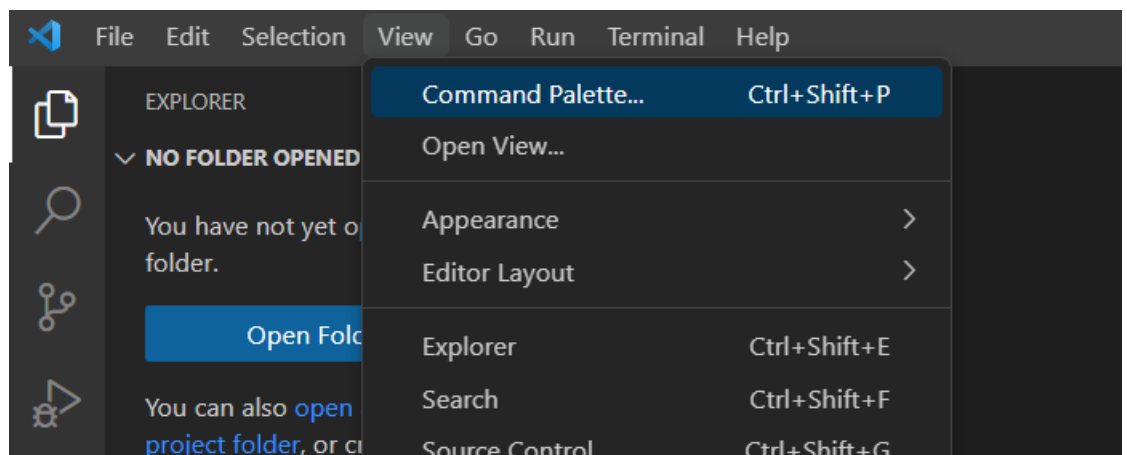
Figure 4 Command Palette

Enter "**Add Existing View**" text in search box and hit enter or click on listed "**Add Existing View…**" option.
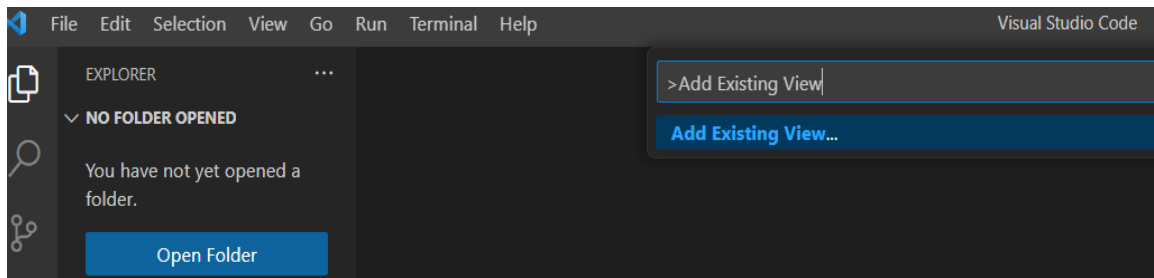
Figure 5 Add Existing View command palette

It opens a new dialog box. Navigate to a directory within the VersionVault view and click on **'Open View'** button to add the view path into Visual Studio Code IDE.
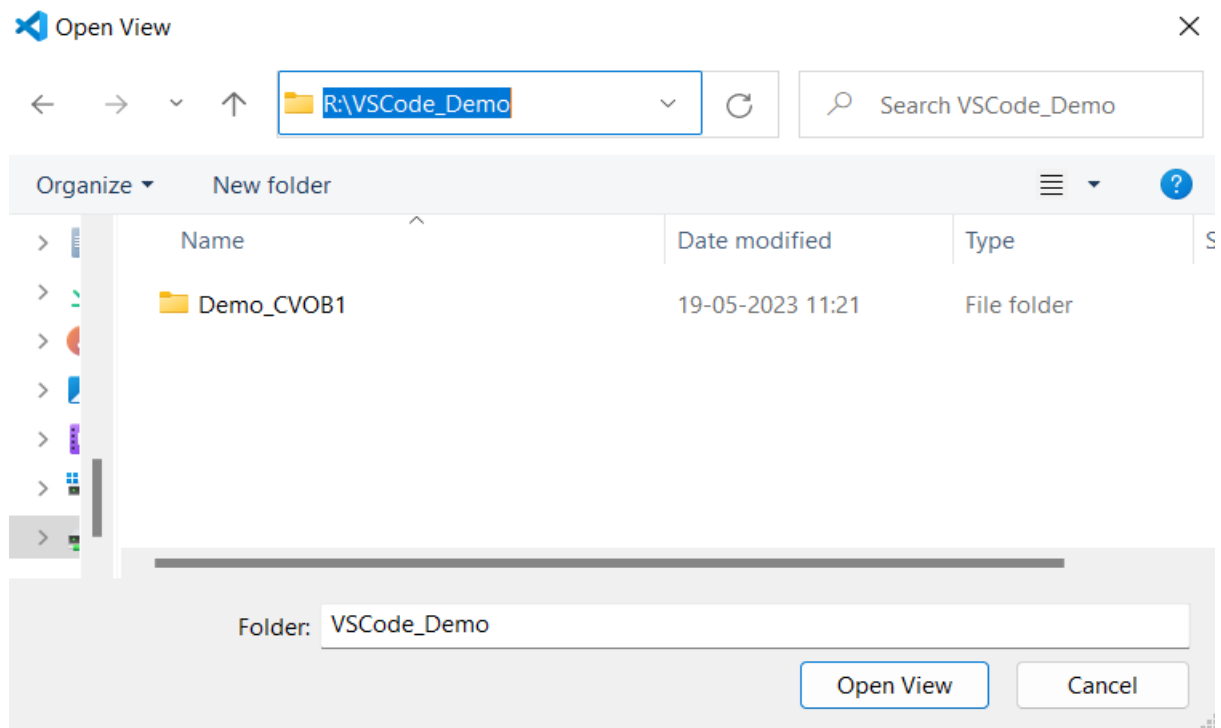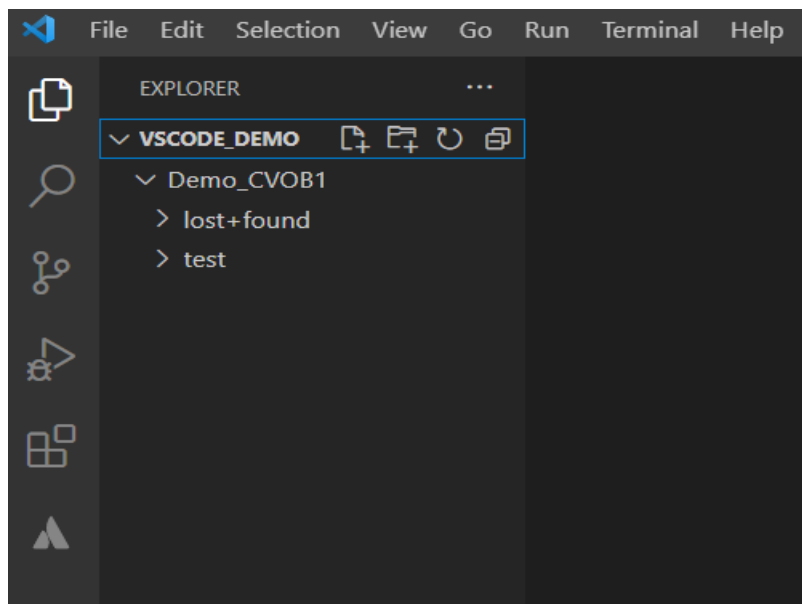

Figure 6 Select View Path

Figure 7 View Path imported in Visual Studio Code

b. Add using "**Open Folder**" option:
Click **File->Open Folder** menu option, which opens a new dialog box. Navigate to the root directory of the view and click on **'Select Folder'** button to add the view path into Visual Studio Code IDE.
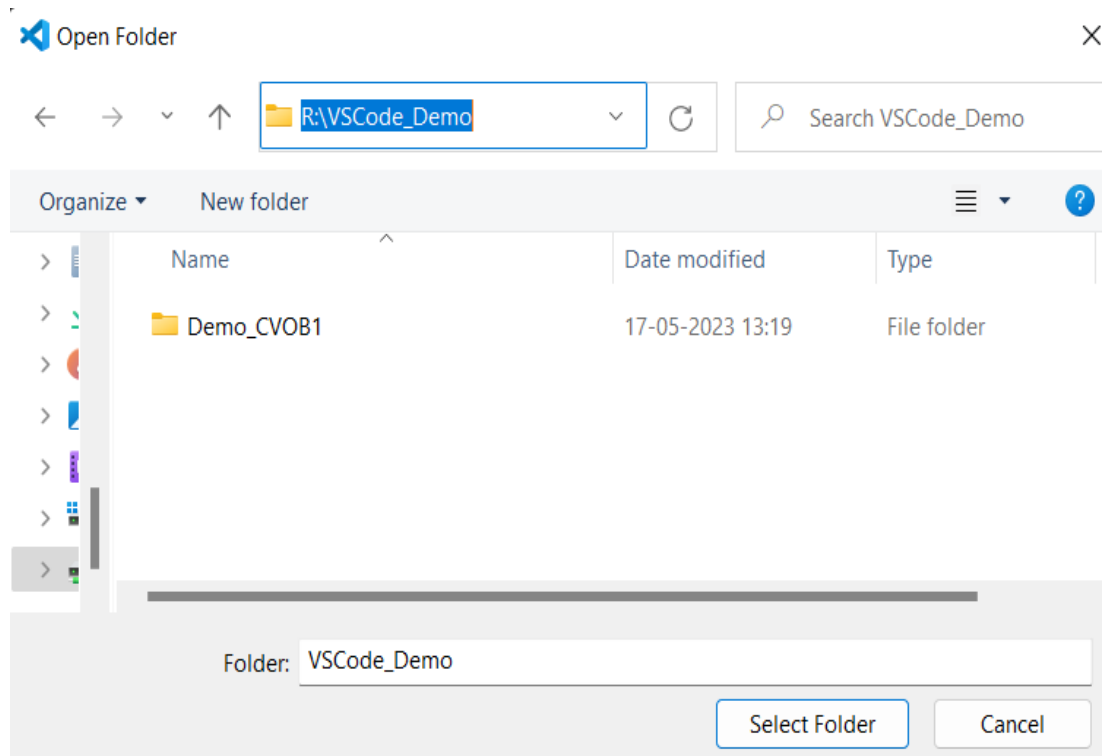
Figure 8 Select View Path folder
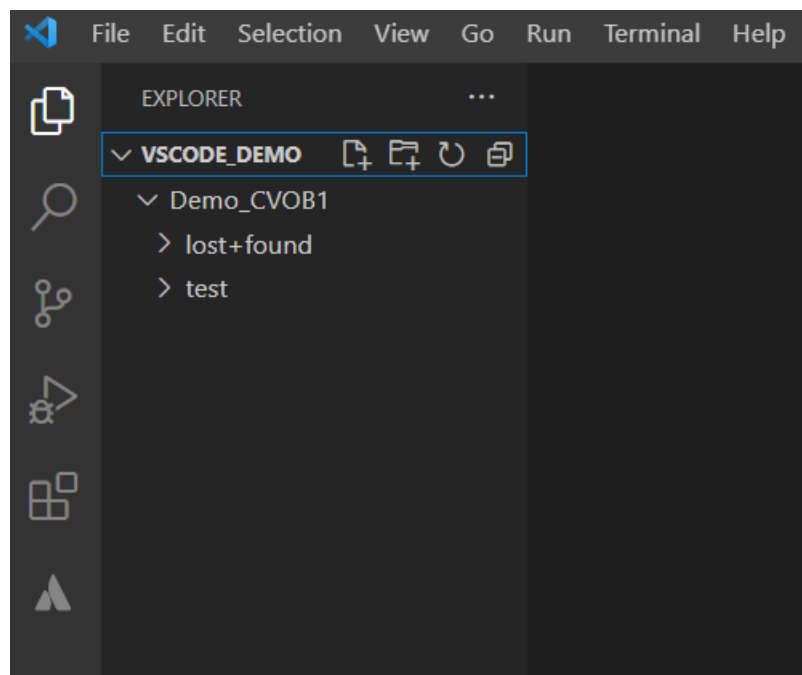
Figure 9 View Path imported in Visual Studio Code

3.2 Connect to WAN Server

Since this integration supports WAN based view types (Web and Automatic), the user needs to authenticate with the VVRC WAN server. Select the **Source Control** icon from left panel, then click the "more" option **...** (three dots), which shows the **VersionVault->Connect to WAN Server** option.
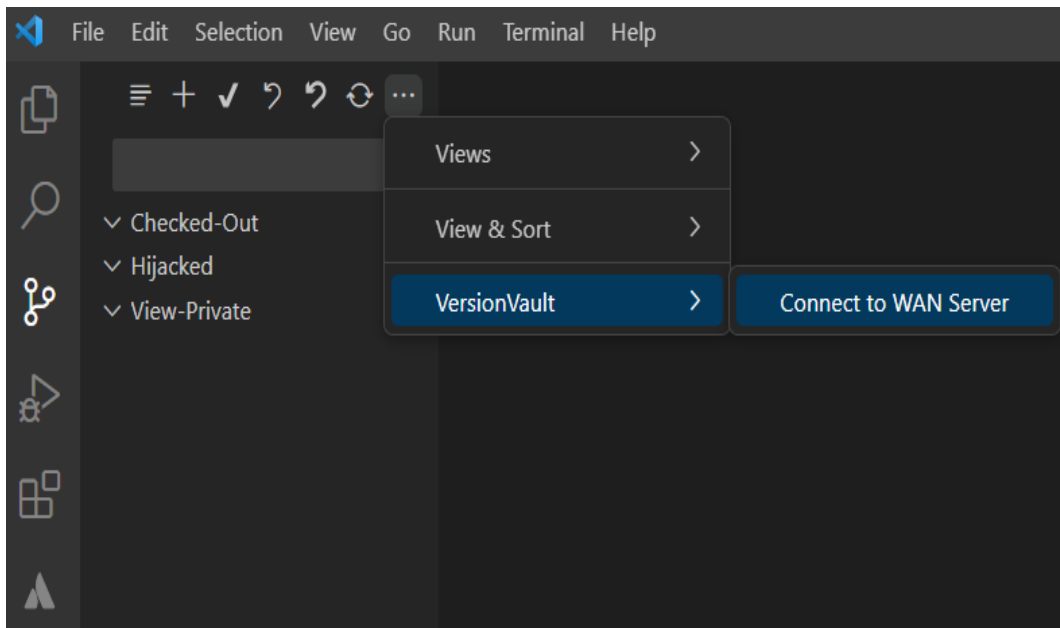
Figure 10 Connect to WAN Server

Click on the **VersionVault->Connect to WAN Server** option which will prompt to enter the WAN server URL and credentials. Enter the VVRC WAN server URL, User Name and Password to authenticate.
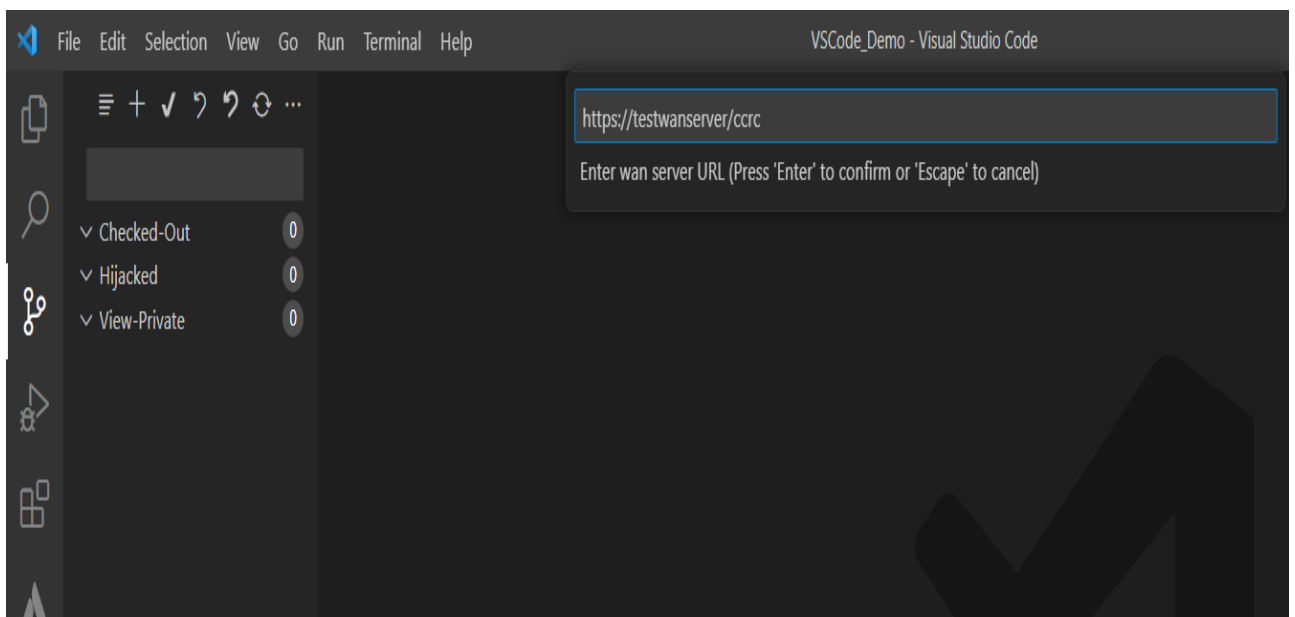
Figure 11 VVRC WAN server URL
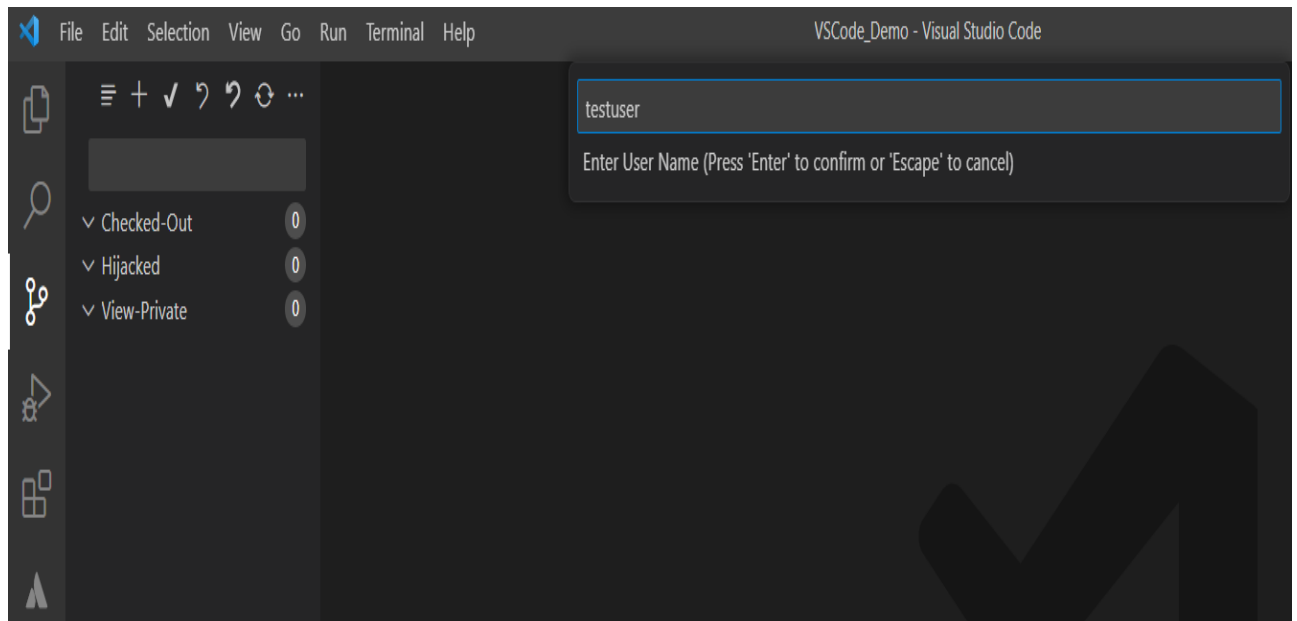
Figure 12 User Name
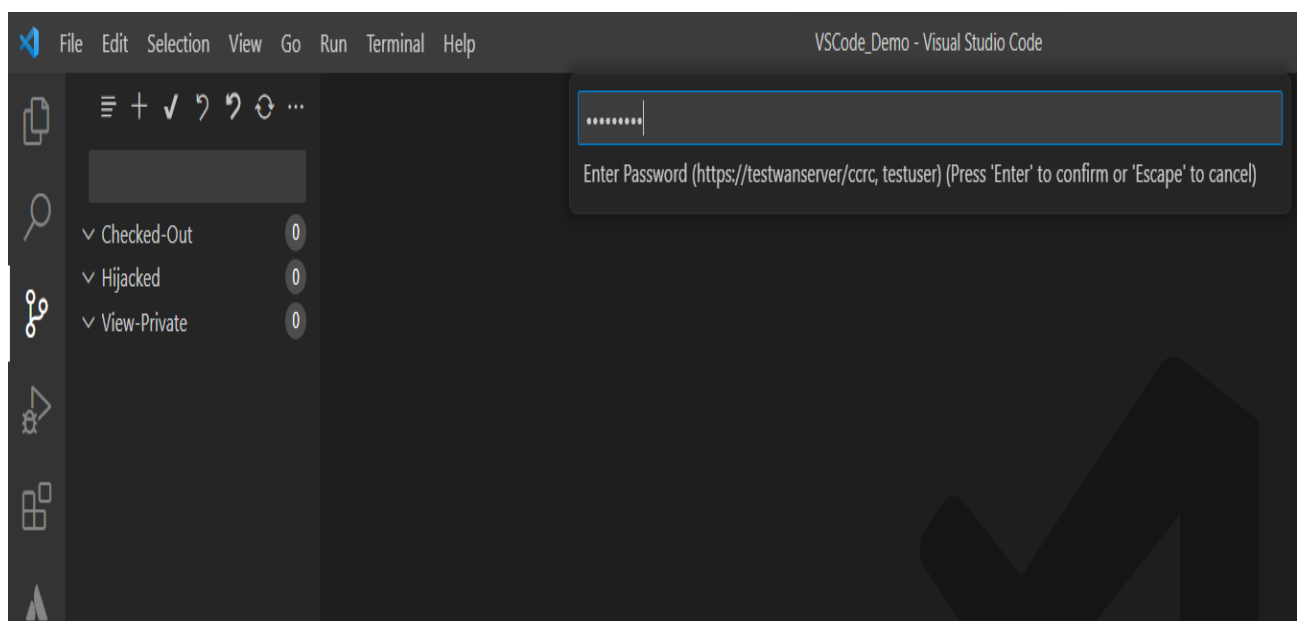
Figure 13 Password

You will see an authentication successful notification message at the bottom.
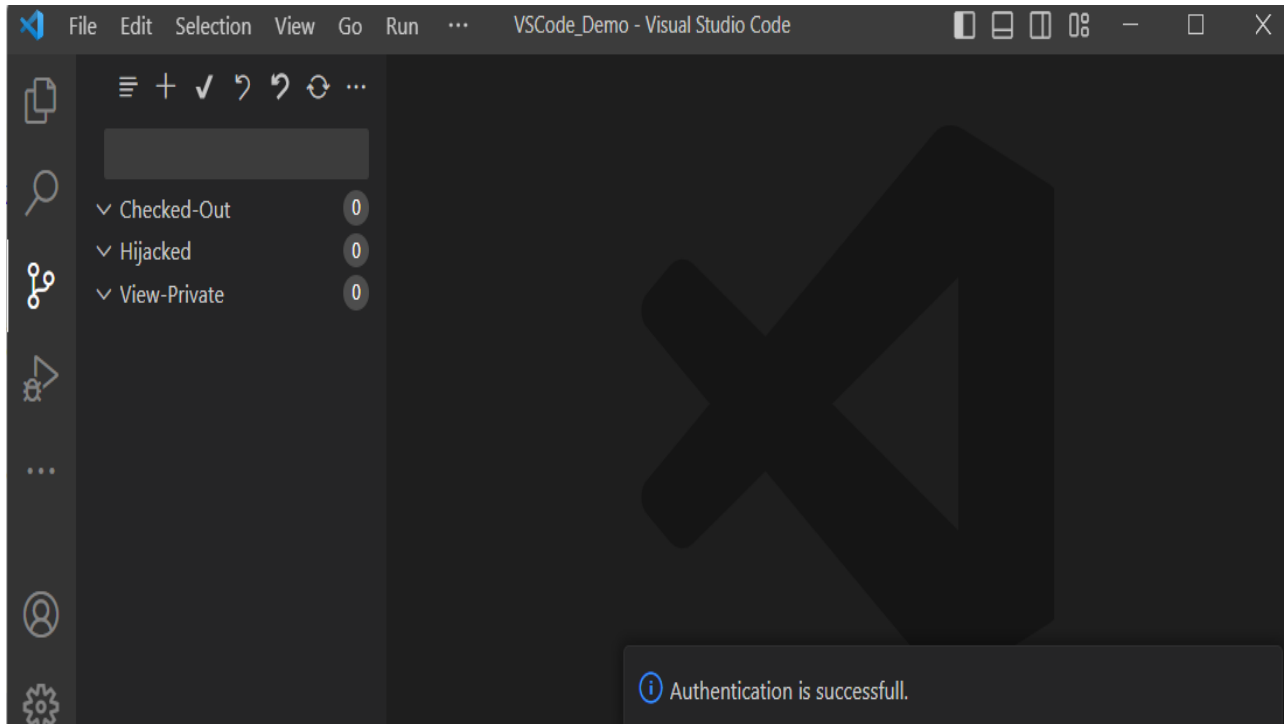
Figure 14 Authentication is successful.

Note: If you are connecting to a VVRC WAN server over HTTPS, you might encounter an SSL certificate issue. In case the certificate is not stored and trusted on the client, you must use the VersionVault Explorer or rcleartool to establish a connection with the VVRC WAN server and permanently accept the certificate of the WAN server. This step is necessary before proceeding with using Visual Studio Code.

3.3 Create Activity

For a UCM view you need to create an activity and set it in the view. Select the **Source Control** icon from left panel, then click the "more" option **…** (three dots) and click the **VersionVault->Create Activity…** option.
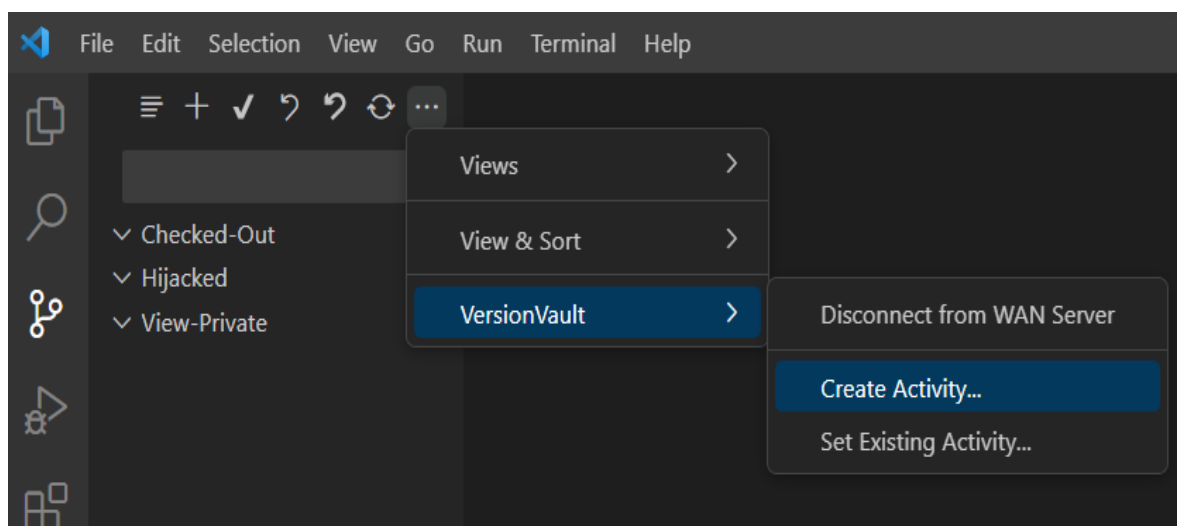
Figure 15 Create Activity

Enter the Activity Headline
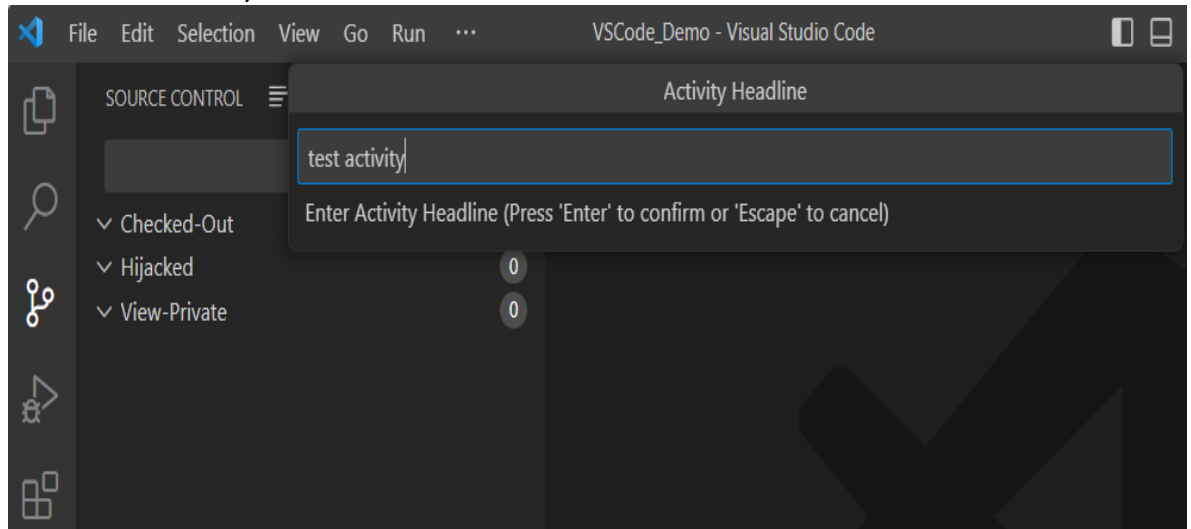
Figure 16 Activity Headline

Enter the Activity ID
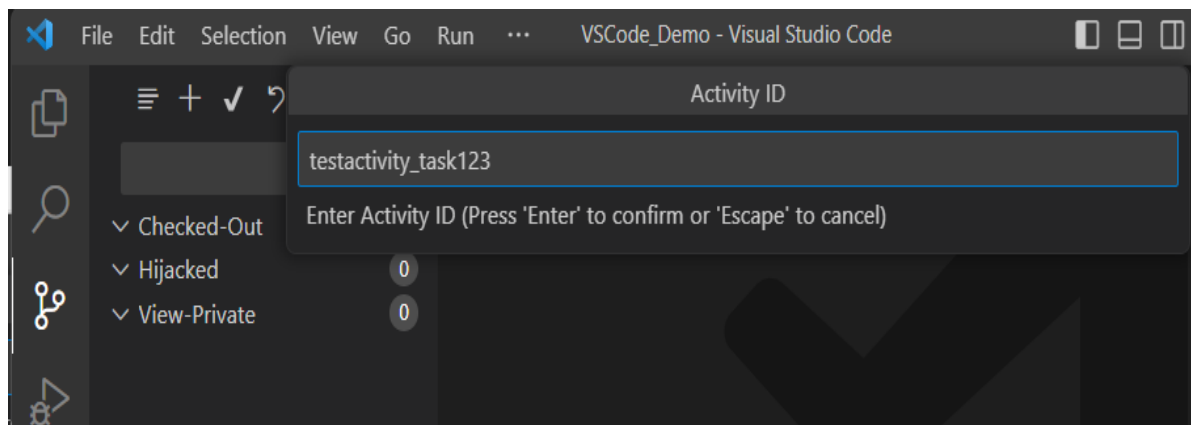
Figure 17 Activity ID

Note: Create Activity will also set the activity in the view
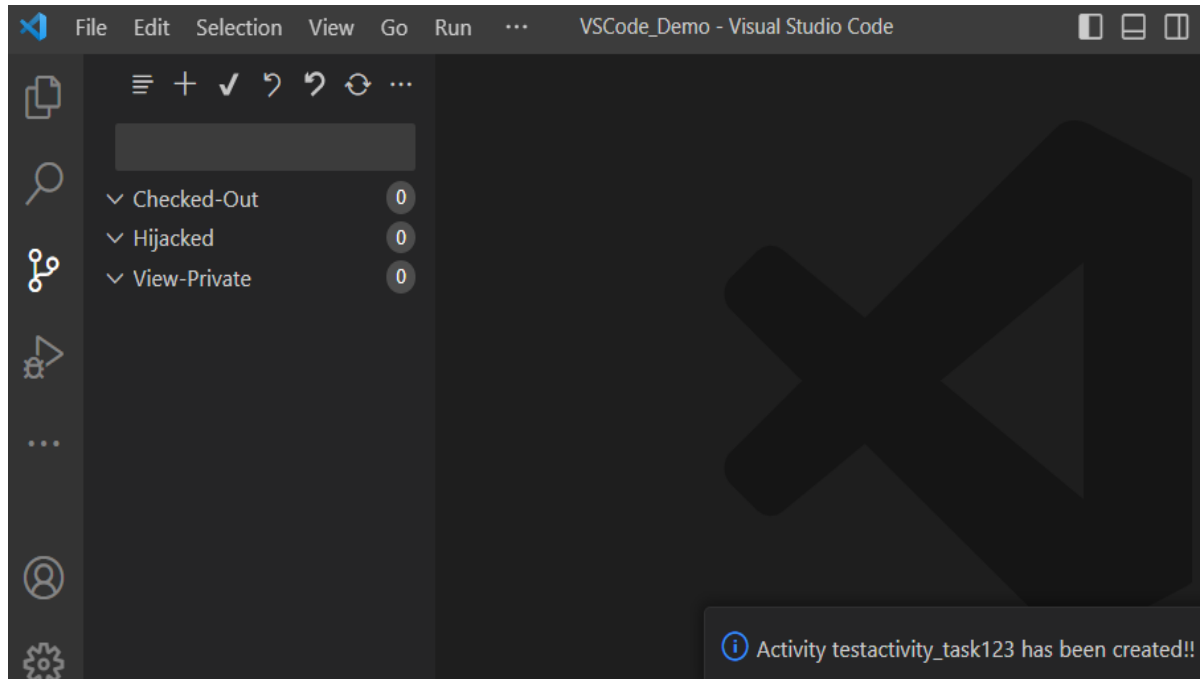
Figure 18 Activity is created

3.4 Set Activity

You can set the activity in the current imported view. Select the **Source Control** icon from the left panel, then click the "more" option **…** (three dots) and click the **VersionVault->Set Existing Activity…** option.
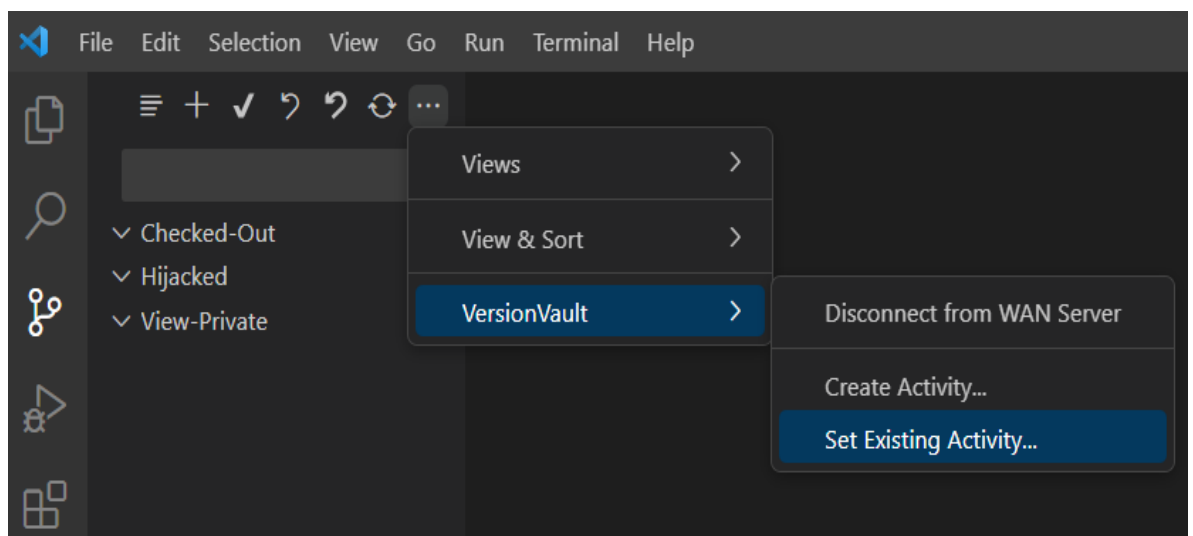

Figure 19 Set Existing Activity

Select the activity and hit enter to set the activity.
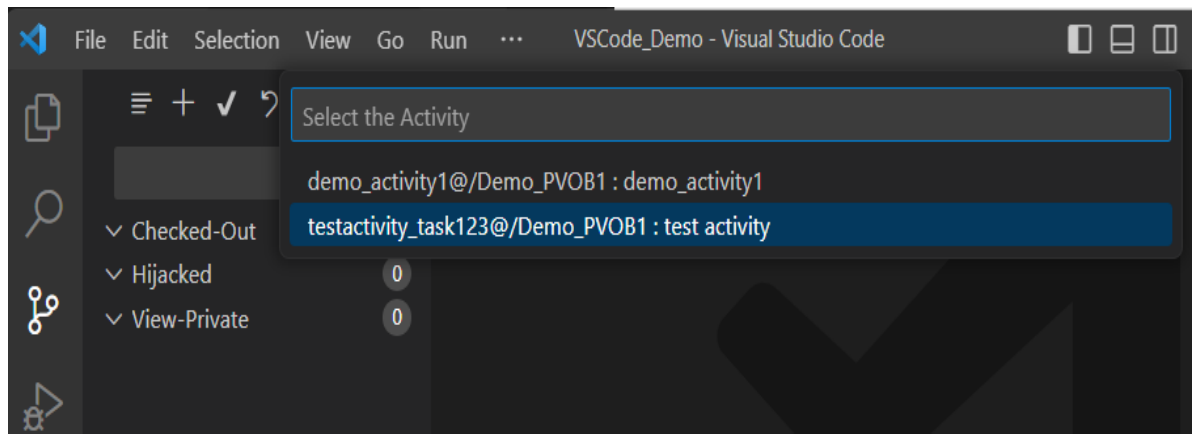
Figure 20 List of Existing Activities
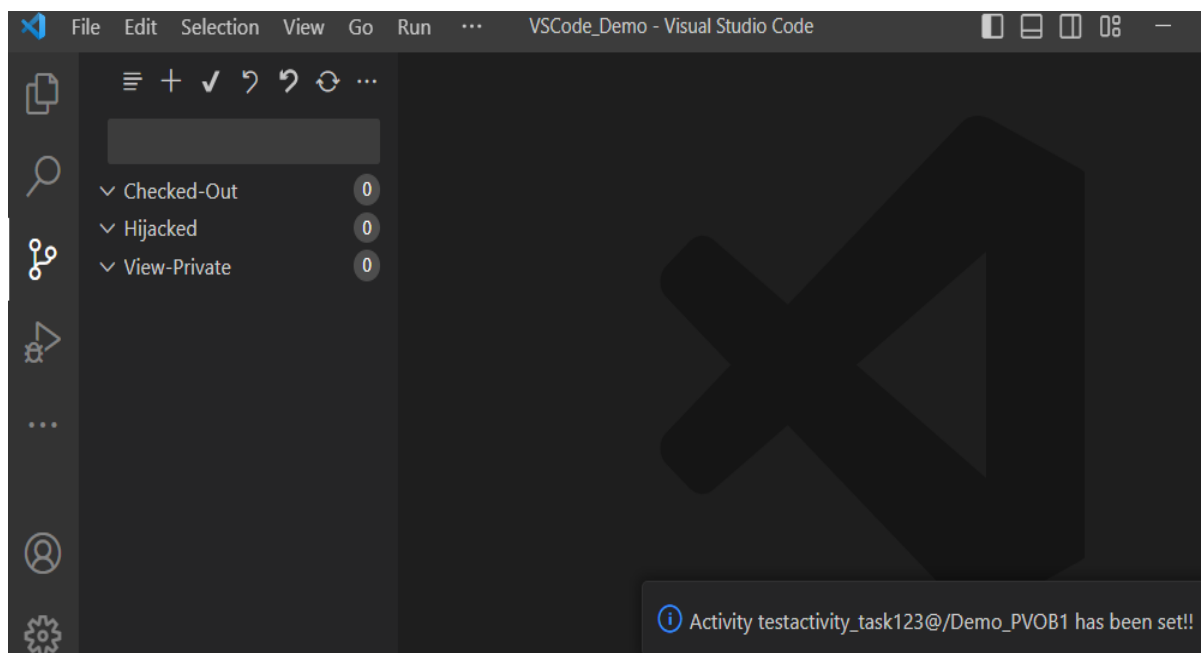
Figure 21 Activity has been set into view

Note: You can also execute the **Create** and **Set Activity** operations from the file editor context menu. Open a file in an editor and right click. You should see **VersionVault->Create Activity…** and **VersionVault-> Set Existing Activity…**
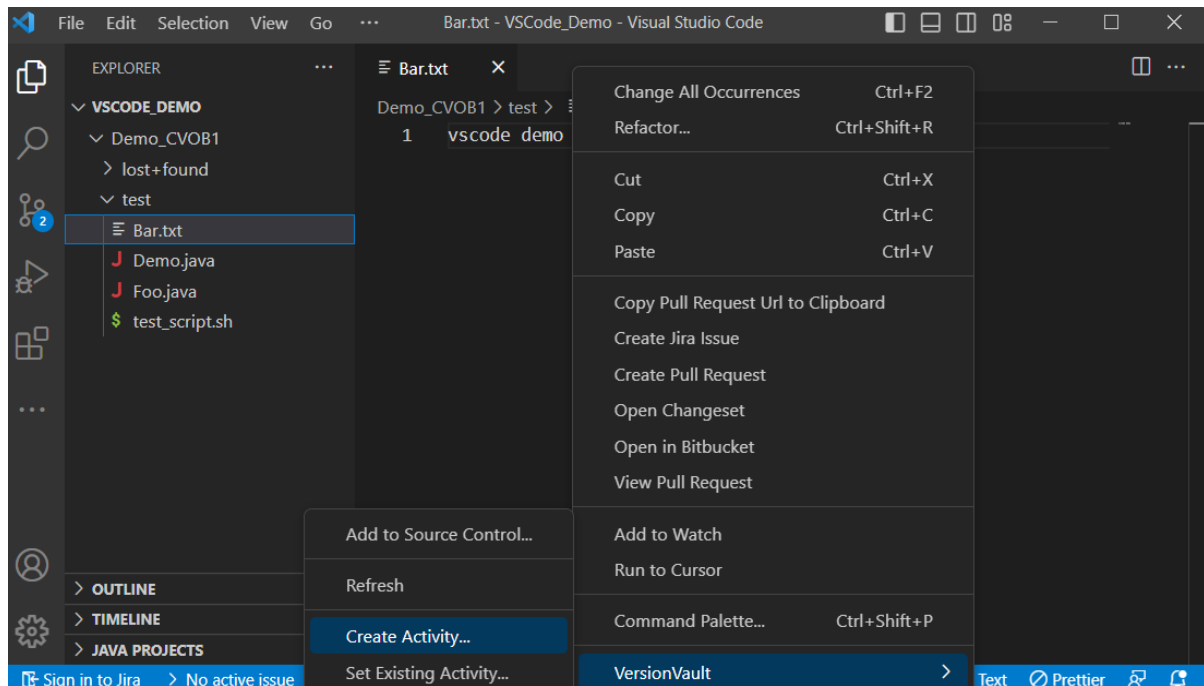
Figure 22 Create and Set Existing Activity

3.5 Add To Source Control

You can add new files/folders to a VersionVault source control repository (VOB). New resources are called **view private** files/folders. In the Visual Studio Code Explorer, you can create new files/folders. When you initially create files/folders, these resources will be available under **View-Private** section in **Source Control** page.
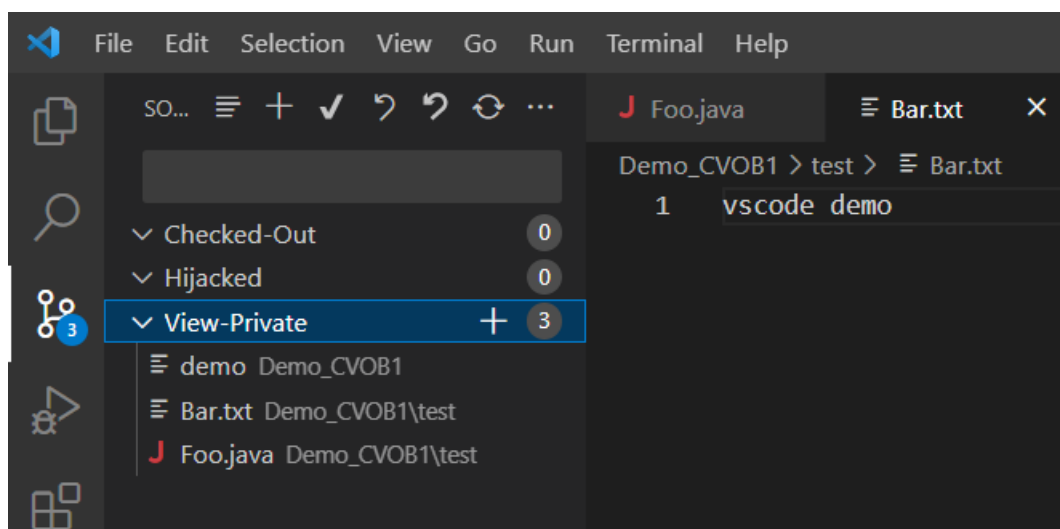
Figure 23 View Private files/folders list

Once you have created a new file/folder, you likely want to bring it under source control, there are two places from where you could execute "**Add To Source**" operation.

a. File Context Menu

Open the view private file and right click in file editor, select the **VersionVault->Add to Source Control…** option which prompts you for a comment. Using this option, you can add a single file to source control.
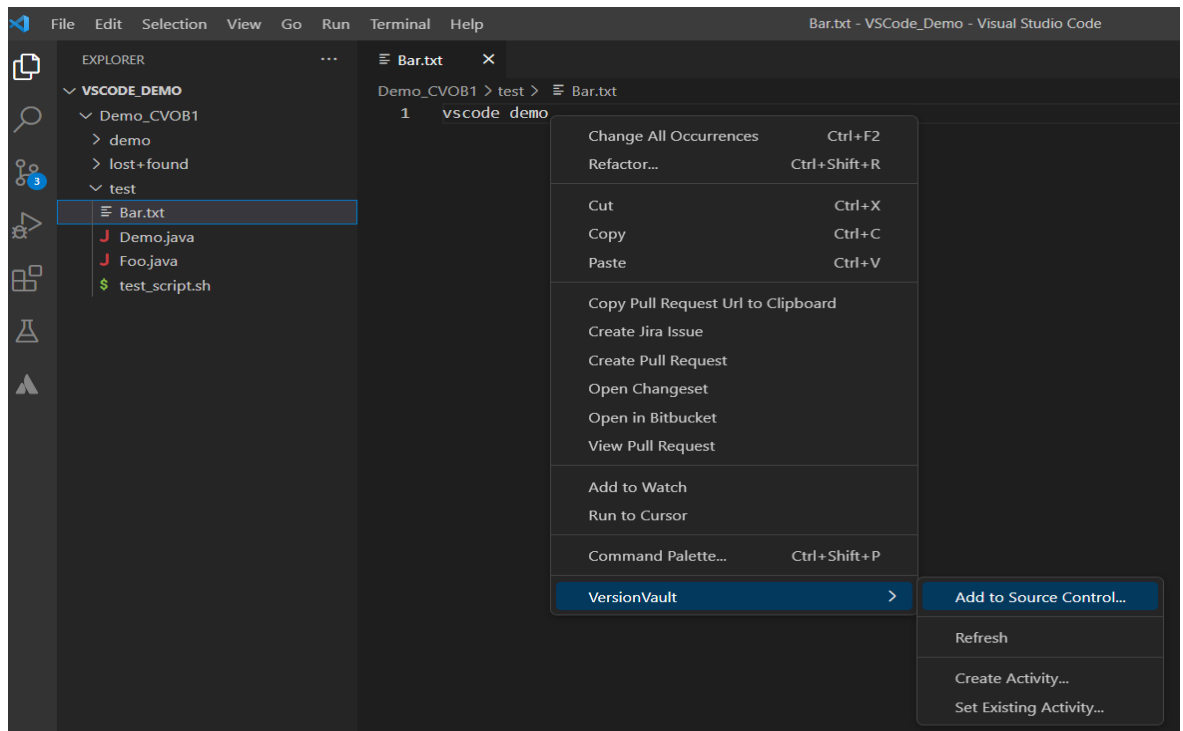
Figure 24 File Context Menu - Add to Source Control...

Enter the comment and hit enter to add the file to source control.
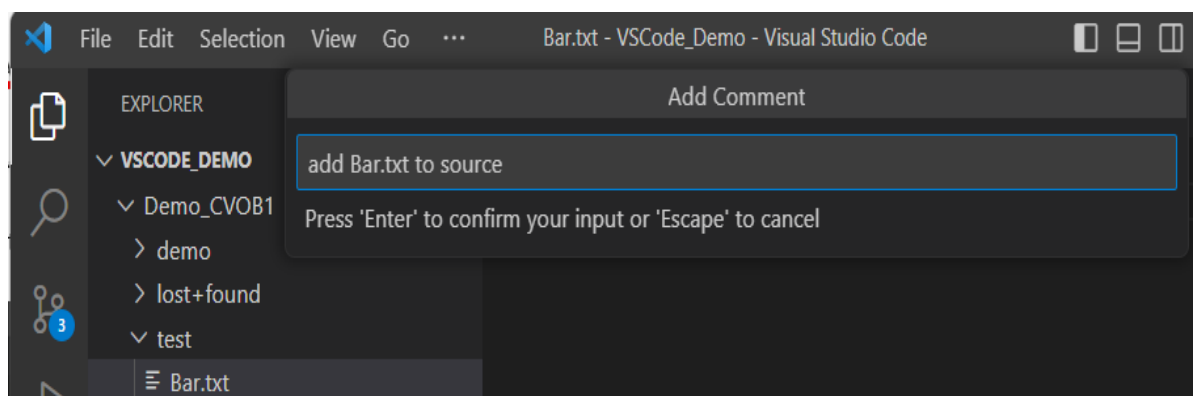
Figure 25 Add to Source Comment

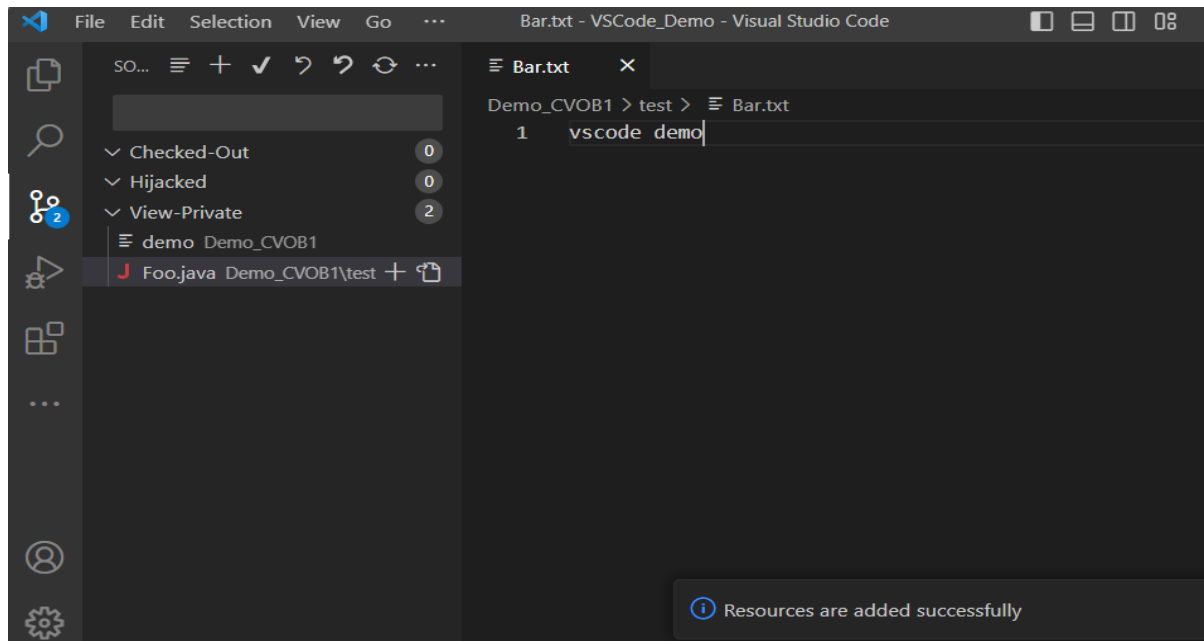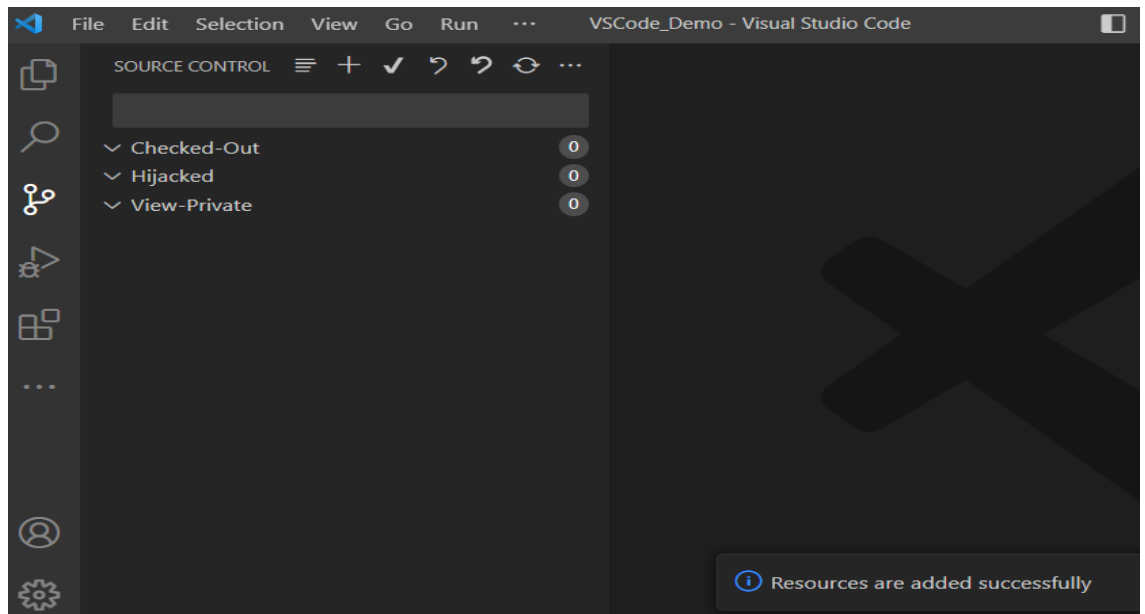After adding file to source control, the item will no longer appear in the **View-Private** section.

Figure 26 File Added to Source Control

b. Source Control Page (SCM)

Select the **Source Control** icon from the left panel to see a list of view private files under "**View-Private**". From here you can add a single, selected, or all view private files/folders to source control.
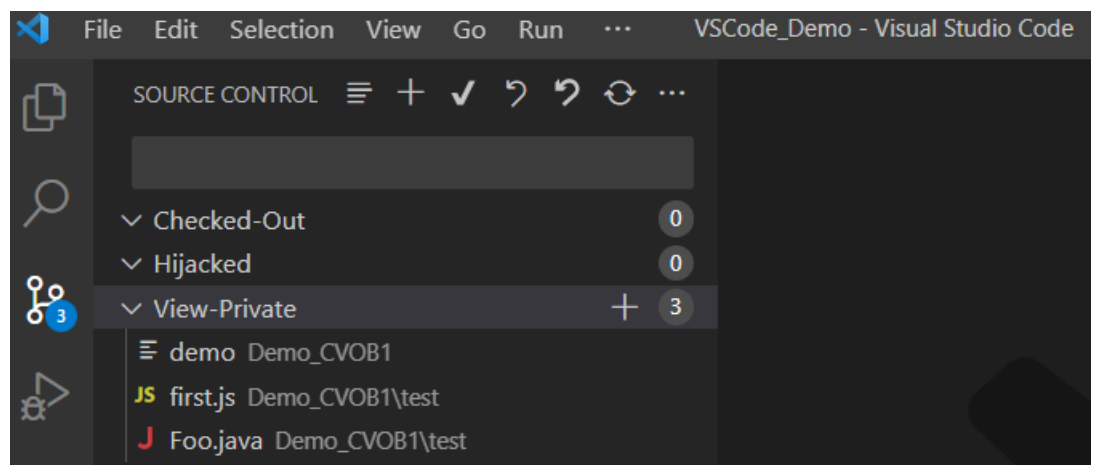
Figure 27 List of View Private files/folders

You can click the "**+**" symbol in front of the file/folder item under **View-Private** to add a single resource to source control.
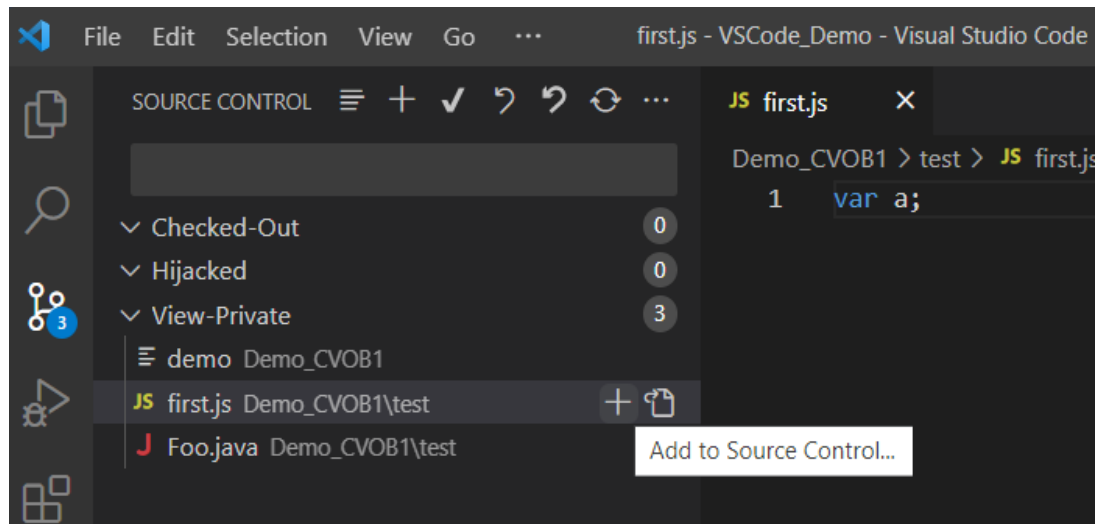
Figure 28 Adding single resource to source control

You can select the file/folder items under **View-Private** and right click on the selected items, then click "**Add to Source Control...**" to add files/folders to source control.
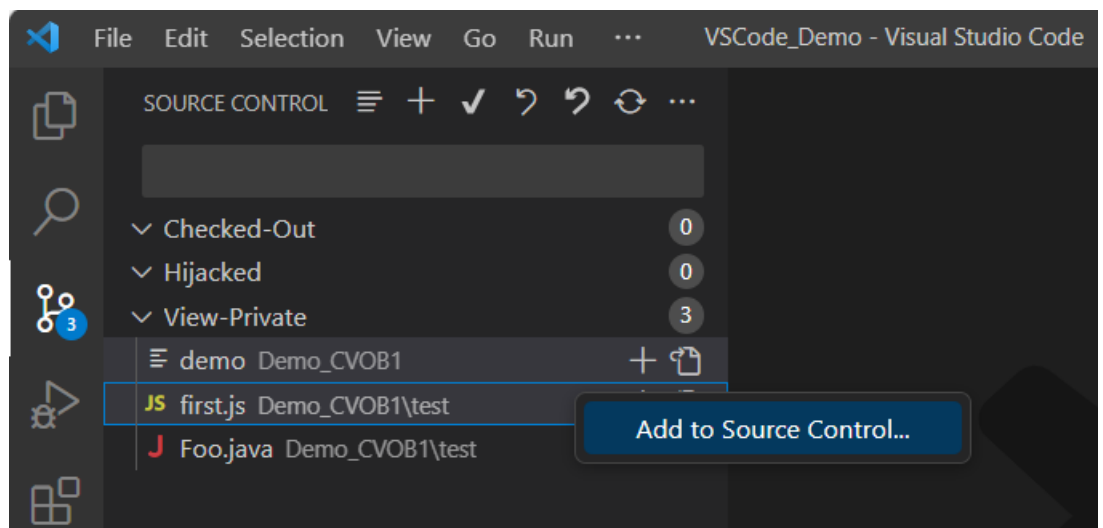
Figure 29 Add selected items to source control

To add all view-private files/folders to source control, click the "**+**" symbol in front of "**View-Private**" or "**SOURCE CONTROL**".
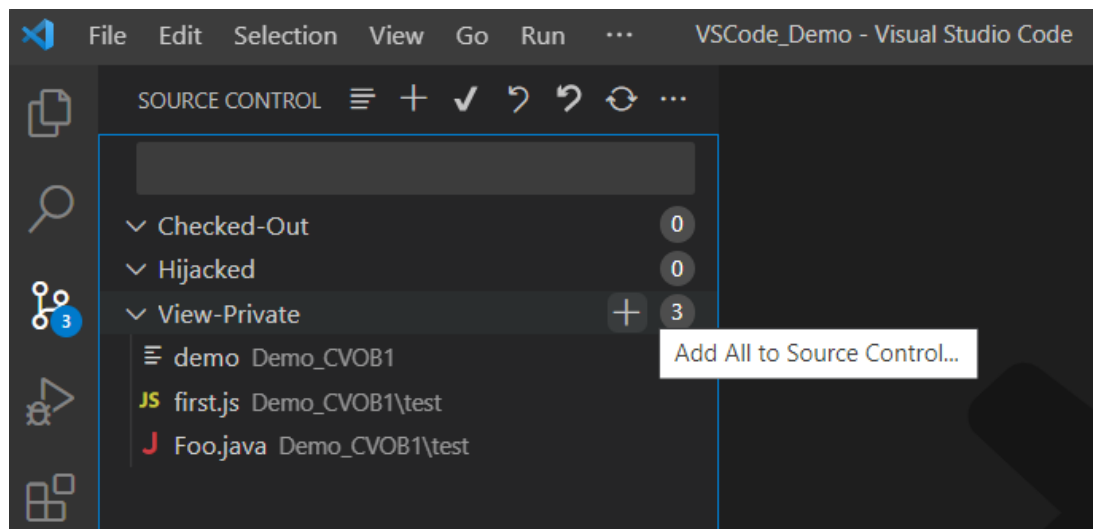
Figure 30 Add all view private items to source control (click '+' in front of View-Private)
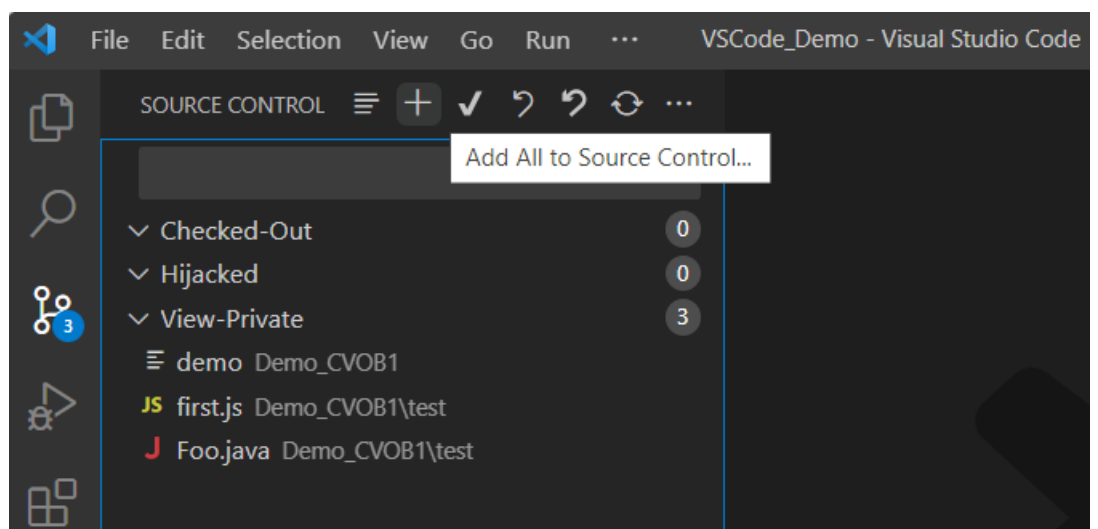
Figure 31 Add all view private items to source control (click '+' in front of SOURCE CONTROL)

Enter the comment and press enter to add the resources to source control.
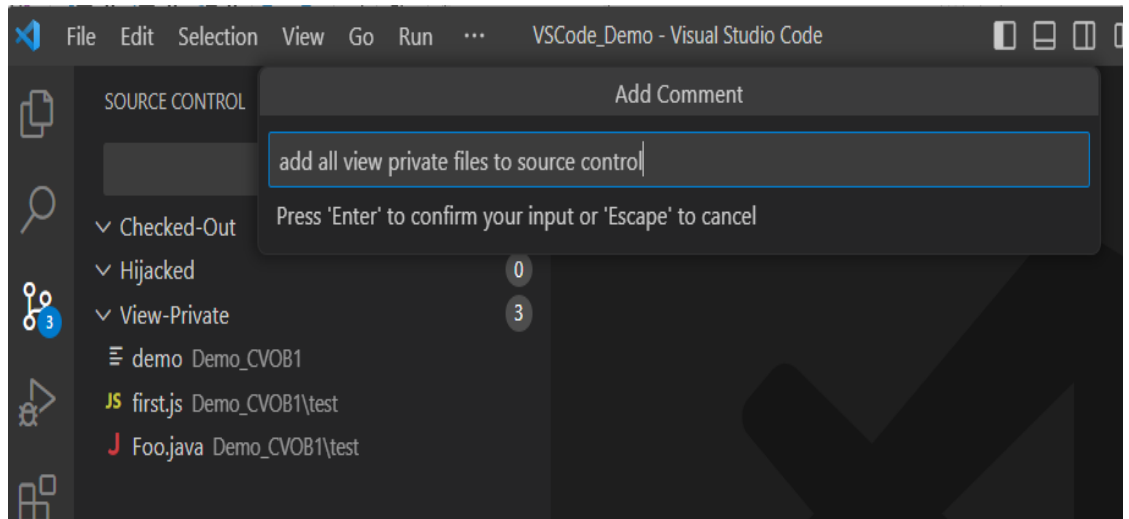
Figure 32 Add to Source Comment

Figure 33 Added resources to source control

After adding the resources to source control, the items will no longer appear in the **View-Private** section.

3.6 Check Out

To modify a source-controlled file, you must checkout or hijack the file. To check it out, open the file and right click in the file editor, select the **VersionVault->Check Out…** option which prompts for a comment.
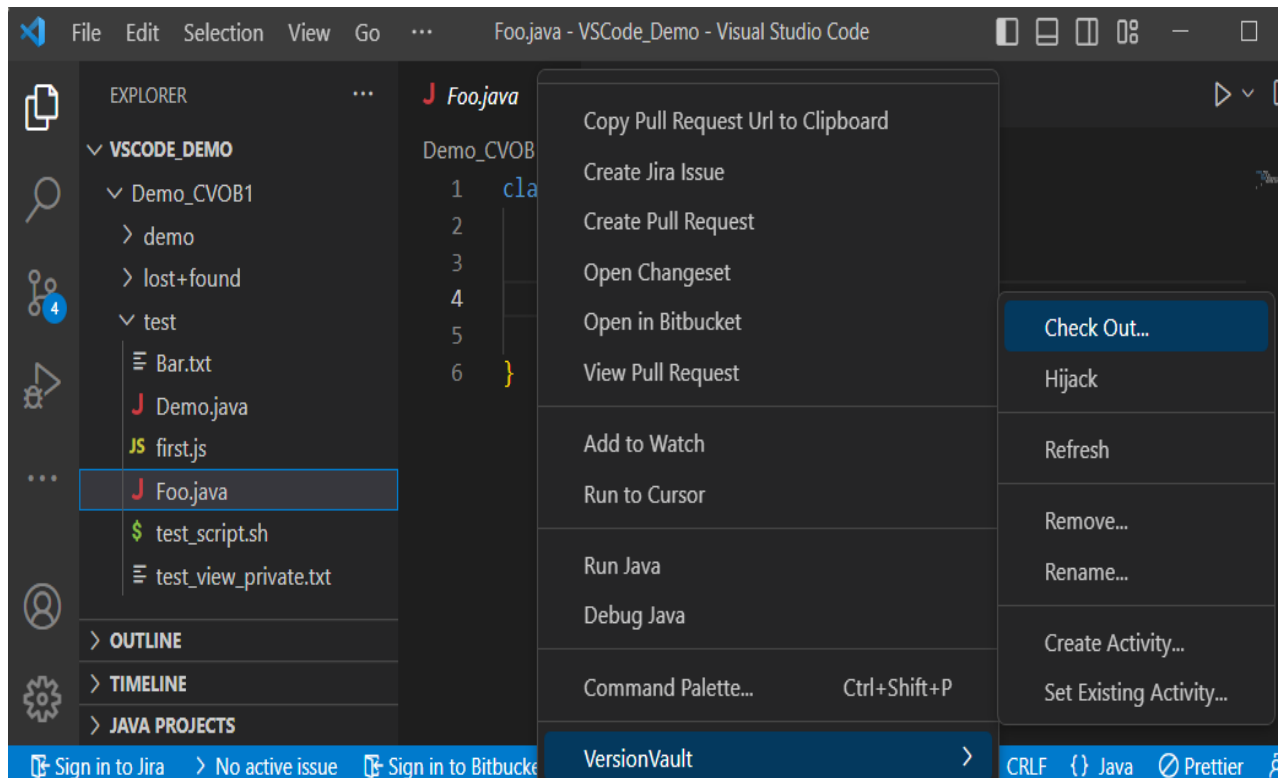
Figure 34 Checkout the resource

Enter the comment and press enter to check out the file.
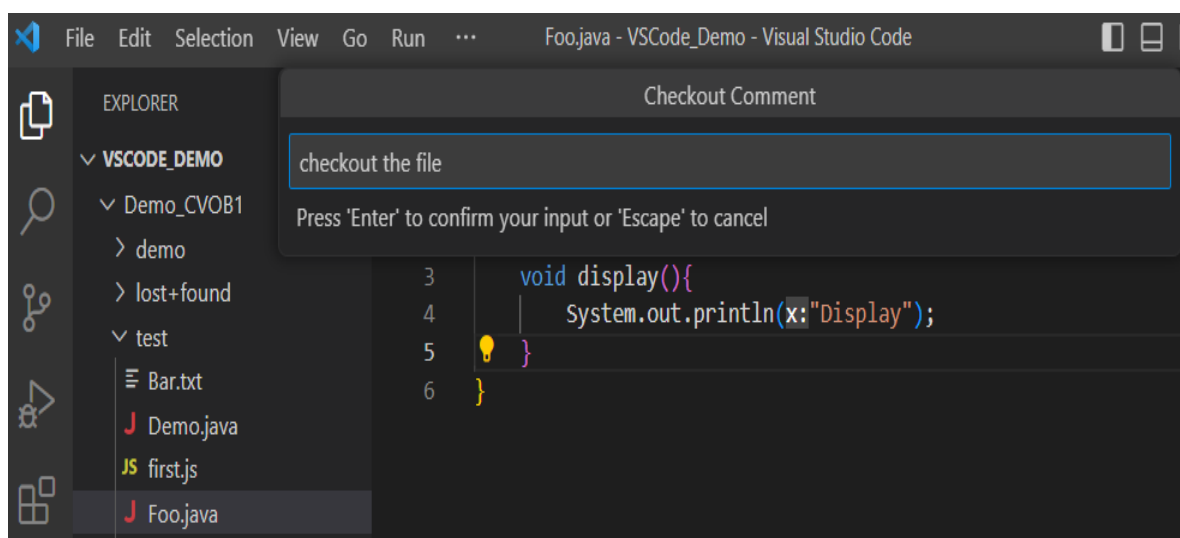
Figure 35 Checkout Comment

Post checkout, files will be listed under the **Checked-Out** section in the **Source Control** page.
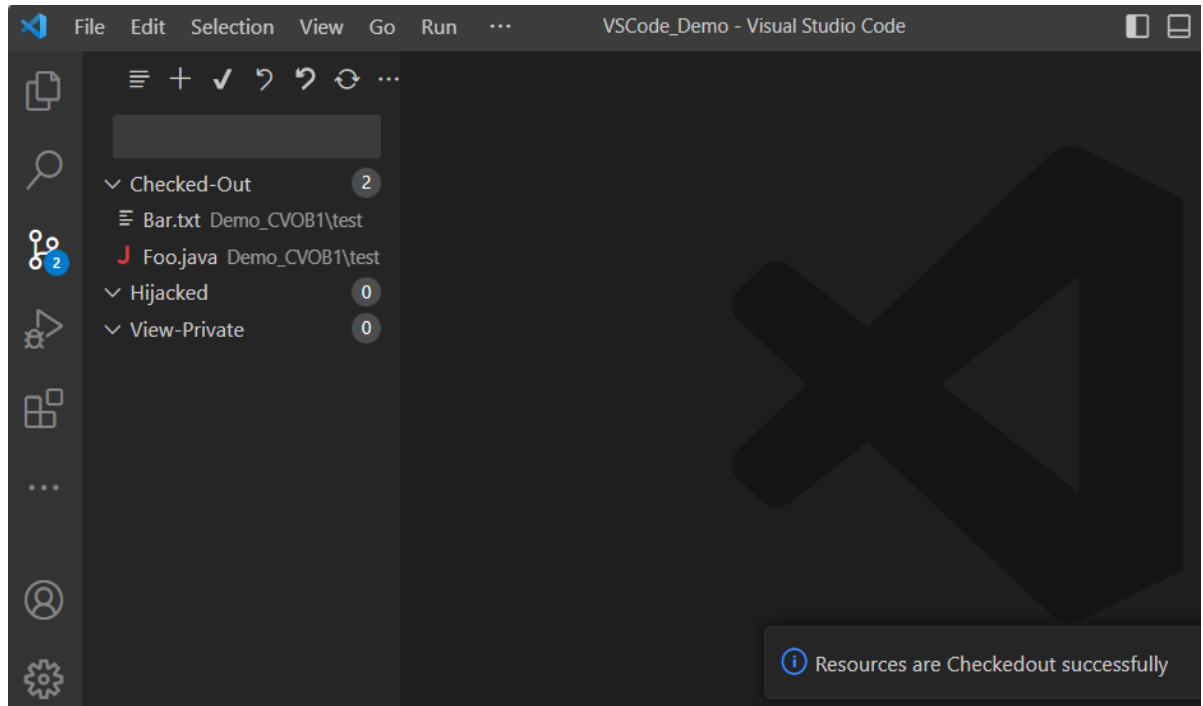
Figure 36 List of checked out resources

3.7 Check In

Check in the checked out files will create new versions in the VOB. There are two places from where you can execute a "**Check In…**" operation.

a. File Context Menu

Open the checked out file and right click in the file editor, select the **VersionVault->Check In…** option which prompts for a comment. Using this option, you can check in a single file.
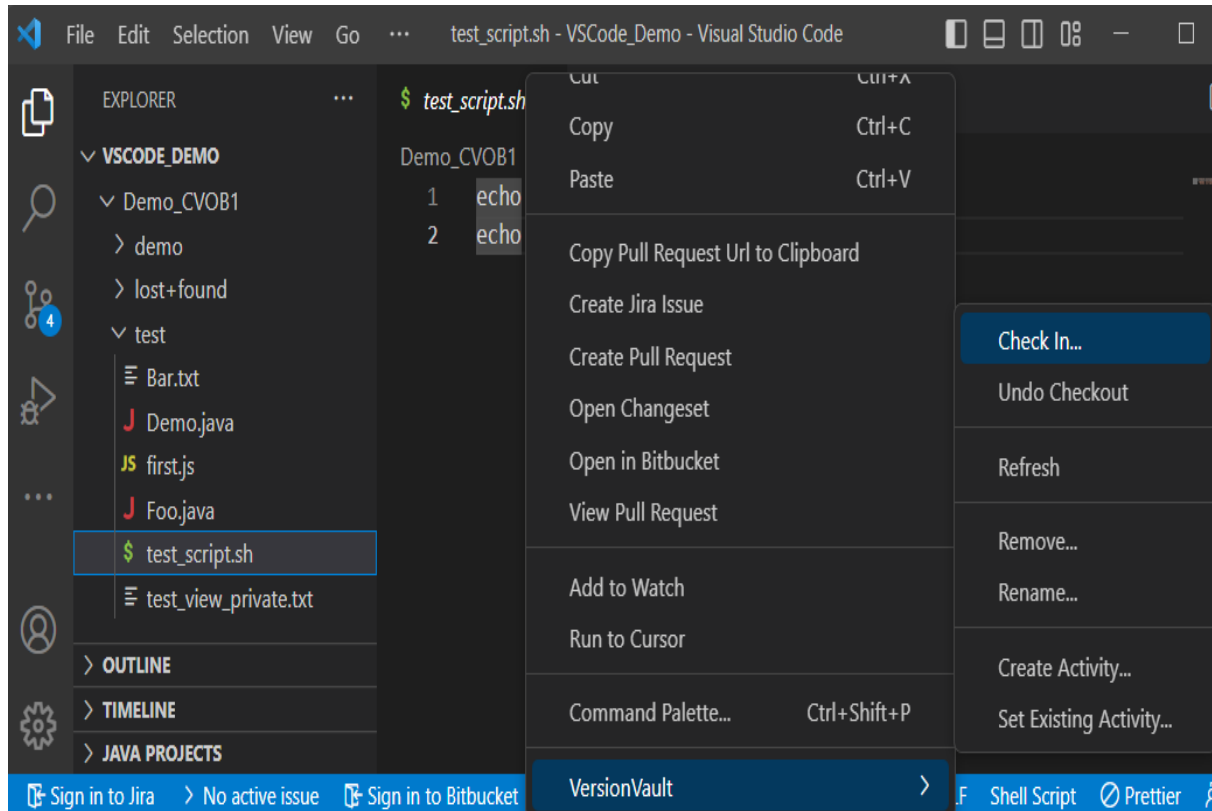
Figure 37 File Context Menu – Check In

Enter the comment and press enter to check in the file.
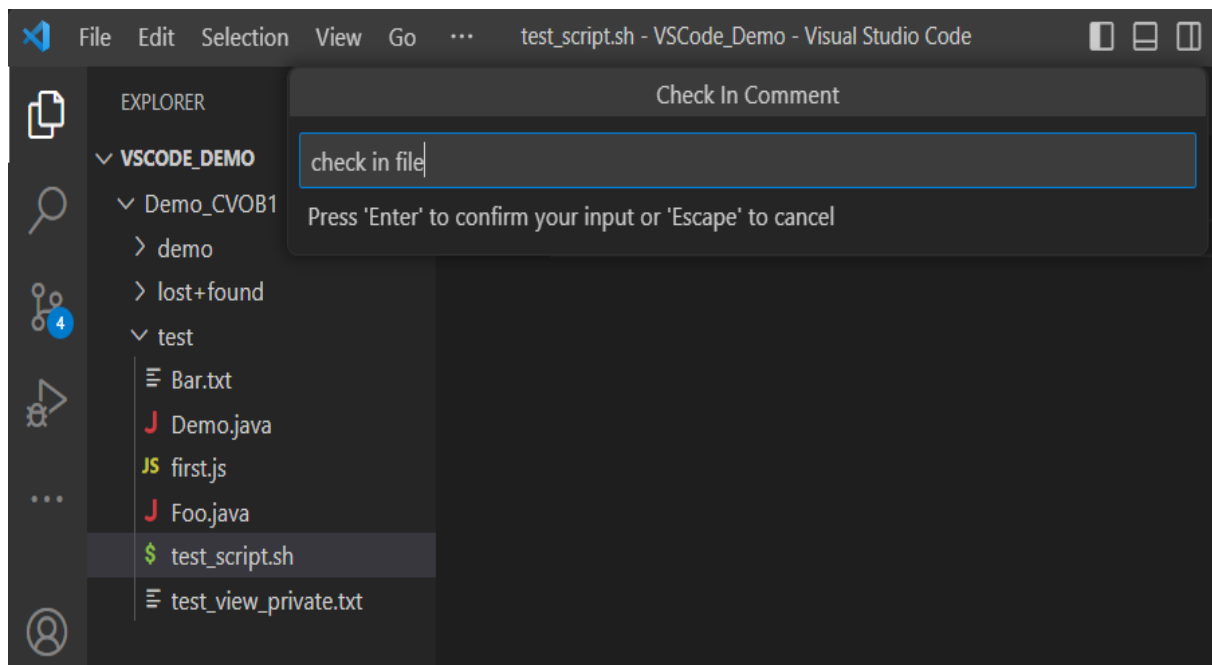
Figure 38 Check In Comment

After checkin, the file will no longer appear in the **Checked-Out** section in the **Source Control** page.
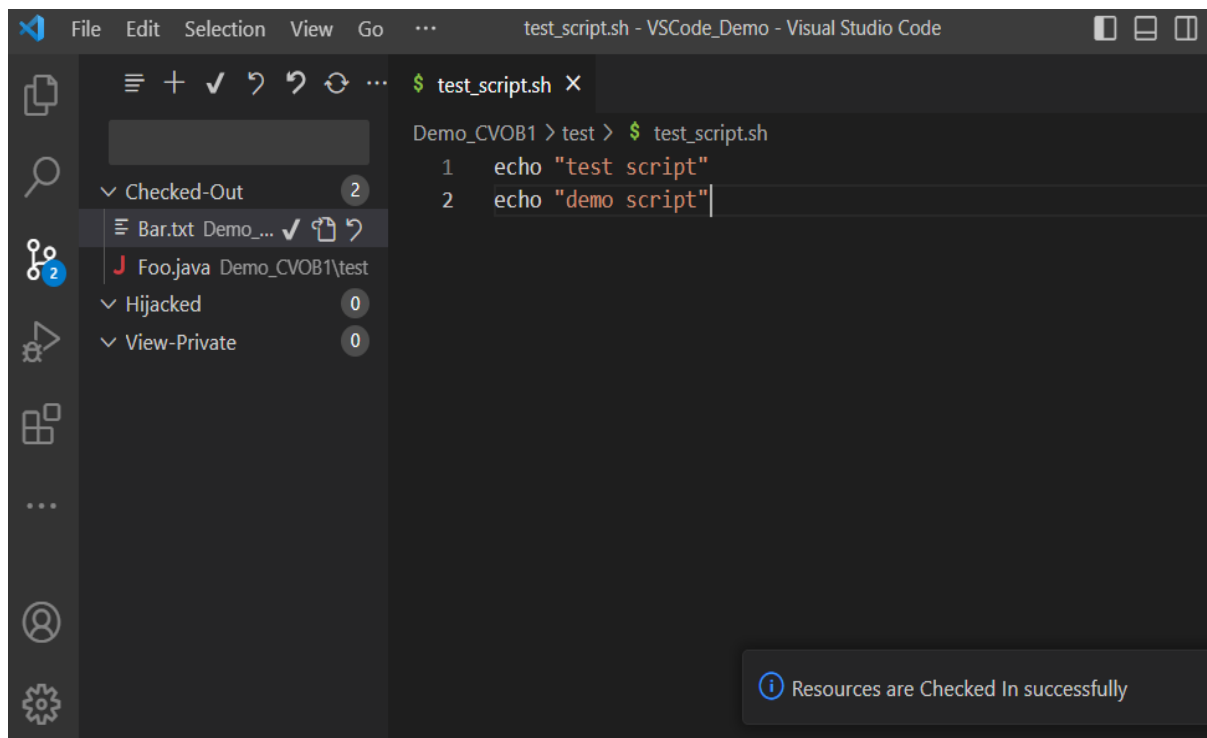
Figure 39 Resource is Checked In

b. Source Control Page (SCM)

Select the **Source Control** icon from the left panel, you will see list of checked out resources under "**Checked-Out**". From here you can check in single, selected, or all checked out files/folders.
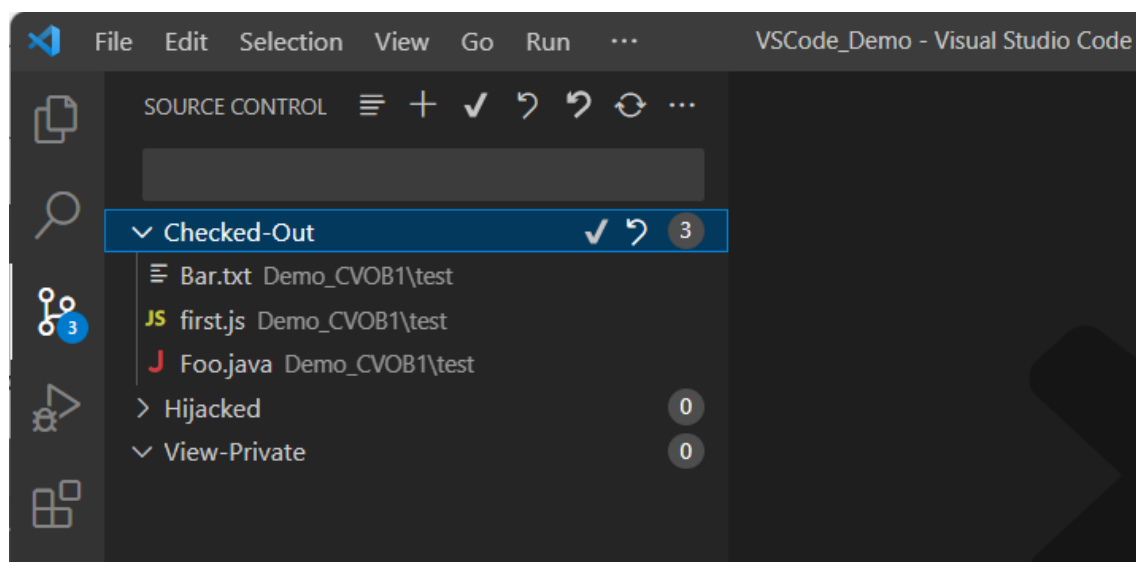
Figure 40 List of checked out files

You can click the "✓" symbol in front of file/folder items under **Checked-Out** to check in the file/folder.
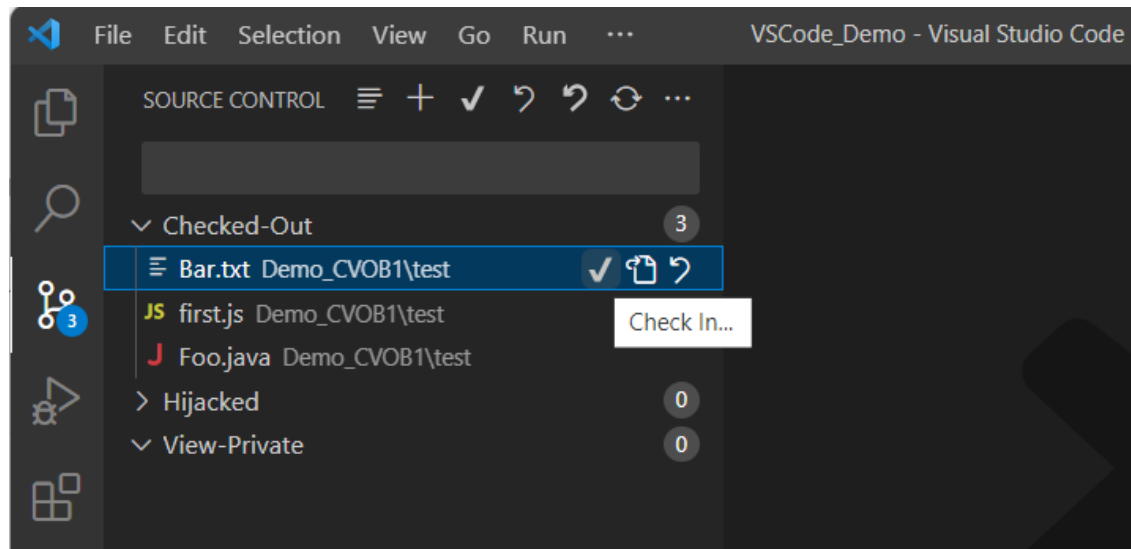
Figure 41 Check In single file

You can select the file/folder items under **Checked-Out** and right click on selected item, then click "**Check In…**" to check in the files/folders.
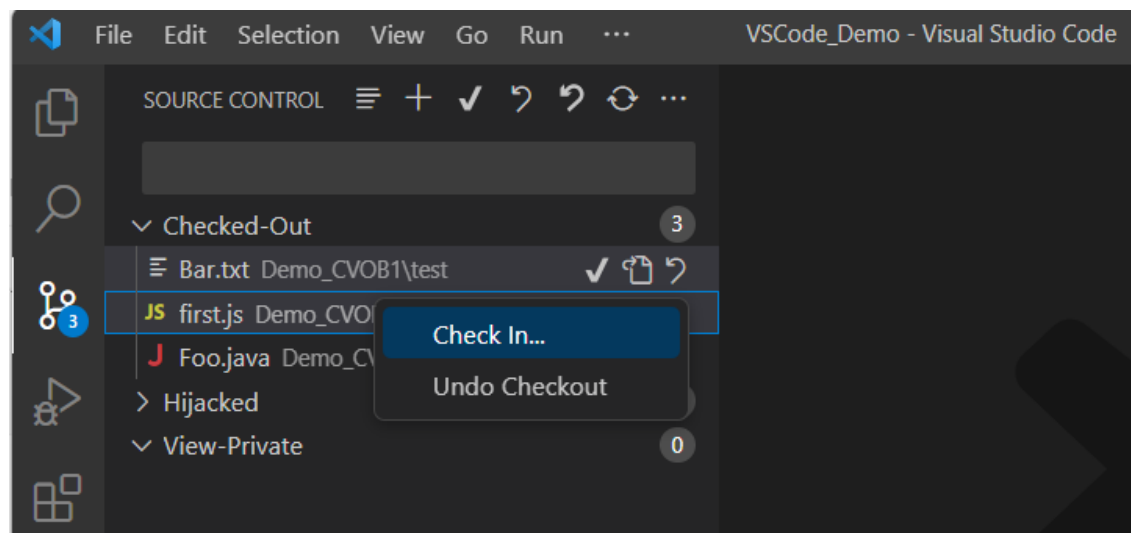
Figure 42 Check In selected items

To check in all checked out files/folders, click "✓" symbol in front of "**Checked-Out**" or "**SOURCE CONTROL**".
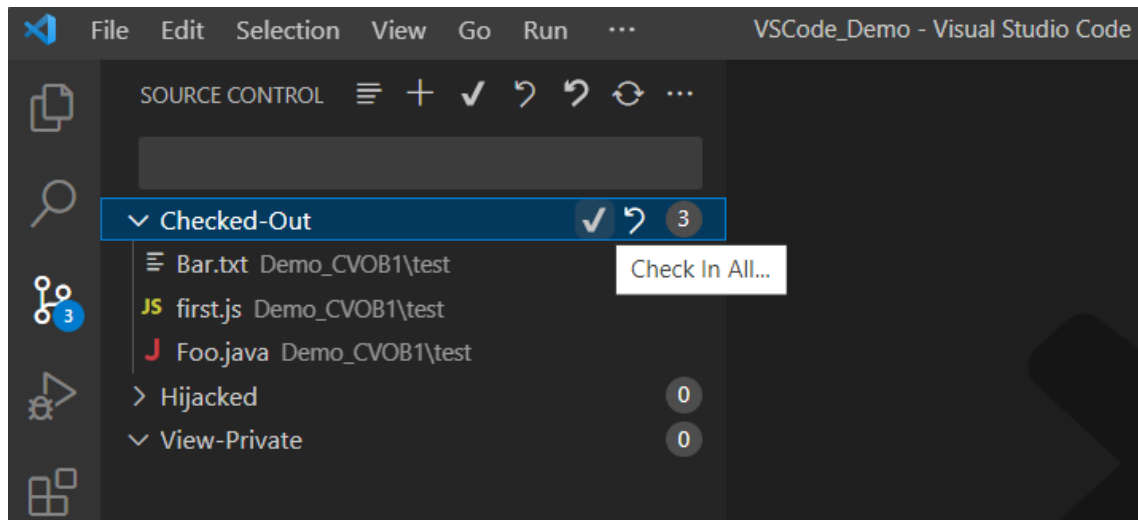
Figure 43 Check In all checked out files (click '✓' in front of Checked-Out)
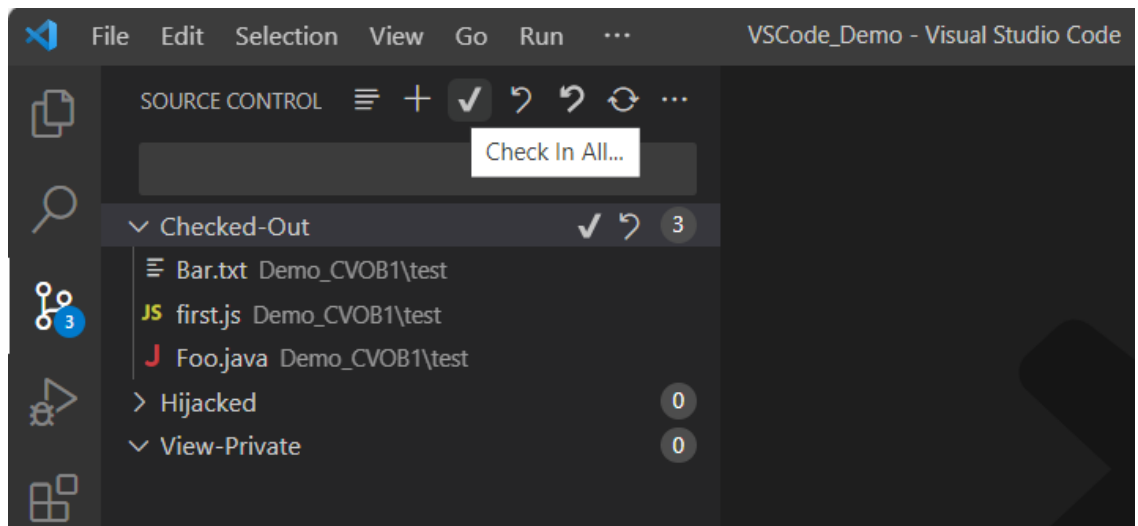
Figure 44 Check In all checked out files (click '✓' in front of SOURCE CONTROL)

Enter the comment and press enter to check in the resources.
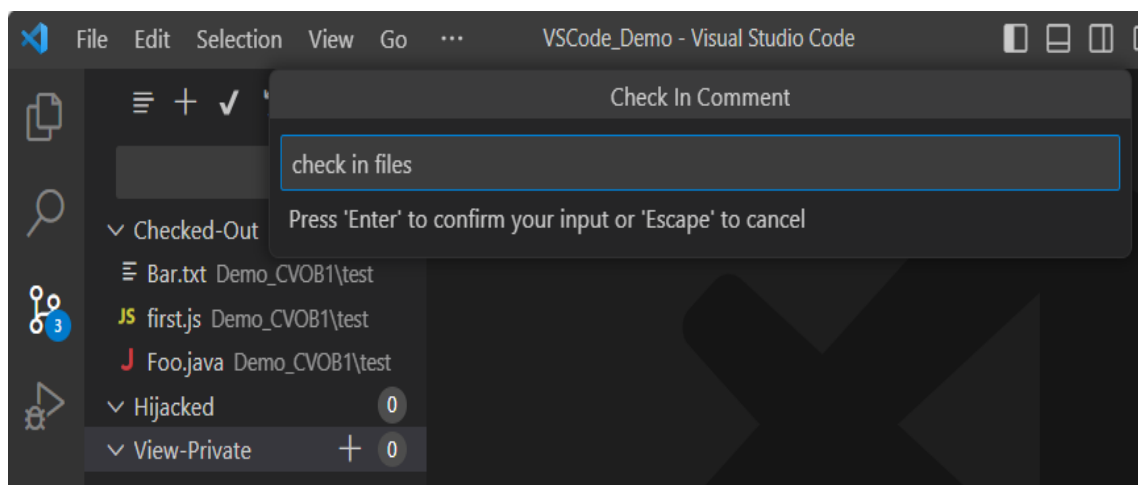
Figure 45 Check In files
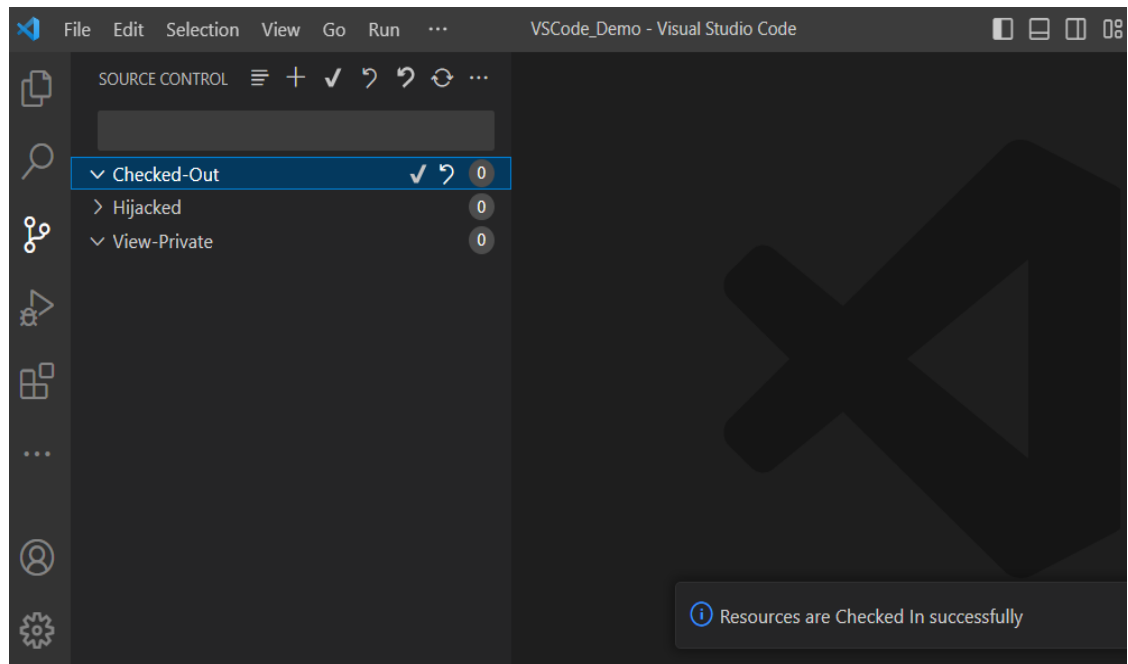
After checking in resources, the items will no longer appear in the **Checked-Out** section.

3.8 Undo Checkout

Reverts the checked out file/folder to its pre-checked out state. It retains a backup copy of the file with a suffix '**.keep**', if the '**Undo Checkout Keep**' option is enabled in the **VersionVault** Settings page (Refer **Error! Reference source not found.**). There are two places from where you can execute "**Undo Checkout…**" operation.

a. File Context Menu

Open the checked out file and right click in the file editor, select **VersionVault->Undo Checkout…** Using this option, you can undo checkout on a single file.
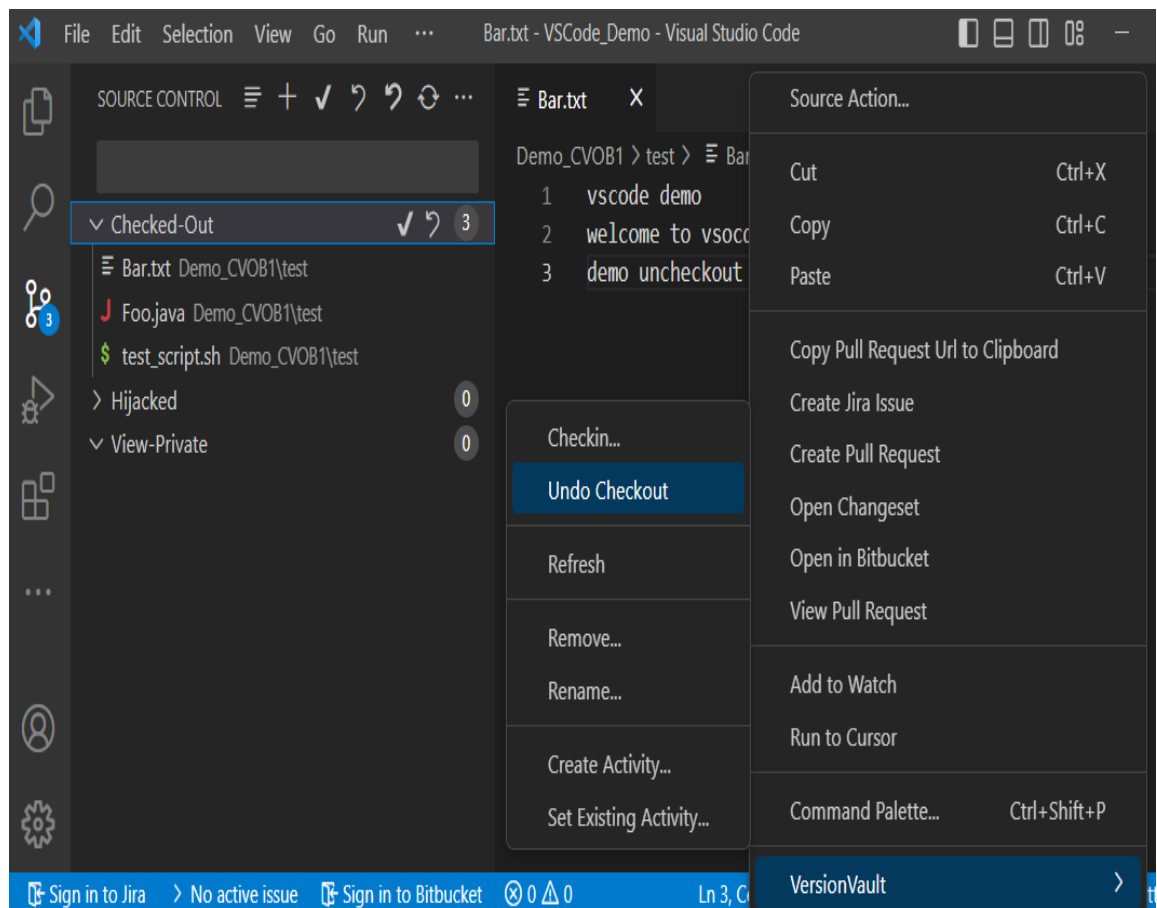
Figure 47 Undo Checkout single file
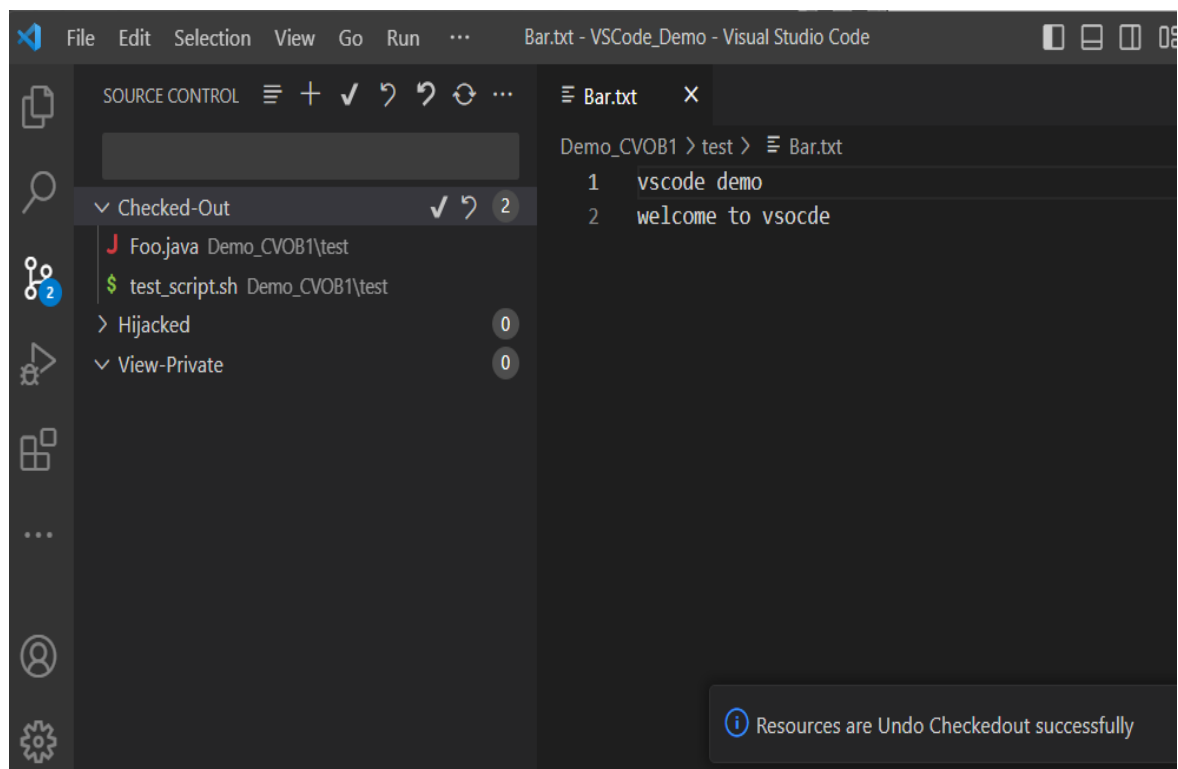
Figure 48 Resource is undo checked out

After undoing the checkout, the file will no longer appear in the **Checked-Out** section in the **Source Control** page and changes will be reverted.

b. Source Control Page (SCM)

Select the **Source Control** icon from the left panel, you will see list of checked out resources under "**Checked-Out**". From here you can undo checkout on a single, selected, or all checked out files/folders.
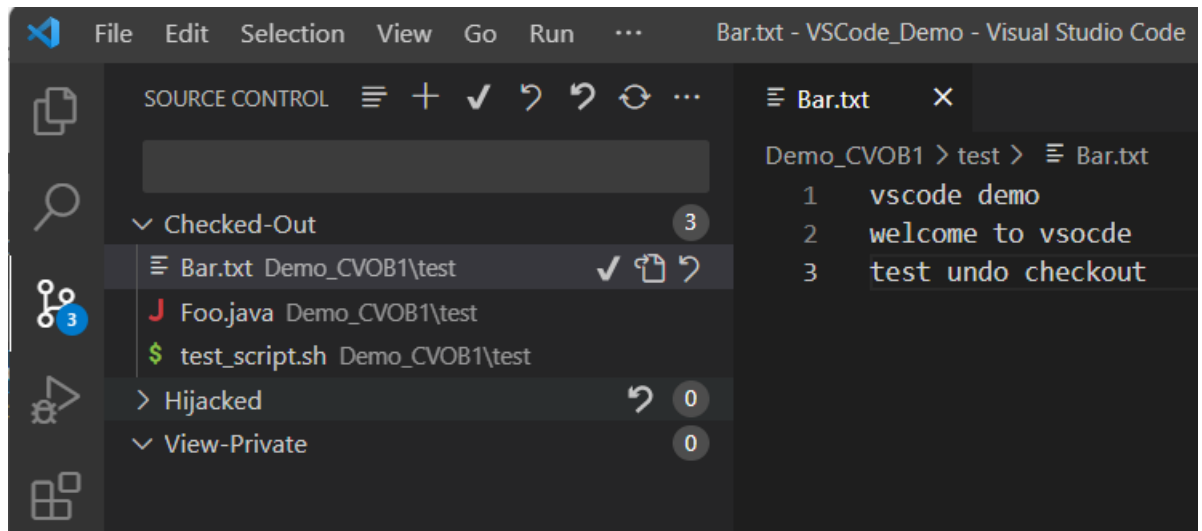

Figure 49 List of checked out files

You can click the 🔁 symbol in front of file/folder item under **Checked-Out** to revert the file/folder.
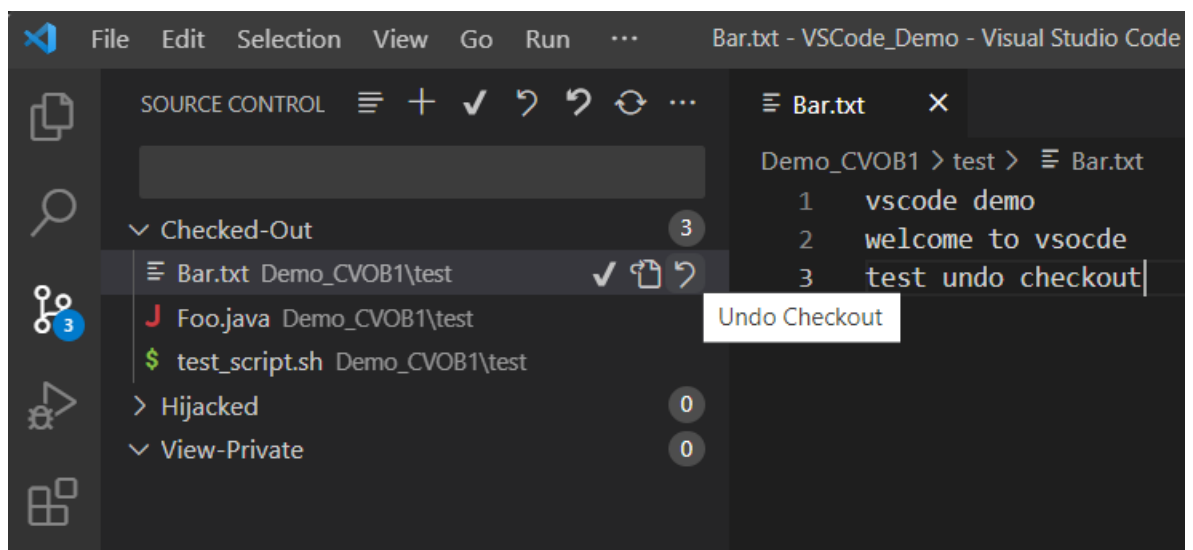
Figure 50 Undo Checkout single file from SCM

You can select the file/folder items under **Checked-Out** and right click on the selected item, then click "**Undo Checkout…**" to revert the changes.
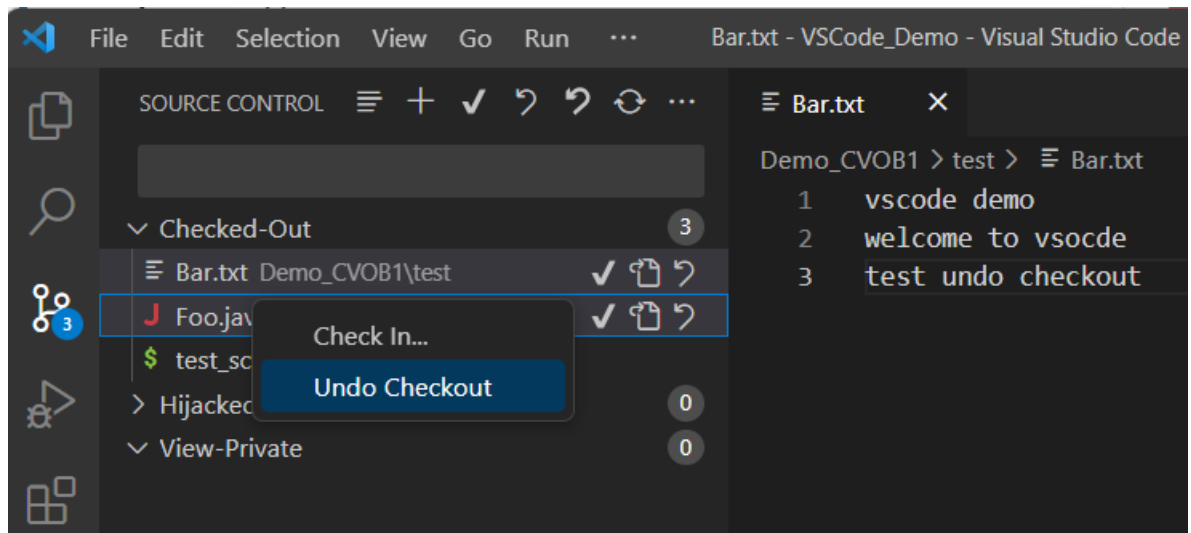
Figure 51 Undo Checkout selected files

To revert all the checked out files/folders, click the ⟲ symbol in front of "**Checked-Out**" or "**SOURCE CONTROL**".
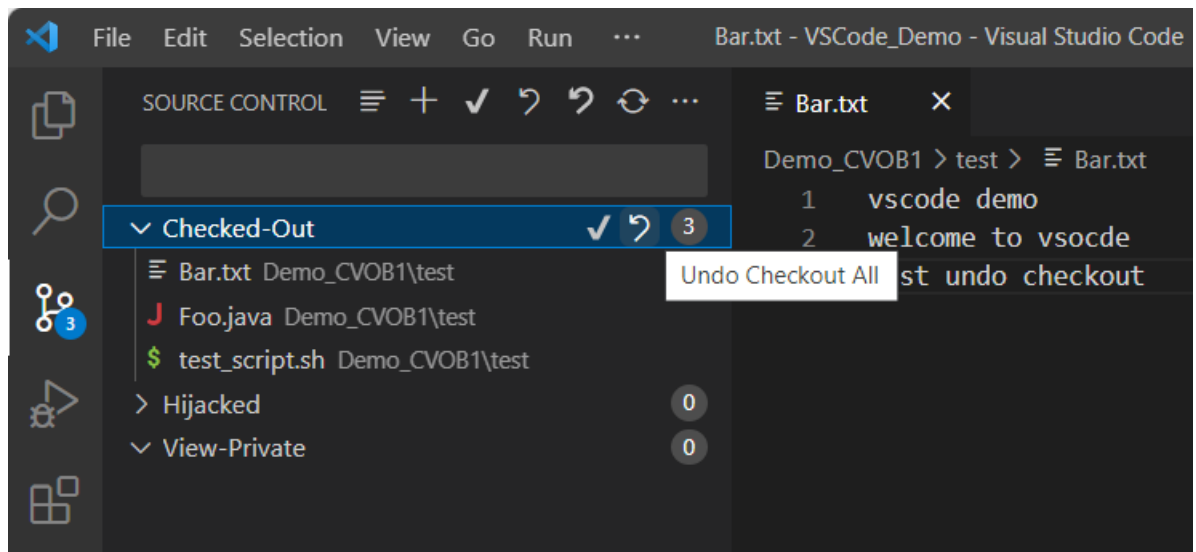

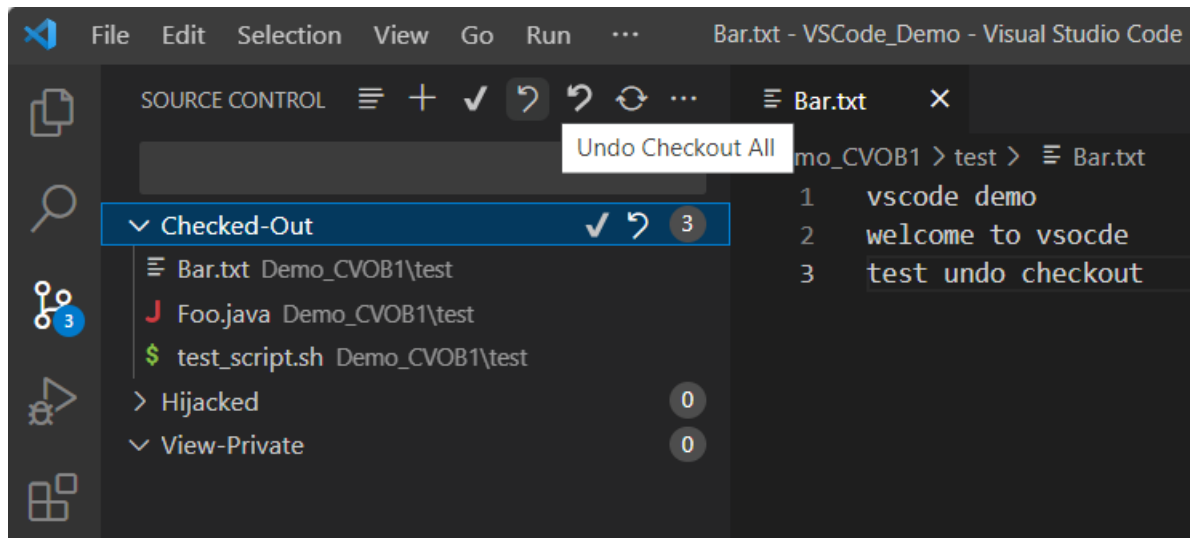*Figure 52 Undo Checkout all checkedout files*

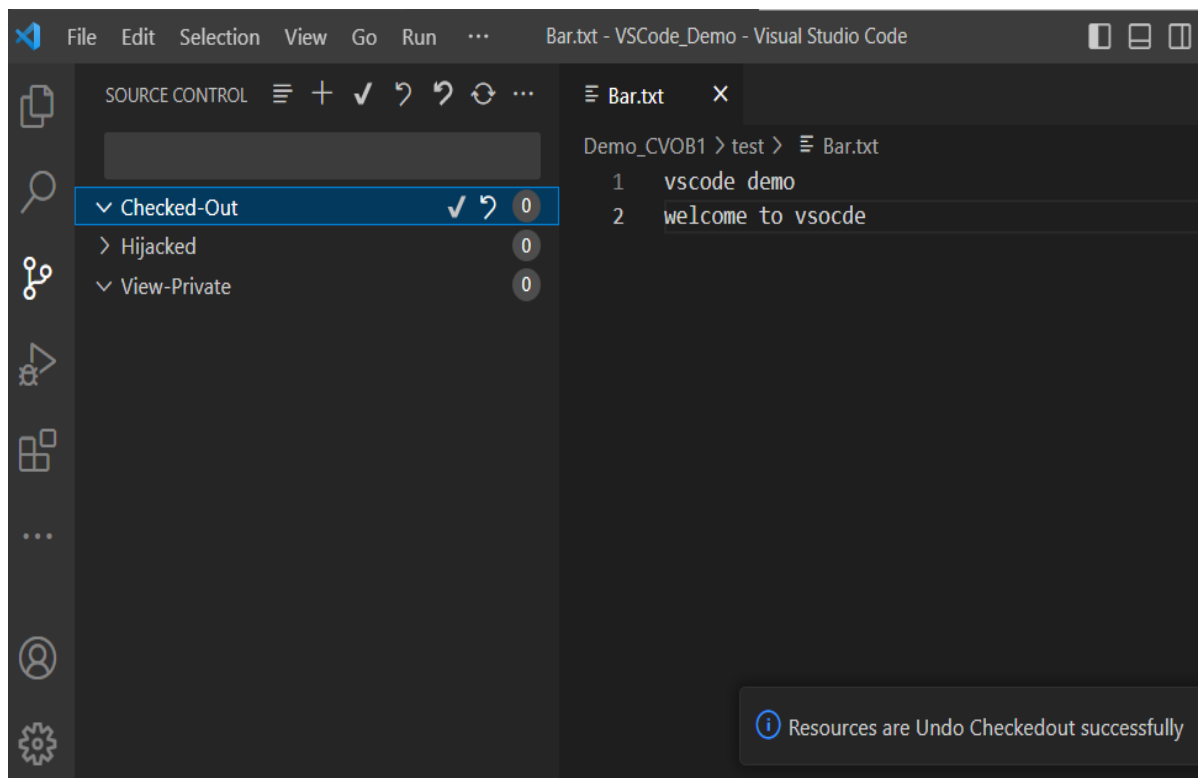Figure 53 Undo Checkout all checkedout files

Figure 54 Resources are undo checked out

Post undo checkout, the items will no longer appear in the **Checked-Out** section and will be reverted to the pre-checked out state.

3.9 Hijack

Loaded files may be hijacked to allow modification without checking them out.
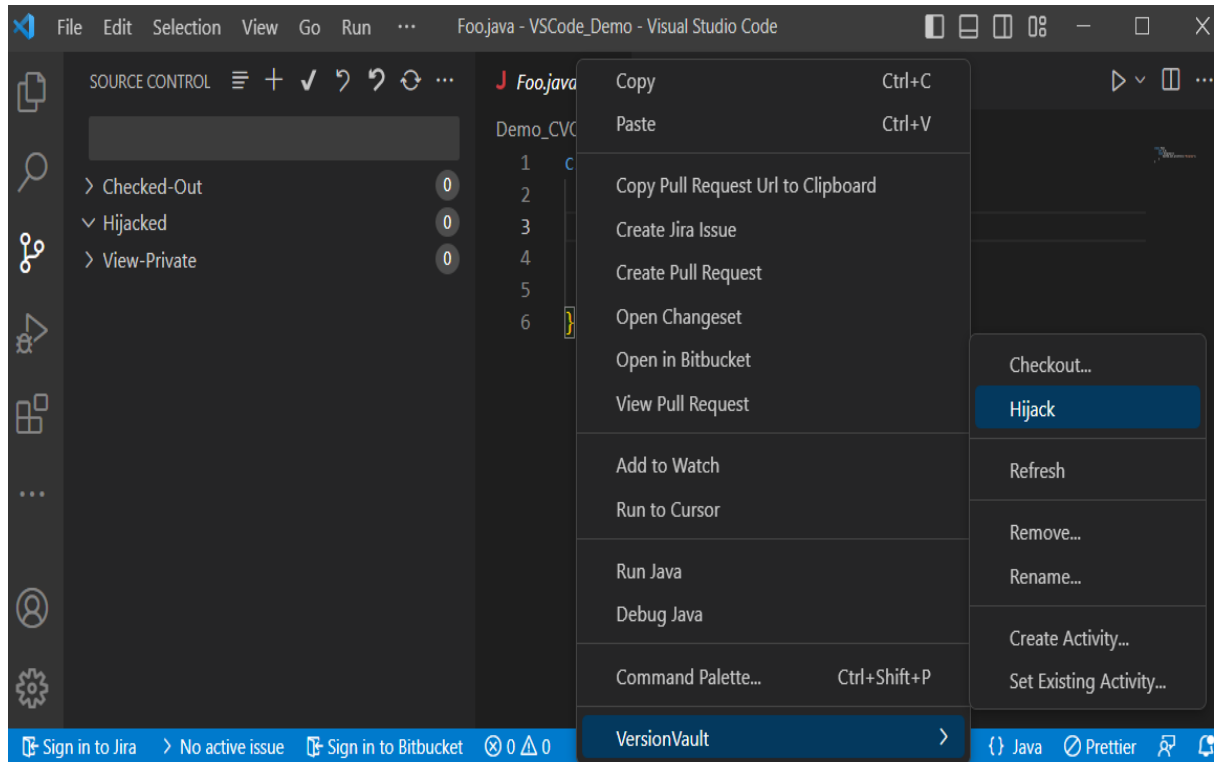Open the file and right click in file editor, select **VersionVault->Hijack…**

Post hijack, files will be listed under **Hijacked** section in **Source Control** page.
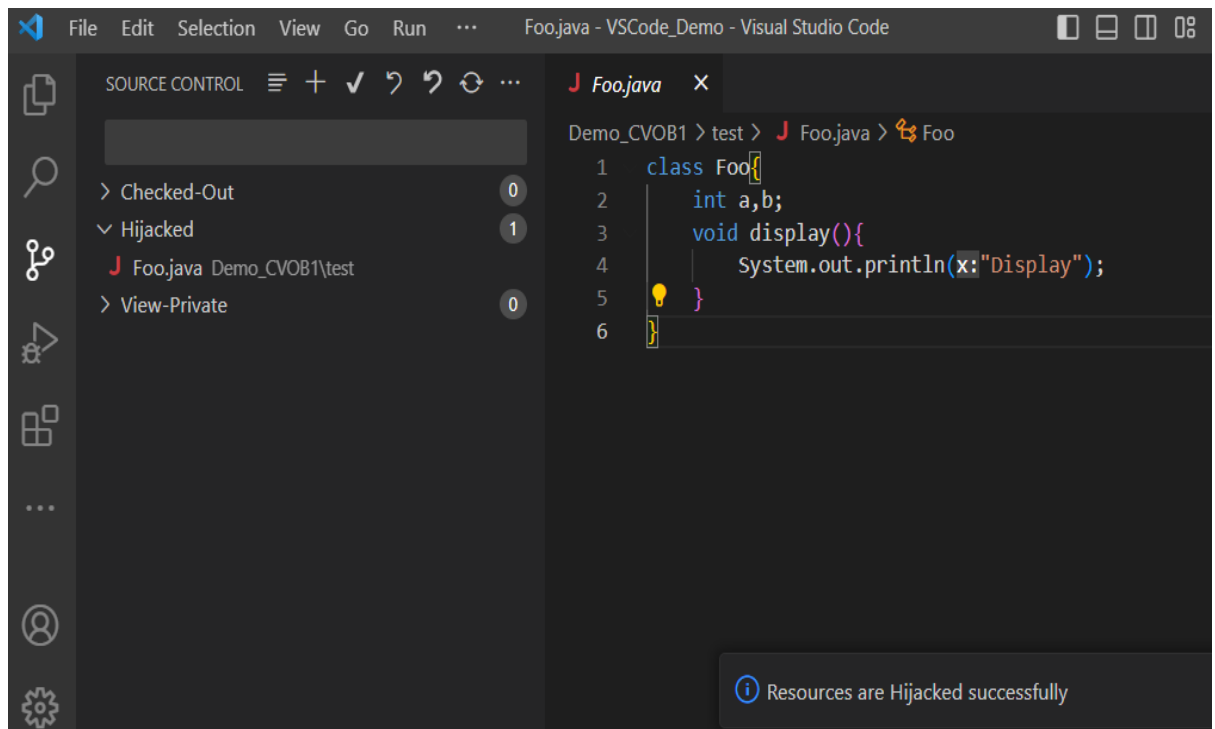
3.10 Undo Hijack

Reverts the hijacked file to its pre-hijacked state. Retains a backup copy of the file with a suffix '.keep' if the '**Undo Hijack Keep**' option is enabled in the

VersionVault Settings page (Refer **Error! Reference source not found.**). If you do not want to create a '**.keep**' file, then disable the option. There are two places from where you can execute "**Undo Hijack…**" operation.

a. **File Context Menu**

Open the hijacked file and right click in the file editor, select **VersionVault->Undo Hijack…** Using this option, you can undo hijack of a single file.
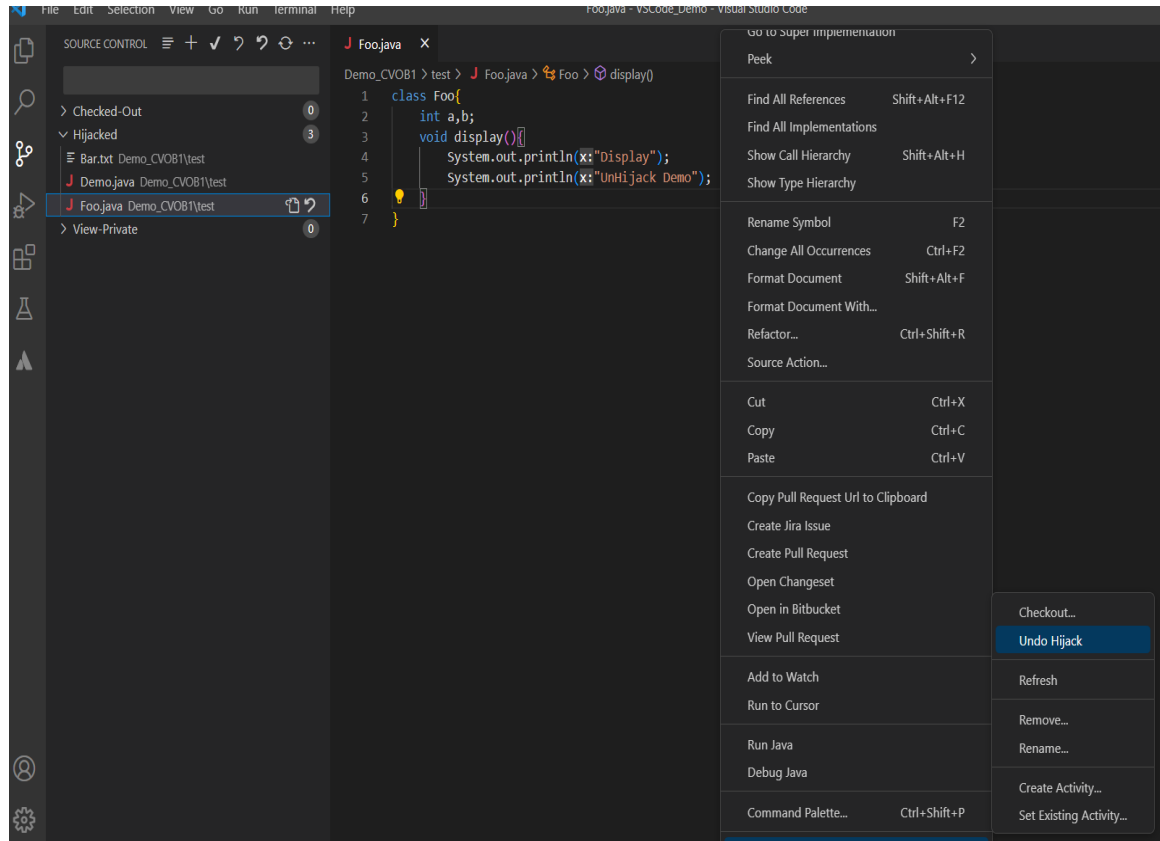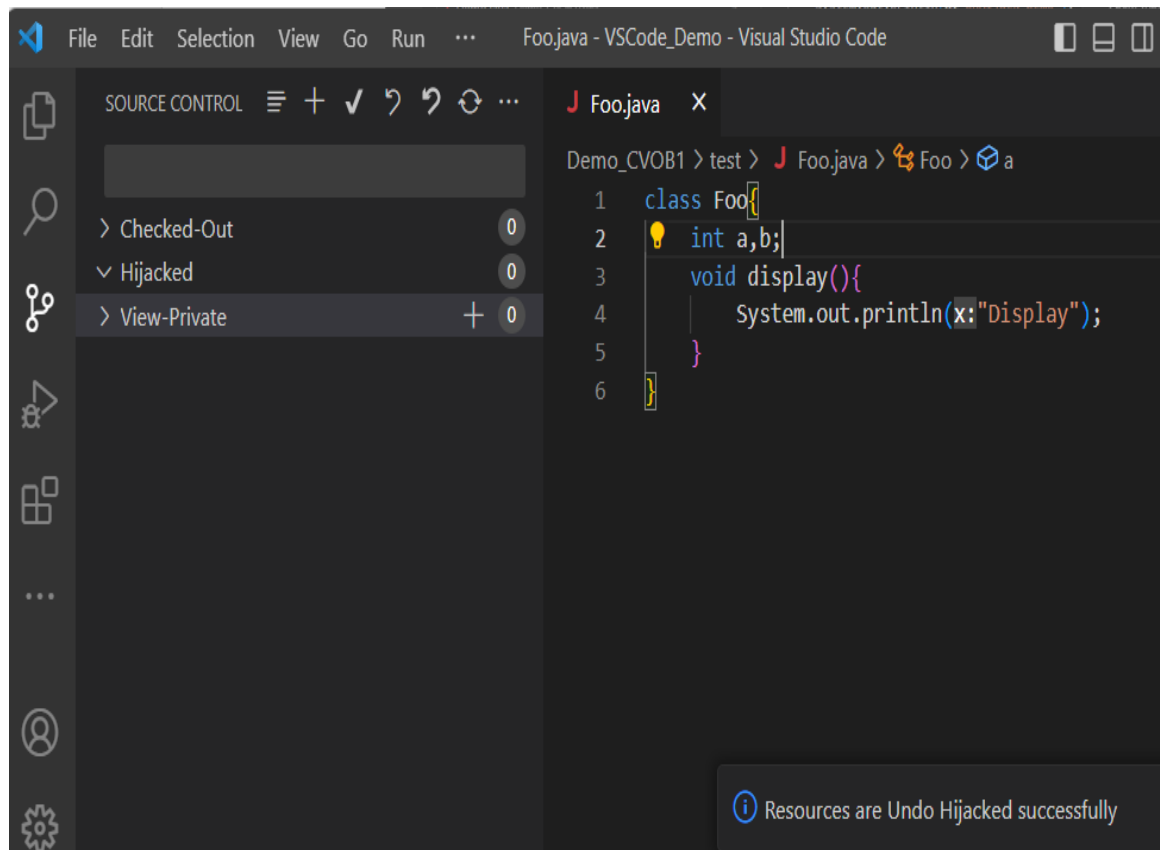

Figure 57 Undo Hijack file

Figure 58 Resource is undo hijacked

Post undo hijack, the file will no longer appear in the **Hijacked** section in **Source Control** page and changes will be reverted.

b. Source Control Page (SCM)
Select the **Source Control** icon from left panel to see the list of hijacked resources under "**Hijacked**". From here you can undo hijack of a single, selected, or all hijacked files.
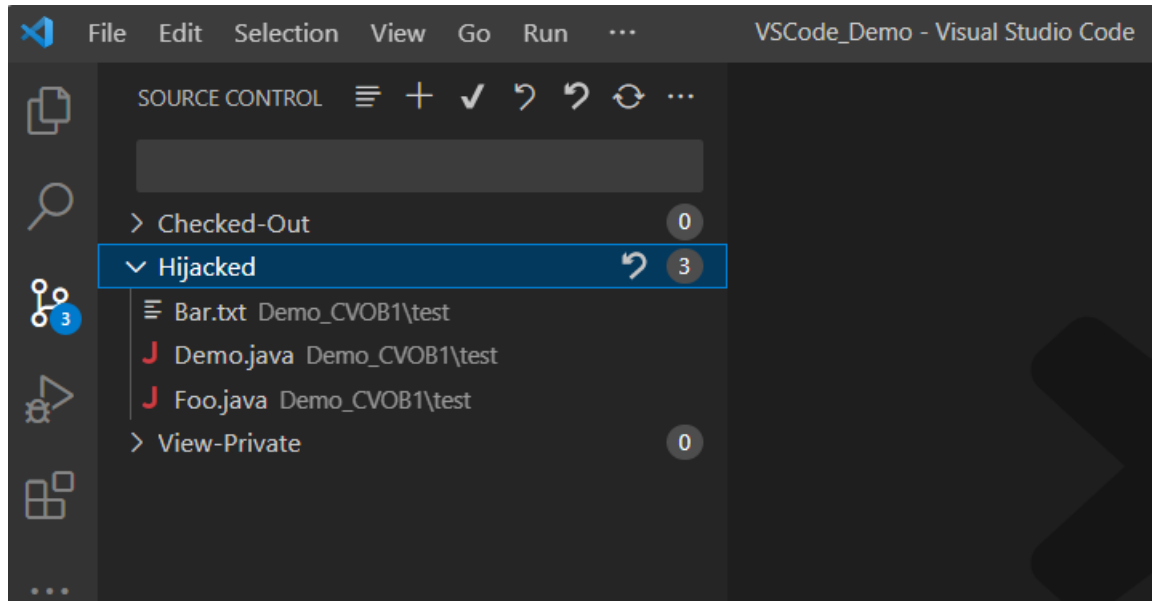
Figure 59 List of hijacked files

You can click the ⤺ symbol in front of file item under **Hijacked** to undo hijack on the file.
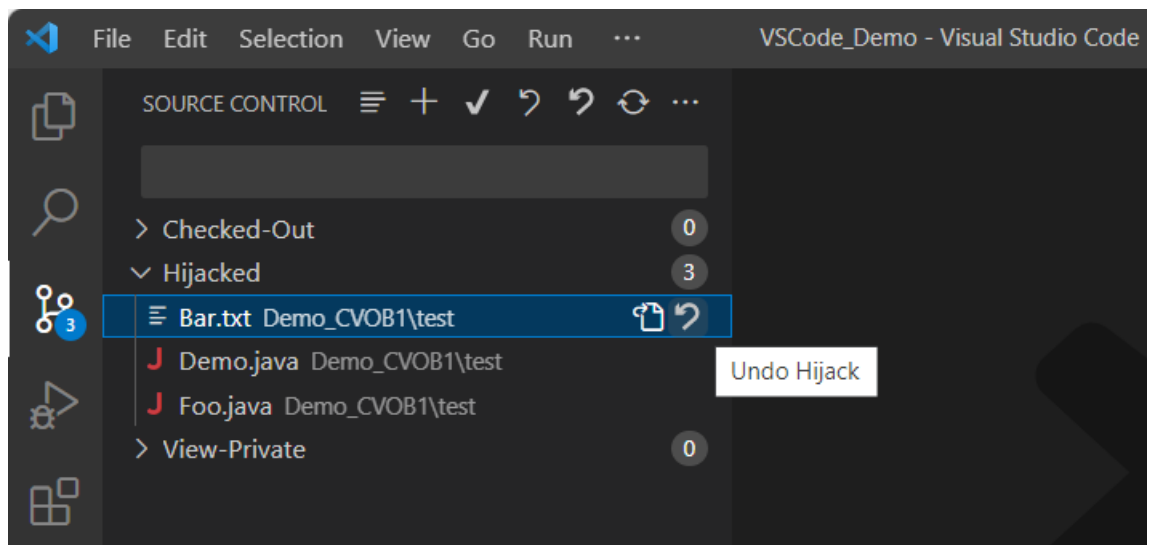
Figure 60 Undo Hijack single file

You can select the file items under **Hijacked** and right click on selected item, then click "**Undo Hijack…**" to revert the changes.
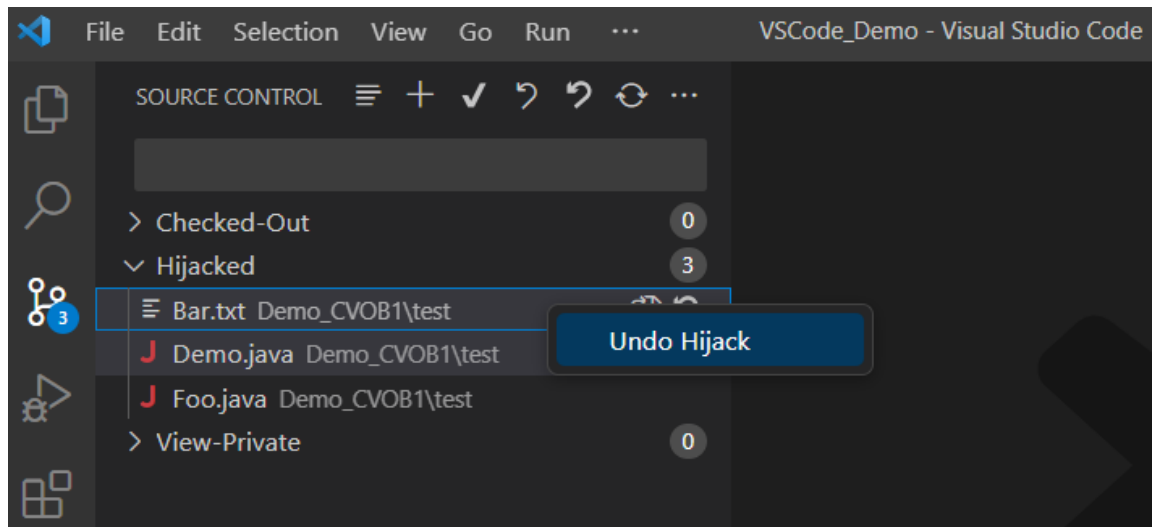
Figure 61 Undo Hijack selected files

To revert all the hijacked files, click the symbol in front of "**Hijacked**" or "**SOURCE CONTROL**".
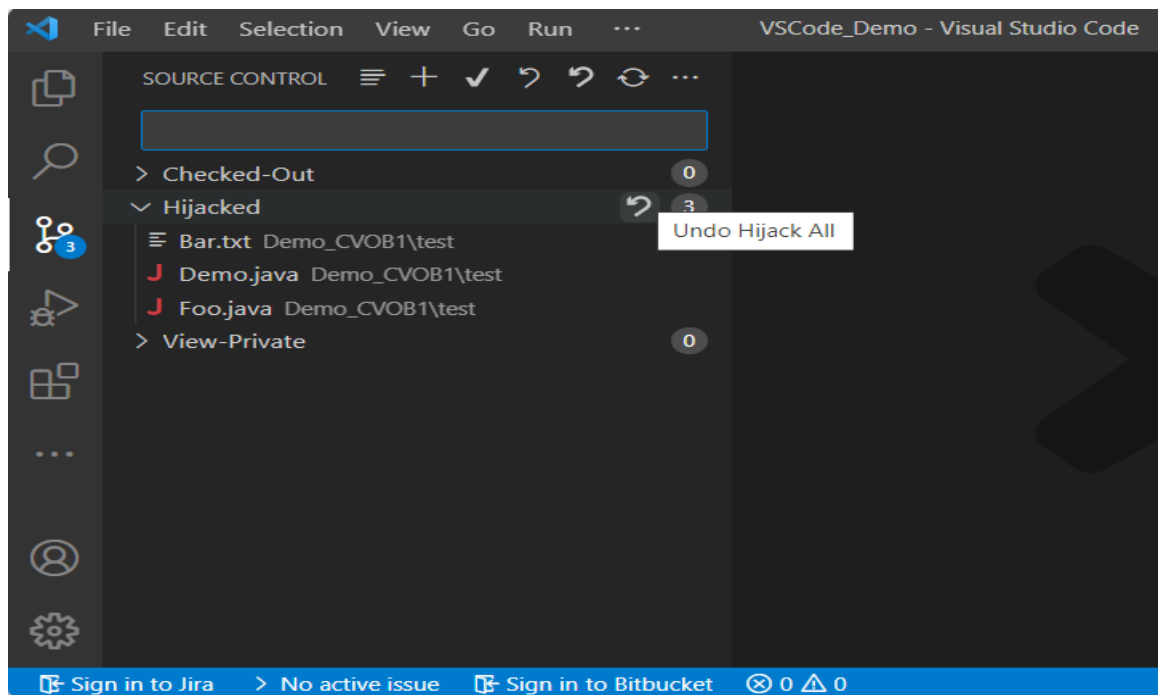

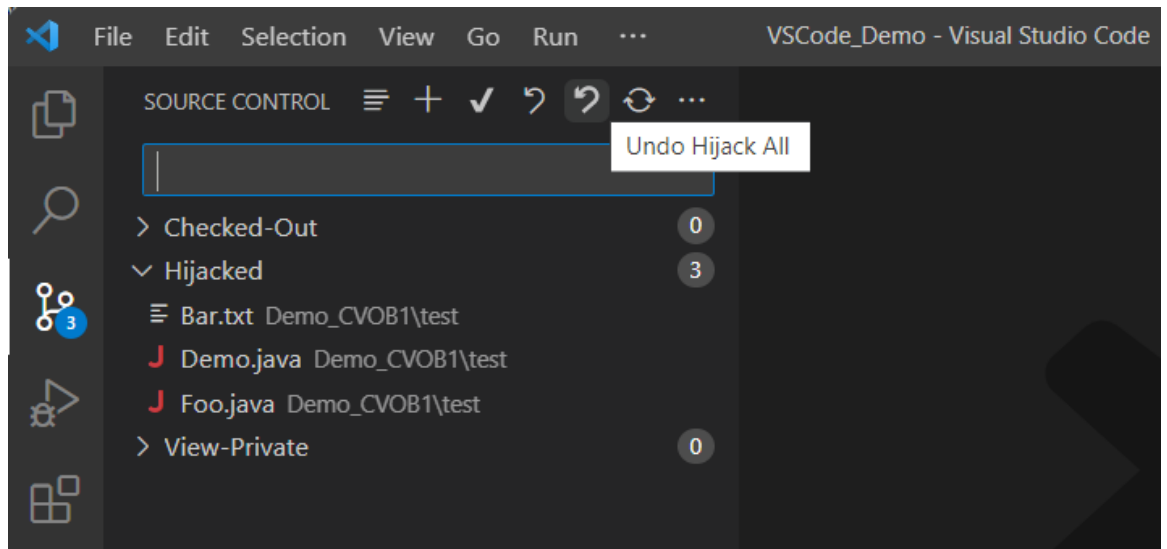*Figure 62 Undo Hijack all checked out files*
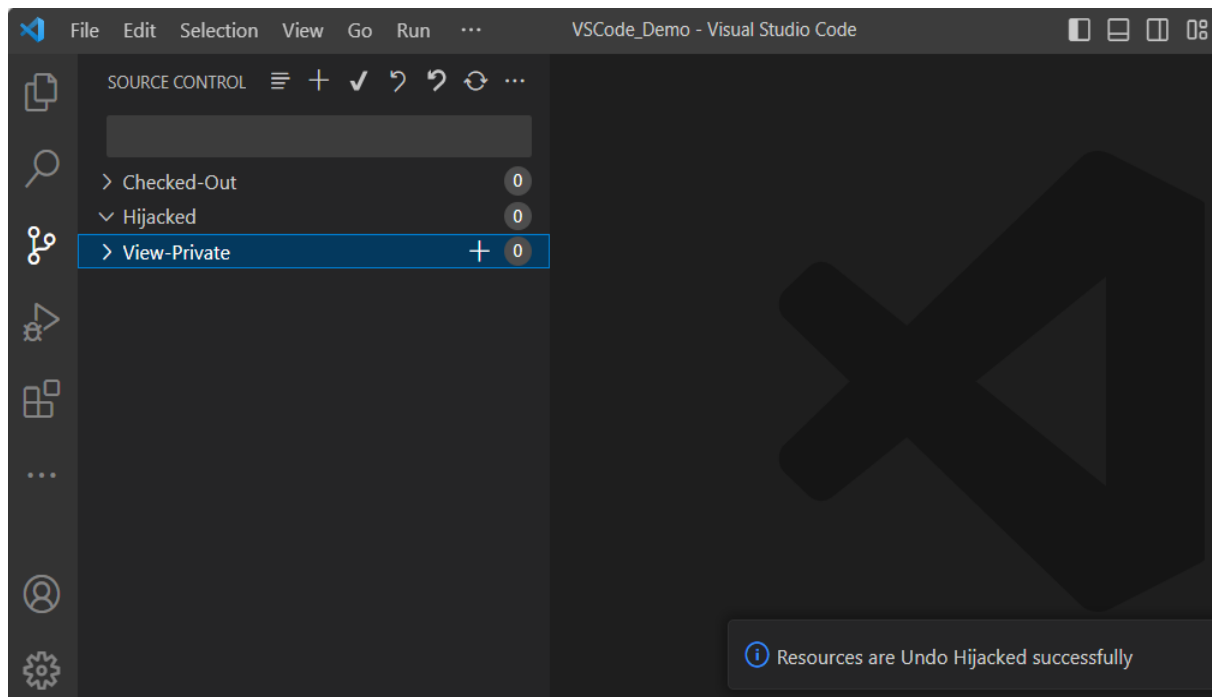
Figure 63 Undo Hijack all checked out files

Figure 64 Resources are undo hijacked

Post undo hijack, the items will no longer appear in the **Hijacked** section and the files will be reverted to their pre-hijacked state.

3.11 Rename

Rename the version-controlled resource to a new name in the same directory. Open the file and right click in the file editor, select the **VersionVault->Rename…** option which prompts you to enter a new name.
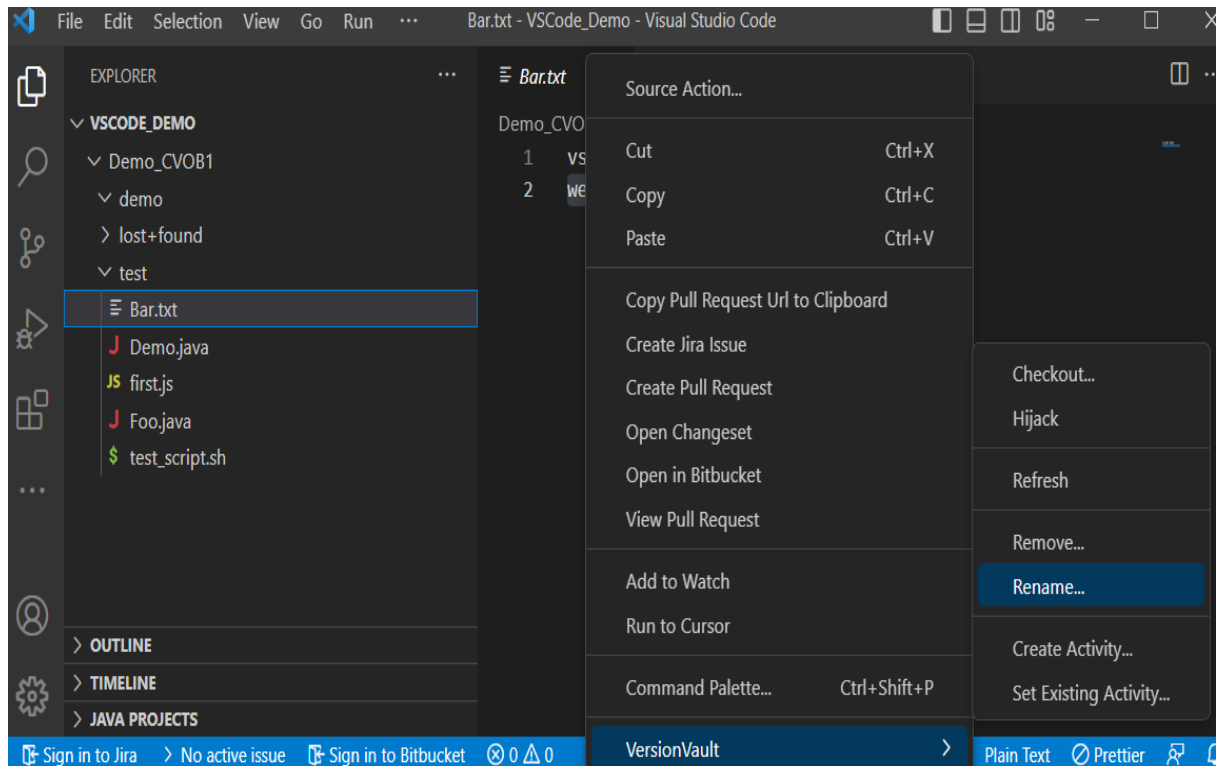
Figure 65 Rename resource

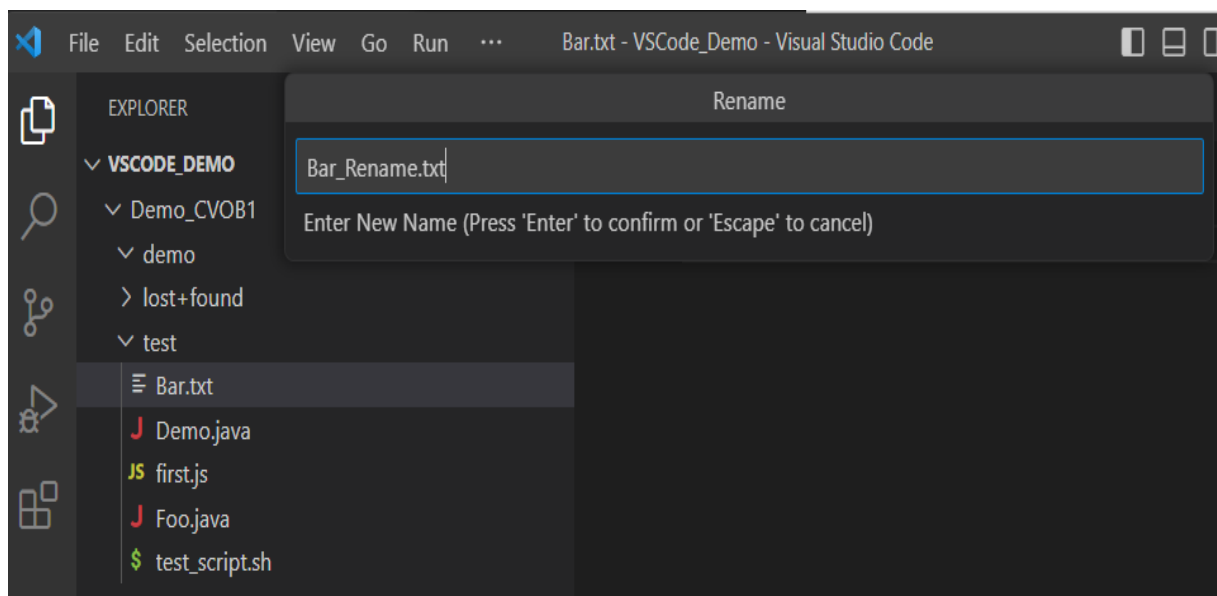Enter the new name and press enter to rename the file.
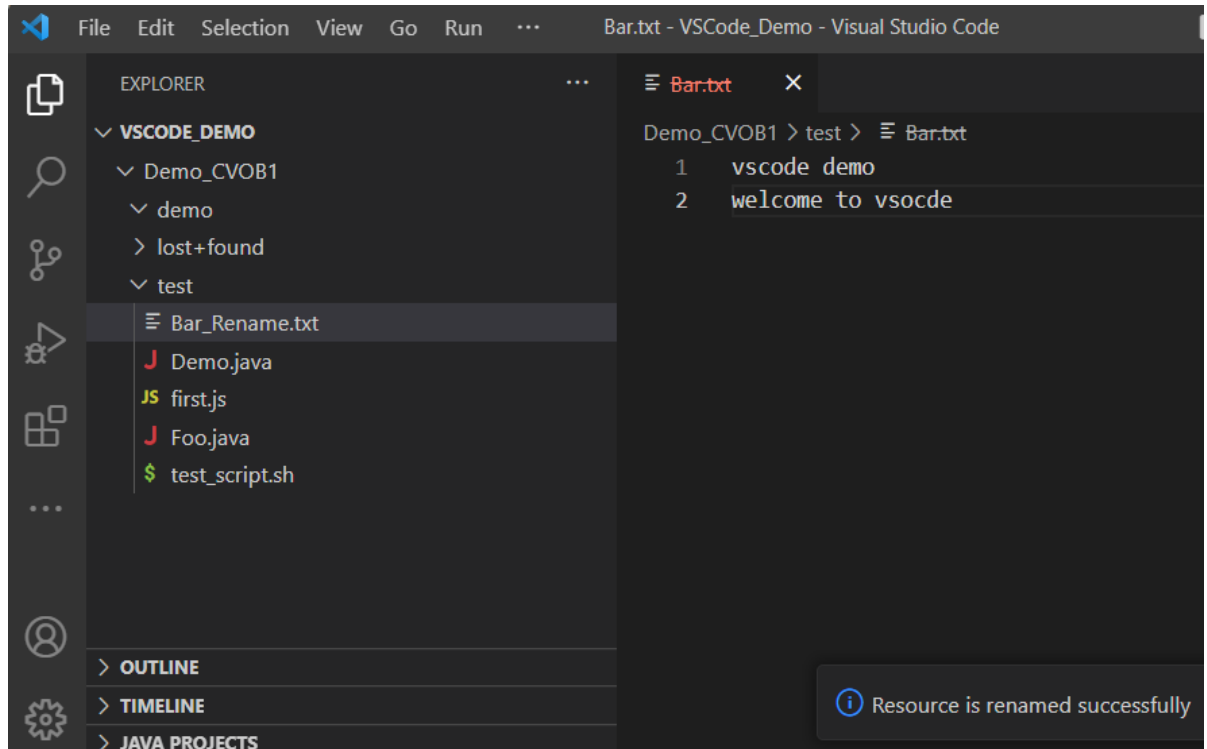

Figure 66 Rename Comment

Figure 67 Resource Renamed

3.12 Remove

Delete the source-controlled resource. It is like using the VersionVault `rmname` command. This command removes the resource from future versions of its parent directory, but it does not remove the element (i.e., the version history) from the VOB so it is still present in older versions or baselines. Its history will also remain accessible. Open the file and right click in the file editor, the select the **VersionVault->Remove…** option which prompts you for a comment.
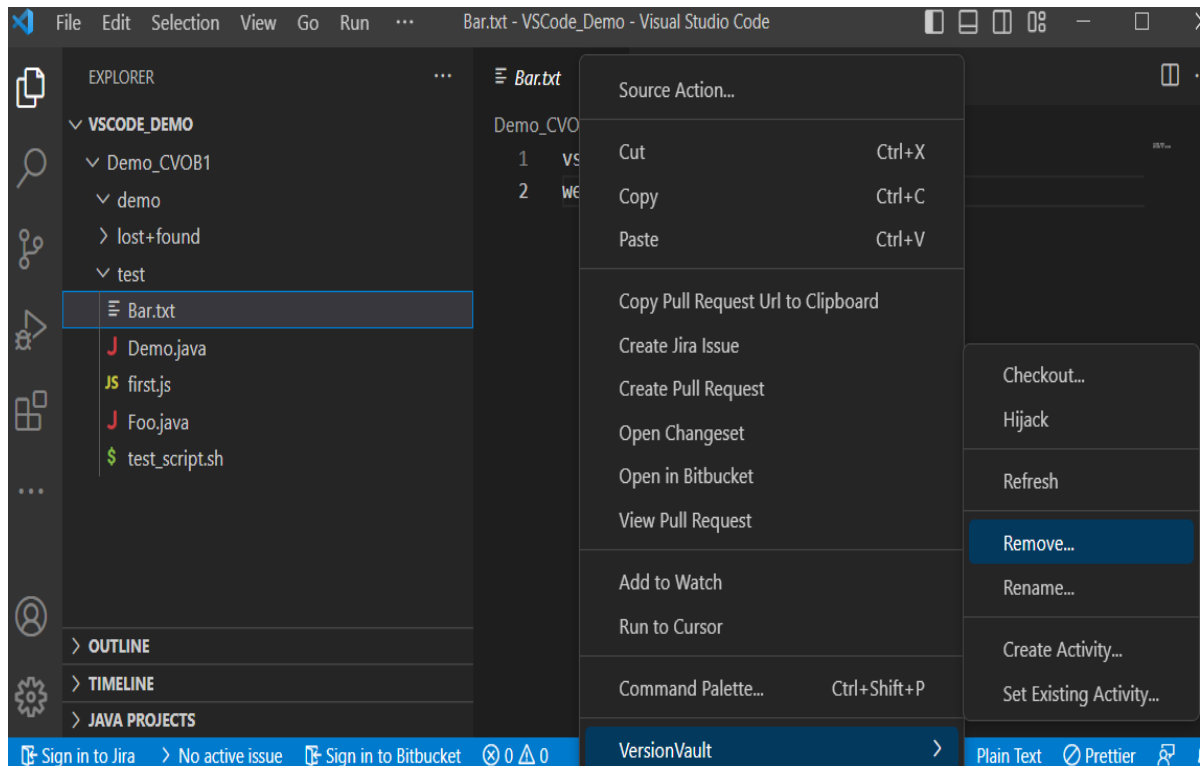
Figure 68 Remove resource

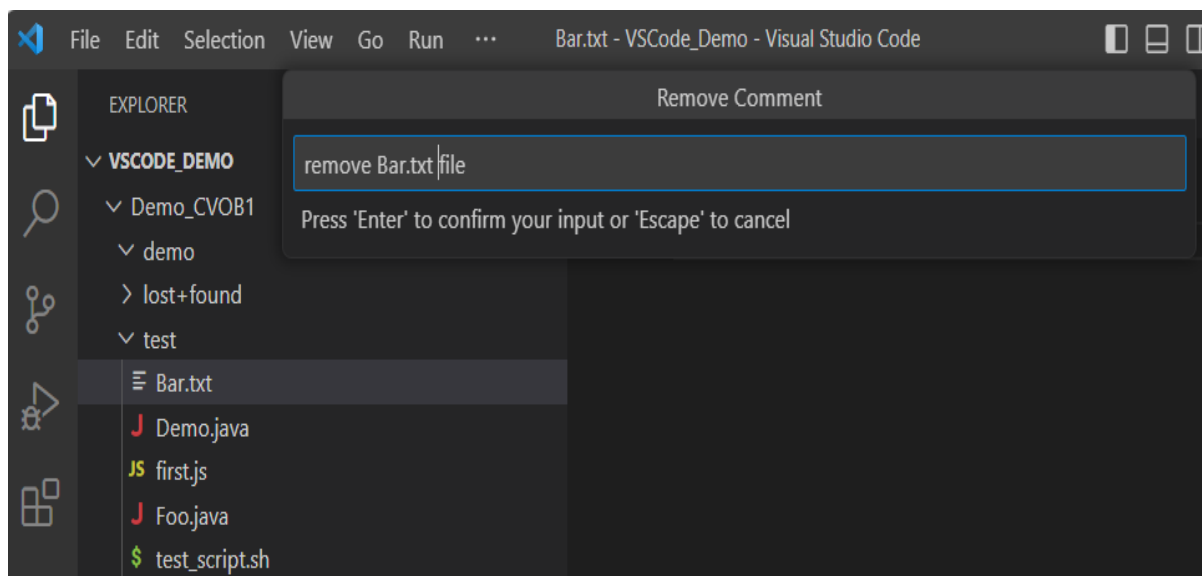Enter the comment and press enter to remove the file.
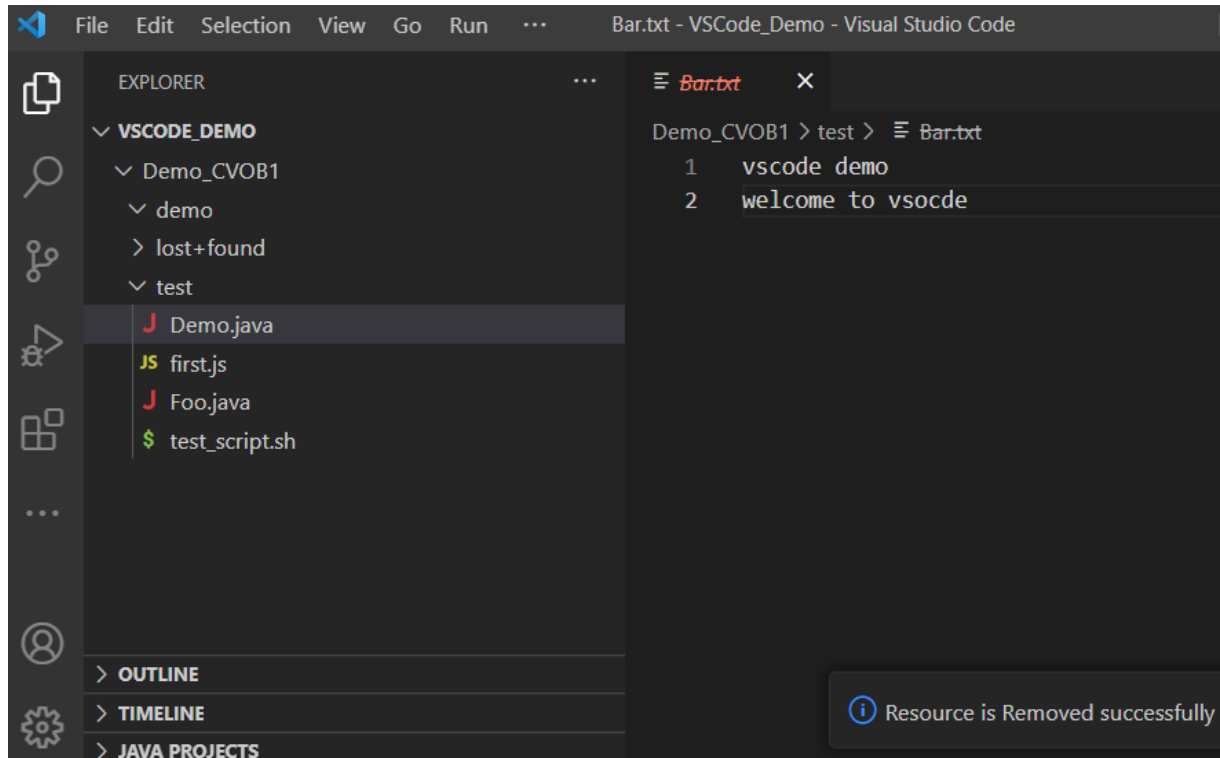

Figure 69 Remove Comment

Figure 70 Resource is removed

3.13 Refresh

Refresh the version-controlled resource. If the resource was changed outside Visual Studio Code which includes by any other VersionVault editor tools, **Refresh** will update Visual Studio Code with the changes. Open the file and right click in file editor, select **VersionVault->Refresh.**
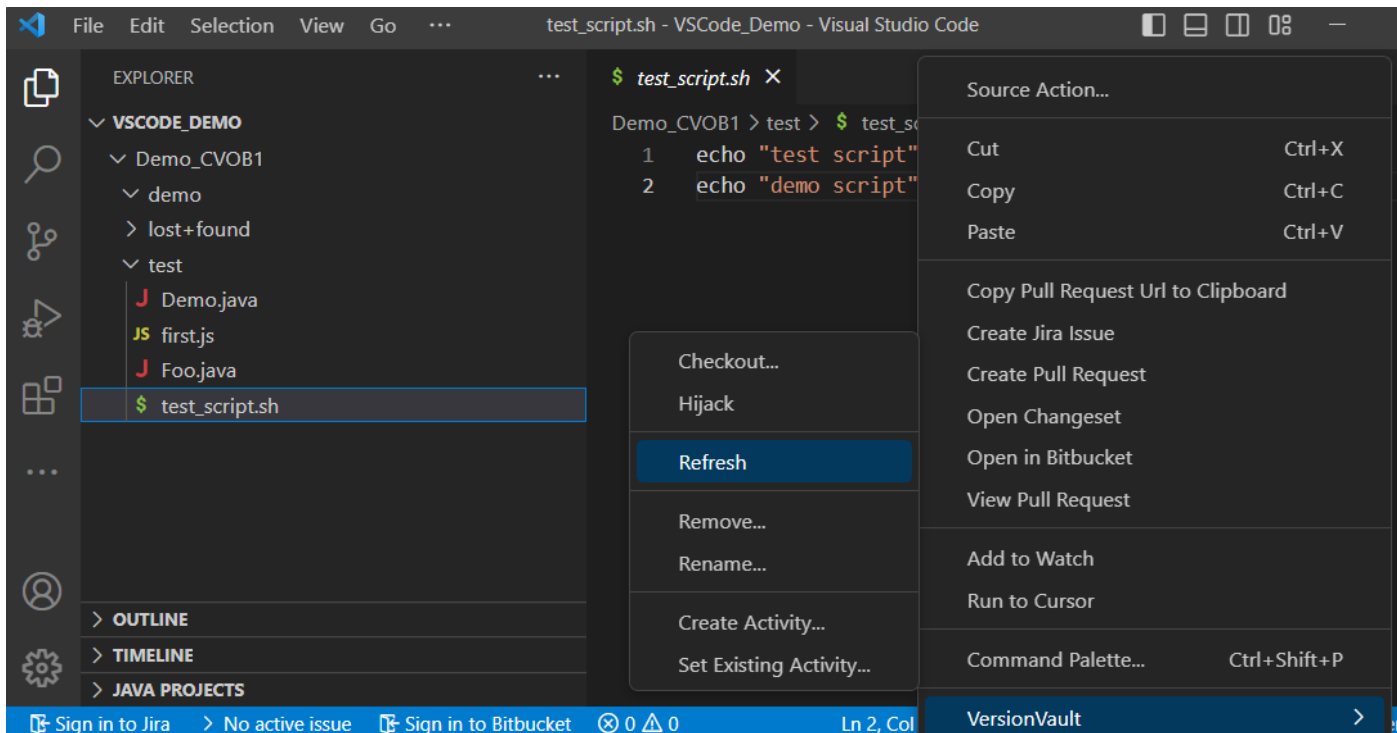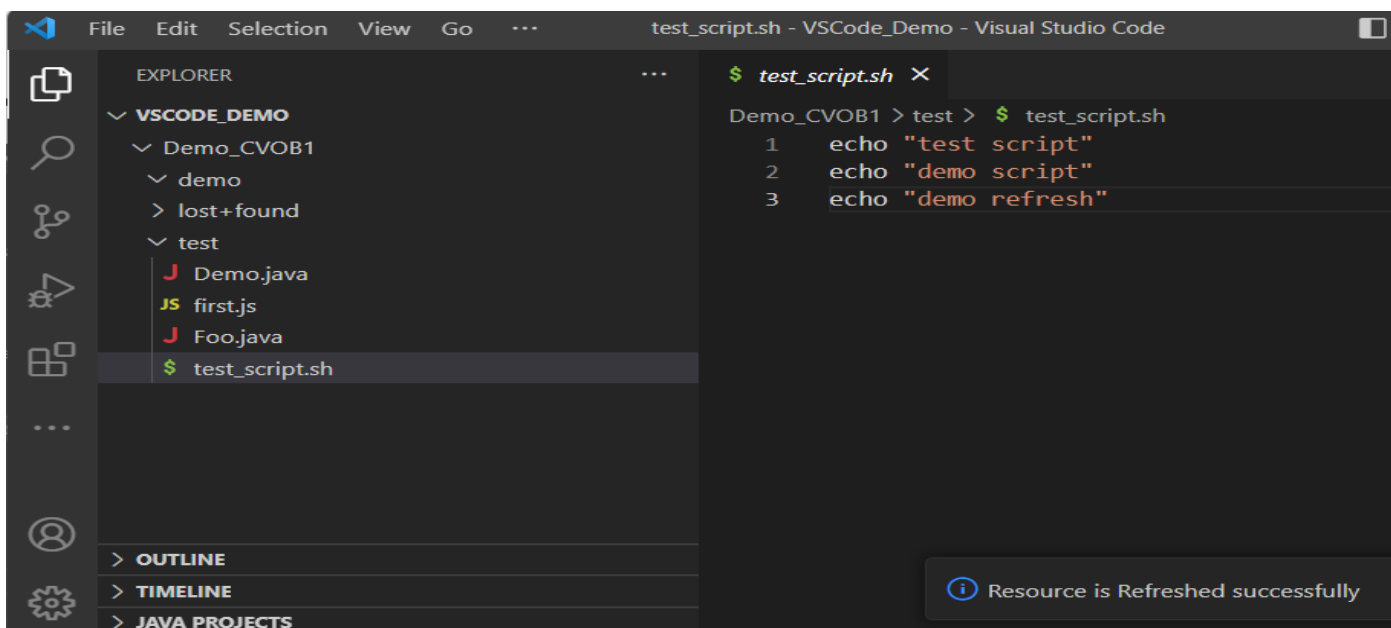
Figure 71 Refresh resource

Figure 72 Resource is refreshed

3.14 Update from Repository

If any changes were made to the VOB, perhaps by other users delivering their code, **Update from Repository** will update the resources to the latest versions and refresh the sections in SCM page like **Checked-Out**, **Hijacked** and **View-Private**
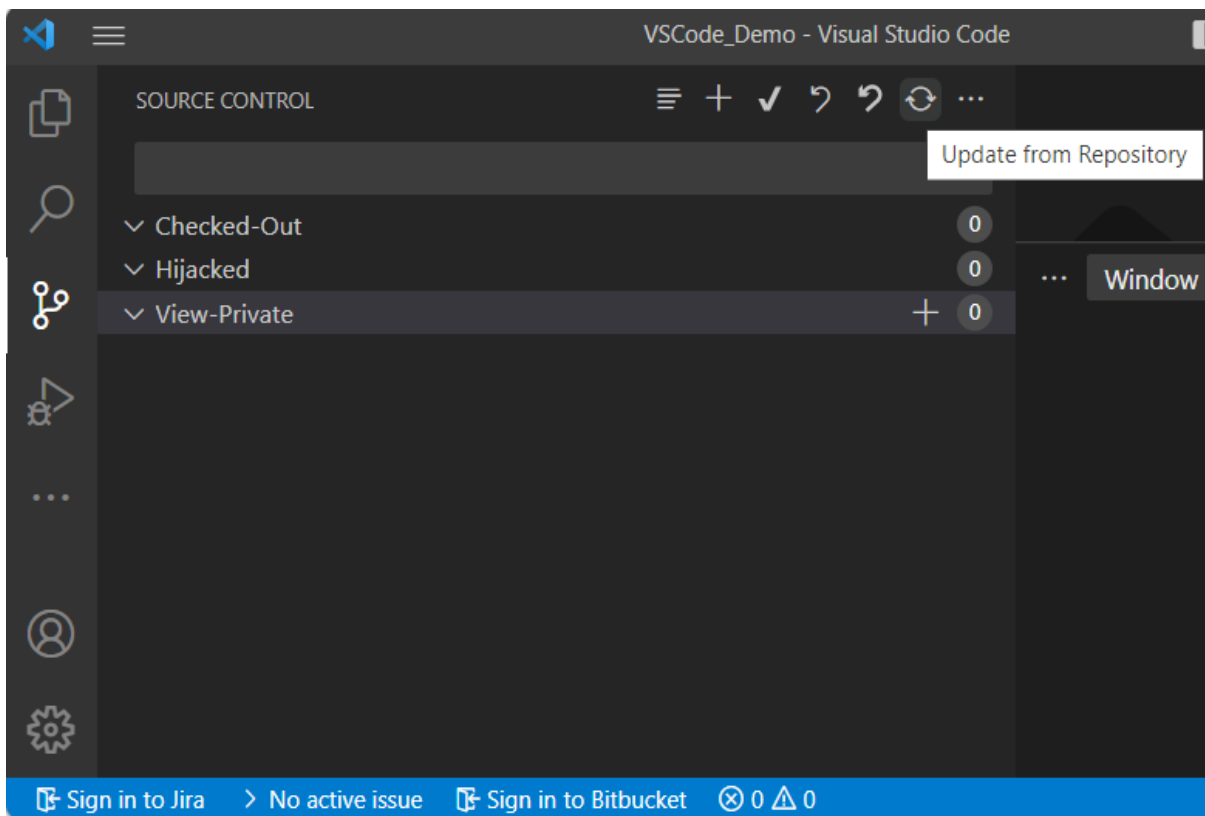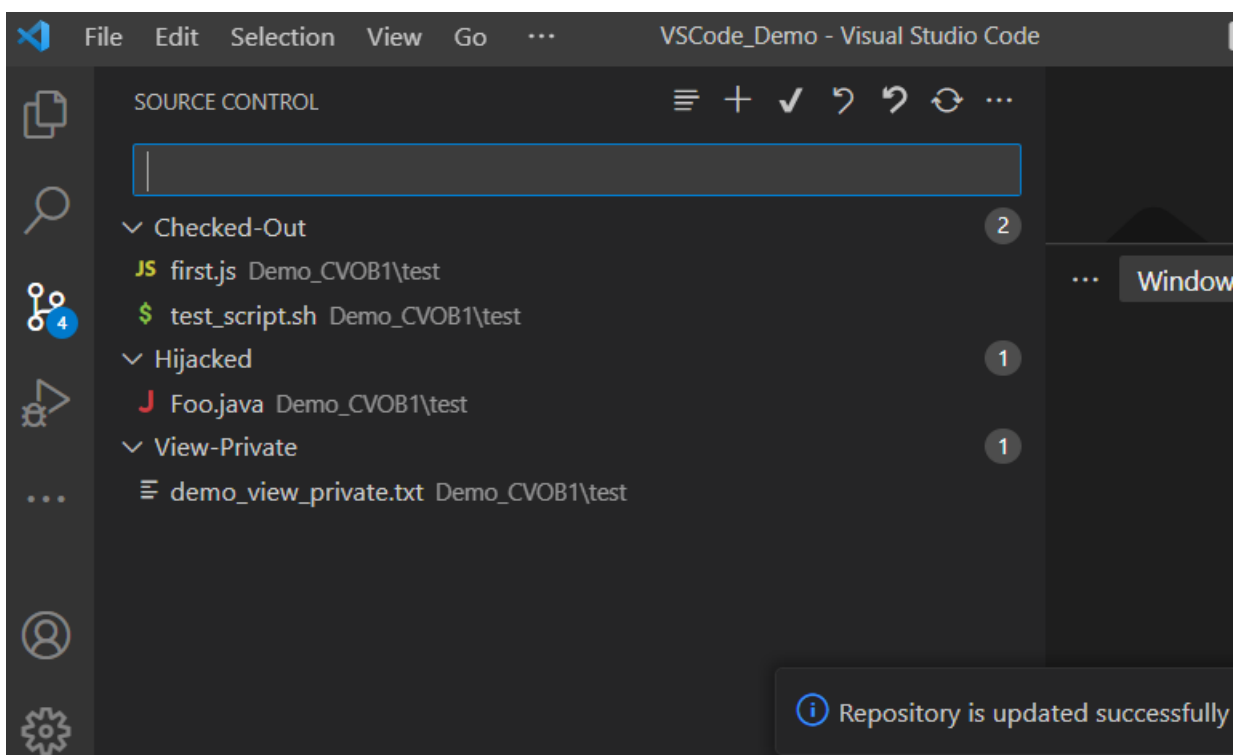
Figure 73 Update from Repository

Figure 74 Repository Updated

3.15 Disconnect from WAN Server

To log out, disconnect from the WAN server. Select the **Source Control** icon from the left panel, then click the "more" option **…** (three dots) and click the **VersionVault->Disconnect from WAN Server** option.
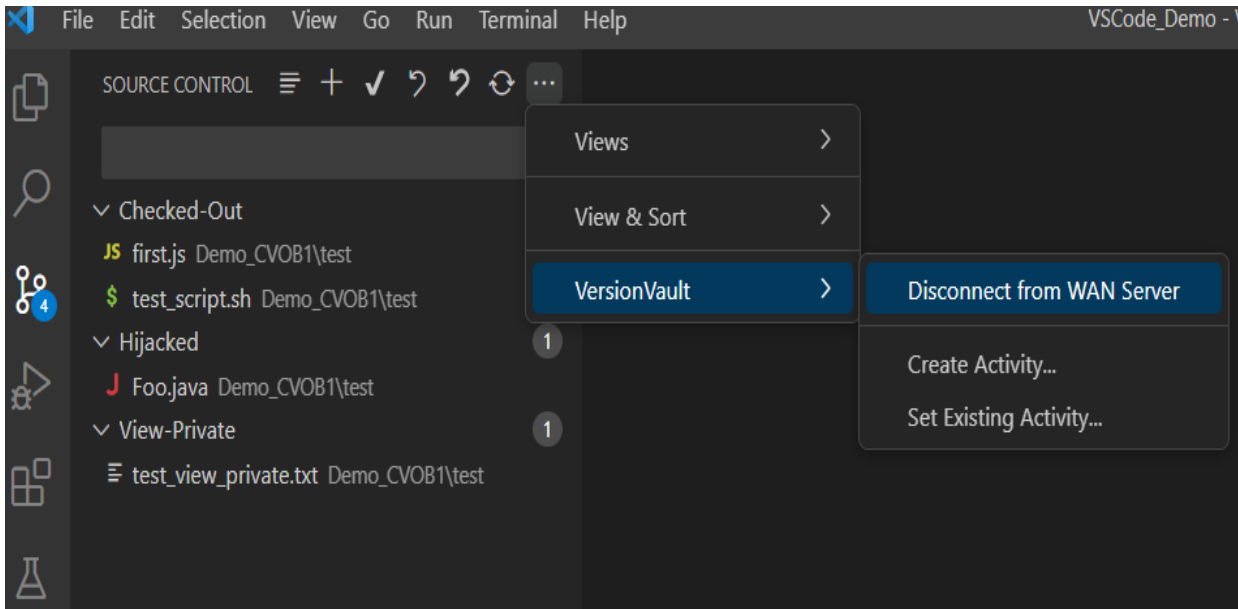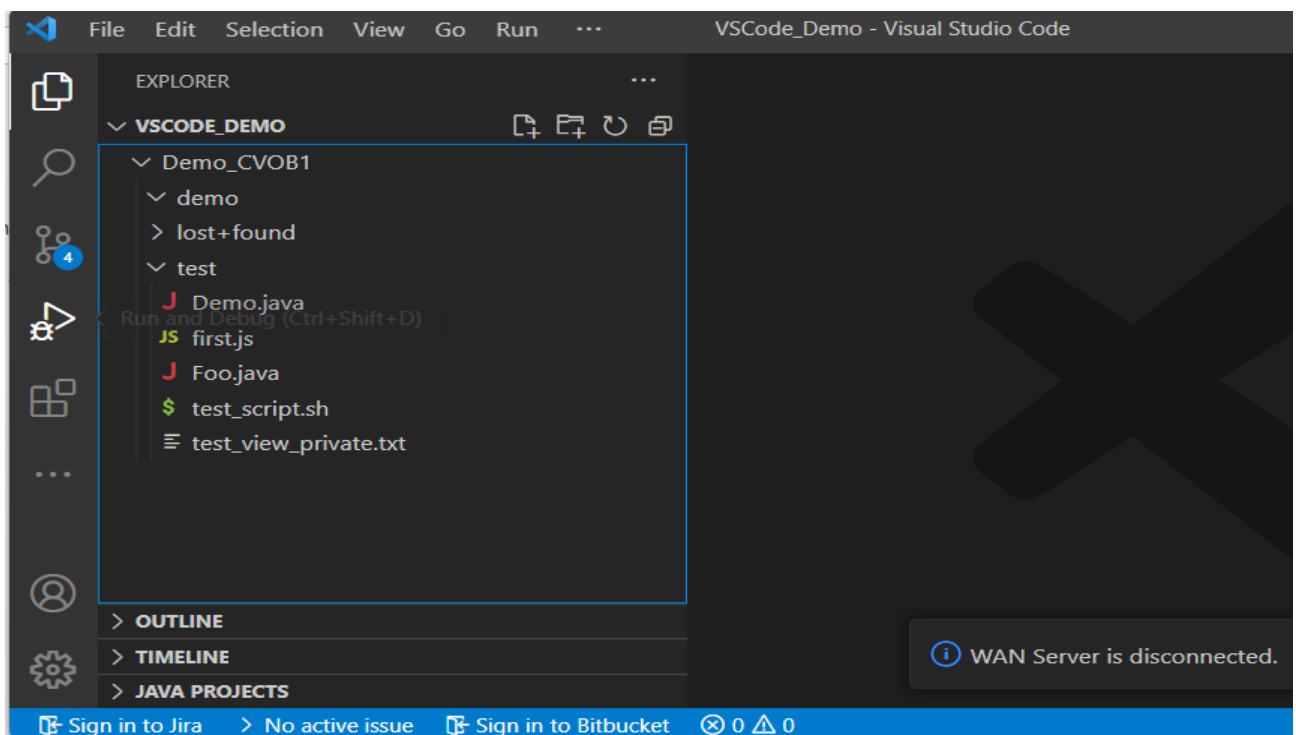

Figure 75 Disconnect from WAN server

Figure 76 WAN server is disconnected