

VERD STUDIO USER MANUAL

TABLE OF CONTENTS

TABLE OF CONTENTS	2
FILE TYPES LIST	3
TABLE LIST	3
SCREENSHOT LIST	3
1. VERD Studio.....	6
2. Files used by VERD Studio	8
2.1. Login File.....	8
2.2. Workspace File	9
2.3. Project File	10
3. Functions of VERD Studio.....	11
3.1. Create New Workspace	11
3.2. Add Existing Workspace	12
3.3. Select from previously added Workspaces	13
3.4. Import Existing Project.....	14
3.5. Create New VERD Project	16
3.6. Open a closed VERD Project.....	18
3.7. Close an opened VERD Project.....	19
3.8. Create New Folder under a Project.....	21
3.9. Create New File under a Project	23
3.10. Delete an existing VERD project.....	25
3.11. Remove an existing VERD project from workspace.....	27
3.12. Rename a VERD project.....	29
3.13. Paste a copied folder under a VERD project	31
3.14. Paste a copied file under a VERD project	33
3.15. Create New Folder under a Folder.....	35
3.16. Create New File under a Folder.....	37
3.17. Delete an existing folder under a VERD project	39
3.18. Rename a folder under a VERD project	41
3.19. Copy a folder under a VERD project.....	43
3.20. Paste a copied folder under a folder of VERD project	44
3.21. Paste a copied file under a folder of VERD project	46
3.22. Delete a file under a VERD project	48
3.23. Rename a file under a VERD project	50
3.24. Copy a file under a VERD project.....	52
3.25. Close an already opened file under a VERD project	53
3.26. Search and Replace in the current file.	55
3.27. Search and Replace in the Workspace.....	58
3.28. Open all projects in a Workspace	61

3.29. Close all projects in a Workspace	63
3.30. Format Selected Text	65
3.31. Build VERD Project	68
3.32. Build Workspace.....	70
3.33. Running VERD Project	72
3.34. Debug VERD Project.....	74
3.34.1. Add Breakpoint	76
3.34.2. Remove Breakpoint.....	78
3.34.3. Step Over to Next Instruction	80
3.34.4. Step Out from Current Function	82
3.34.5. Continue	84
3.34.6. Step into Function	86
3.34.7. Navigate through Call Stack	88
3.34.8. Send Content of Variable	90
3.34.9. Send Content of Variable that Mouse is over	92
3.34.10. Display a single row in a table	94
3.34.11. Display a type ref to field within structure.....	96

FILE TYPES LIST

File Type 2.1 Login File Example	8
File Type 2.2 Workspace File Example	9
File Type 2.3 Project File Example.....	10

TABLE LIST

Table 3.1 Options for Search and Replace for current file	56
Table 3.2 Options for Search and Replace for workspace	59

SCREENSHOT LIST

Screenshot 3.1 Initial screen of VERD Studio	11
Screenshot 3.2 Create Workspace Screen	11
Screenshot 3.3 Initial screen of VERD Studio	12
Screenshot 3.4 Add Workspace Screen	12
Screenshot 3.5 Initial screen with dropdown expanded	13
Screenshot 3.6 Main screen after loading workspace file	13
Screenshot 3.7 Menu Path for Open Project.....	14
Screenshot 3.8 Import Project File Dialog	15
Screenshot 3.9 Menu Path for New Project.....	16
Screenshot 3.10 New Project Create Screen.....	17
Screenshot 3.11 Menu Path for opening a closed project.....	18
Screenshot 3.12 Main screen after the project is opened.....	18
Screenshot 3.13 Menu Path for closing an opened project.....	19
Screenshot 3.14 Main screen after the project is closed.....	20
Screenshot 3.15 Menu Path for creating a folder under a project	21
Screenshot 3.16 New Folder Dialog	22
Screenshot 3.17 Menu Path for creating a file under a project.....	23

Screenshot 3.18 New File Dialog	24
Screenshot 3.19 Menu Path for deleting a project	25
Screenshot 3.20 Confirmation screen for deleting a project.....	26
Screenshot 3.21 Menu Path for removing a project	27
Screenshot 3.22 Confirmation screen for removing a project	28
Screenshot 3.23 Menu Path for renaming a project.....	29
Screenshot 3.24 Rename Dialog for project	30
Screenshot 3.25 Menu Path for pasting a folder under a project.....	31
Screenshot 3.26 Paste Dialog for pasting a folder under a project.....	32
Screenshot 3.27 Main screen after pasting a folder under a project	32
Screenshot 3.28 Menu Path for pasting a file under a project	33
Screenshot 3.29 Paste Dialog for pasting a file under a project	34
Screenshot 3.30 Main screen after pasting a file under a project	34
Screenshot 3.31 Menu Path for creating a folder under a folder	35
Screenshot 3.32 New Folder Dialog	36
Screenshot 3.33 Menu Path for creating a file under a folder	37
Screenshot 3.34 New File Dialog	38
Screenshot 3.35 Menu Path for deleting a folder	39
Screenshot 3.36 Confirmation screen for deleting a folder	40
Screenshot 3.37 Menu Path for renaming a folder	41
Screenshot 3.38 Rename Dialog for folder.....	42
Screenshot 3.39 Menu Path for copying a folder	43
Screenshot 3.40 Menu Path for pasting under a folder.....	44
Screenshot 3.41 Paste Dialog for pasting a folder under a folder	44
Screenshot 3.42 Main screen after pasting a folder under a folder	45
Screenshot 3.43 Menu Path for pasting a file under a folder.....	46
Screenshot 3.44 Paste Dialog for pasting a file under a folder.....	47
Screenshot 3.45 Main screen after pasting a file under a folder.....	47
Screenshot 3.46 Menu Path for deleting a file.....	48
Screenshot 3.47 Confirmation screen for deleting a file	49
Screenshot 3.48 Menu path for renaming a file.....	50
Screenshot 3.49 Rename Dialog for file	51
Screenshot 3.50 Menu Path for copying a file.....	52
Screenshot 3.51 Menu Path for closing a file	53
Screenshot 3.52 Main screen after closing a file	54
Screenshot 3.53 Menu Path for searching in current file.....	55
Screenshot 3.54 Search and Replace Dialog for current file	56
Screenshot 3.55 Listing search results in a tree	57
Screenshot 3.56 Menu Path for search and replace in workspace.....	58
Screenshot 3.57 Search and Replace Dialog for workspace.....	59
Screenshot 3.58 Listing search results in a tree	60
Screenshot 3.59 Menu Path for opening all projects	61
Screenshot 3.60 Main Screen after opening all projects.....	62
Screenshot 3.61 Menu Path for closing all open projects.....	63
Screenshot 3.62 Main Screen after closing all open projects	64
Screenshot 3.63 Selecting a portion of text to be formatted.....	65
Screenshot 3.64 Menu Path for formatting selected text.....	66
Screenshot 3.65 Final text after being formatted.....	67
Screenshot 3.66 Menu Path for building a project	68
Screenshot 3.67 Build Log after building a project	69

Screenshot 3.68 Menu Path for building all open projects	70
Screenshot 3.69 Build Log after building all open projects	71
Screenshot 3.70 Menu Path for running a project	72
Screenshot 3.71 Initial screen of program that is run	73
Screenshot 3.72 Menu Path for debugging project.....	74
Screenshot 3.73 Main screen after debugger stops at a breakpoint.....	75
Screenshot 3.74 Opening a file to add a breakpoint	76
Screenshot 3.75 Breakpoint is displayed with a blue circle.	77
Screenshot 3.76 Opening a file to remove a breakpoint.....	78
Screenshot 3.77 Blue circle for breakpoint is removed from line.	79
Screenshot 3.78 Click “ Next ” when debugger stopped at breakpoint.....	80
Screenshot 3.79 Source code is updated after “ Next ” instruction.....	81
Screenshot 3.80 Click “ Step Out ” when debugger stopped at breakpoint	82
Screenshot 3.81 Source code is updated after “ Step Out ” instruction.....	83
Screenshot 3.82 Click “ Continue ” when debugger stopped at breakpoint.....	84
Screenshot 3.83 Source code is updated after “ Continue ” instruction	85
Screenshot 3.84 Click “ Step Into ” when debugger stopped at breakpoint.....	86
Screenshot 3.85 Source code is updated after “ Step Into ” instruction	87
Screenshot 3.86 Double Click “ Details ” icon of a call stack row.....	88
Screenshot 3.86 VERD Studio navigates to source file for the requested call stack....	89
Screenshot 3.86 Double Click “ Details ” icon of a program variable.....	90
Screenshot 3.87 Content of variable displayed by VERD Studio	91
Screenshot 3.88 Keep mouse cursor a program variable for more than 500 ms	92
Screenshot 3.89 Content of variable displayed by VERD Studio	93
Screenshot 3.90 Double Click “ Details ” icon of a table row	94
Screenshot 3.91 Content of table row displayed by VERD Studio	95
Screenshot 3.92 Double Click “ Details ” icon of a type ref to field	96
Screenshot 3.93 Content of type ref to field displayed by VERD Studio	97

1. VERD Studio

The VERD Studio is an integrated development environment that allows a user to upload VERD project files, compile and run VERD projects, and debug them. Users can also search and replace in files within VERD Studio.

VERD Studio works with workspaces. A workspace is simply a collection of projects grouped together. In VERD Studio user can create new workspaces or add existing workspaces. In initial screen of VERD Studio user can choose from one of the workspaces that are previously added or created.

VERD Studio keeps mainly three types of files to maintain its operations. First file that VERD Studio keeps is login file. Login file keep track of information such as workspace path and name about workspaces that are previously created or added. Second file that VERD Studio keeps is workspace files. Workspace files are in XML format and include information about the projects such as project path, project name and breakpoints that projects have. Last file that VERD Studio keeps is project files. Each project in a workspace should have a project file. Project files are in XML format and include information about folder structure of project and files included in project.

VERD Studio supports Unicode characters in workspace paths, folder paths, file paths and file names. VERD Studio requires all files including login, workspace, project and source code files to be encoded in UTF-8 format. VERD Studio also supports Unicode characters during searching and replacing source code files.

VERD Studio supports two types of search and replace operation. First type of search and modify operation that VERD Studio supports, is to search and replace in current file. In this type of search and replace, VERD Studio just uses the currently processing file. In second type search and replace operation that VERD Studio supports search and replace in all projects that are currently opened. In this type of search and replace, VERD Studio uses all files within projects that are already opened.

Source code files that VERD Studio uses should have the “.v” extension which indicates that source code file is VERD language source code file.

All functions that VERD Studio supports are listed below. Each function of VERD Studio is explained in detail in **Section 3**.

- Create new Workspace
- Add existing Workspace
- Select from previously added Workspaces
- Import Existing Project
- Create New VERD Project
- Open a closed VERD Project
- Close an opened VERD Project
- Create New Folder under a Project
- Create New File under a Project
- Delete an existing VERD project
- Remove an existing VERD project from workspace
- Rename a VERD project
- Paste a copied folder under a VERD project
- Paste a copied file under a VERD project
- Create New Folder under a Folder
- Create New File under a Folder
- Delete an existing folder under a VERD project
- Rename a folder under a VERD project
- Copy a folder under a VERD project
- Paste a copied folder under a folder of VERD project
- Paste a copied file under a folder of VERD project
- Delete a file under a VERD project
- Rename a file under a VERD project
- Copy a file under a VERD project
- Close an already opened file under a VERD project
- Search and Replace in the current file or workspace
- Open all projects in a Workspace
- Close all open projects in a Workspace
- Format Selected Text
- Build, Run and Debug VERD Project
- Build Workspace

2. Files used by VERD Studio

2.1. Login File

Login file keep track of information such as workspace path and name about workspaces that are previously created or added. An example of login file is given below.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<Verd_Login_File >
    <FileVersion major="1" minor="1" />
    <Workspace workspacename="workspace" workspacepath="C:\Finance\Finance.xml" />
    <Workspace workspacename="Sales" workspacepath="C: \Sales\Sales.xml" />
    <LastWorkspace workspacename="Sales" workspacepath="C:\akkus\Sales\Sales.xml" />
</Verd_Login_File>
```

File Type 2.1 *Login File Example*

2.2. Workspace File

Workspace file is in XML format and include information about the projects such as project path, project name and breakpoints that projects have. An example of workspace file is given below.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<Verd_Workspace_File >
    <FileVersion major="1" minor="1" />
    <Project projectname="PRSSPD" projectpath="C:\ Sales\PRSSPD\PRSSPD.xml"
openstate="1">
        <Breakpoint filepath="\src\sspd_details.v" lineno="956" state="2" />
        <Breakpoint filepath="\src\sspd_details.v" lineno="717" state="2" />
        <Breakpoint filepath="\src\sspd_details.v" lineno="677" state="2" />
        <Breakpoint filepath="\src\sspd_details.v" lineno="1992" state="2" />
    </Project>
    <Project projectname="PRSSPO" projectpath="C: \Sales\PRSSPO\PRSSPO.xml"
openstate="0">
        <Breakpoint filepath="\src\so_item_cat_det.v" lineno="27" state="2" />
        <Breakpoint filepath="\src\sales_pricing.v" lineno="821" state="2" />
    </Project>
    <Project projectname="PRTSGD" projectpath="C:\Sales\PRTSGD\PRTSGD.xml"
openstate="0">
        <Breakpoint filepath="\src\tsgd_details.v" lineno="1880" state="2" />
        <Breakpoint filepath="\src\tsgd_helper.v" lineno="815" state="2" />
        <Breakpoint filepath="\src\tsgd_save.v" lineno="836" state="2" />
    </Project>
    <Project projectname="PRTSGO" projectpath="C:\Sales\PRTSGO\PRTSGO.xml"
openstate="0"/>
        <Project projectname="RPSSPL" projectpath="C:\Sales\RPSSPL\RPSSPL.xml"
openstate="0"/>
    </Verd_Workspace_File>
```

2.3. Project File

Project file is in XML format and include information about folder structure of project and files included in project. Each project in a workspace should have a project file. An example of project file is given below.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<Verd_Project_File >
  <Project projectname="BKUAGB" >
    <Folder foldername="src" >
      <Unit filename="data_def_com.v"/>
      <Unit filename="system_lock.v"/>
      <Unit filename="uagb_data_def.v"/>
      <Unit filename="uagb_details.v"/>
      <Unit filename="uagb_grid.v"/>
      <Unit filename="uagb_main.v"/>
      <Unit filename="uagb_new.v"/>
    </Folder>
  </Project>
</Verd_Project_File>
```

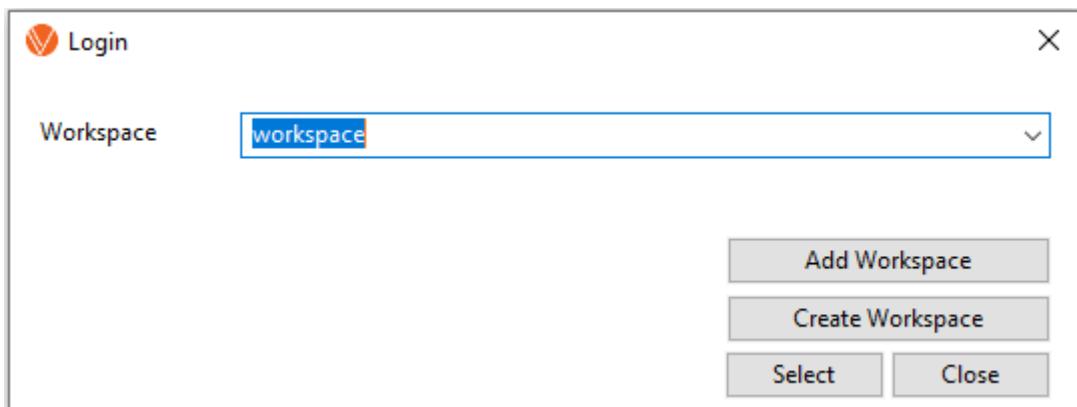
File Type 2.3 Project File Example

3. Functions of VERD Studio

3.1. Create New Workspace

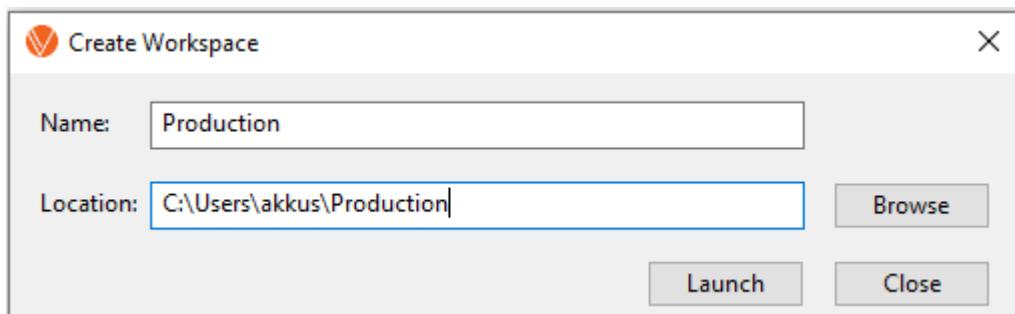
VERD Studio works with workspaces. A workspace is simply a collection of projects grouped together. In VERD Studio user can create new workspaces. Steps for creating new workspace are listed below.

- Click on “**Create Workspace**” button in initial screen of VERD Studio. (*Screenshot 3.1*)



Screenshot 3.1 Initial screen of VERD Studio

- On the next screen, enter name of workspace and the folder in which workspace file would be created in by VERD Studio. After filling name of workspace and the folder in which workspace file would be created in, click on “**Launch**” button to create new workspace. VERD Studio automatically creates a workspace file for the workspace and updates the login file with this new workspace. (*Screenshot 3.2*)

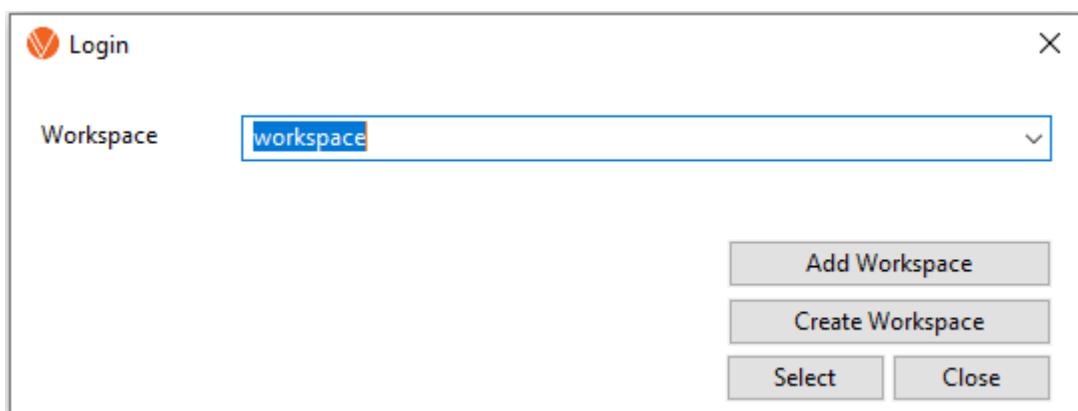


Screenshot 3.2 Create Workspace Screen

3.2. Add Existing Workspace

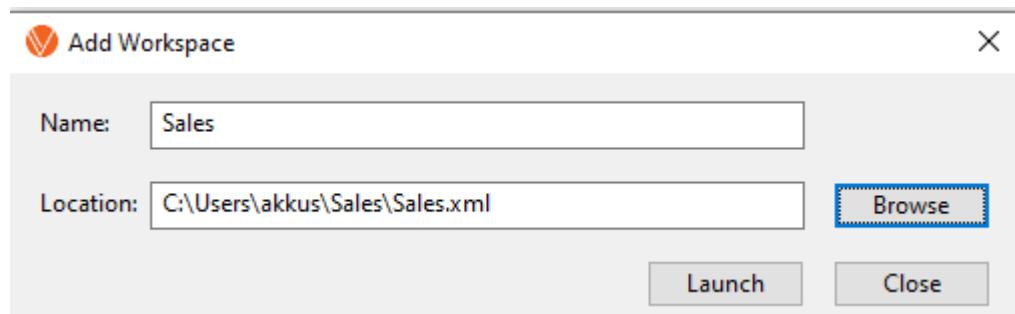
VERD Studio works with workspaces. A workspace is simply a collection of projects grouped together. In VERD Studio user can add existing workspaces to VERD Studio. Steps for creating new workspace are listed below.

- Click on “**Add Workspace**” button in initial screen of VERD Studio. (*Screenshot 3.3*)



Screenshot 3.3 Initial screen of VERD Studio

- On the next screen, enter name of workspace and the path of workspace file that would be read by VERD Studio. Workspace file must exist for adding a workspace to VERD Studio. Details about format of VERD workspace file is given in **Section 2**. After filling name and path of workspace click on “**Launch**” button to create new workspace. VERD Studio updates the login file with this newly added workspace. (*Screenshot 3.4*)

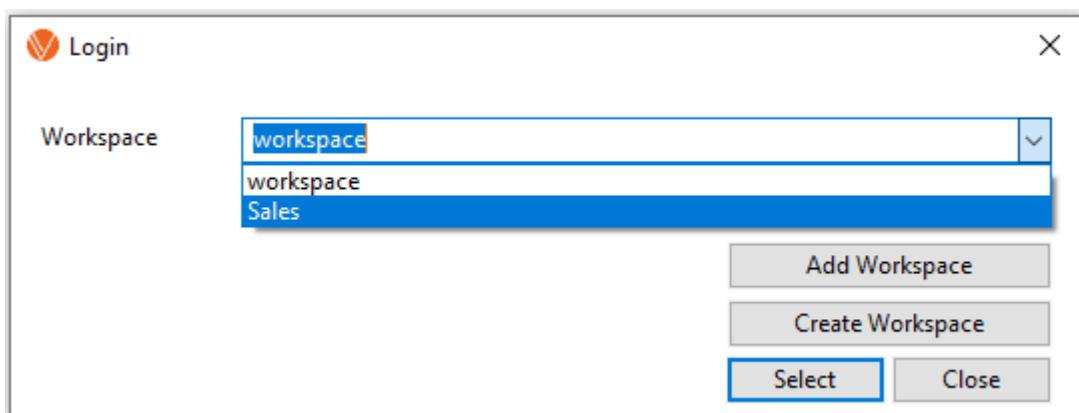


Screenshot 3.4 Add Workspace Screen

3.3. Select from previously added Workspaces

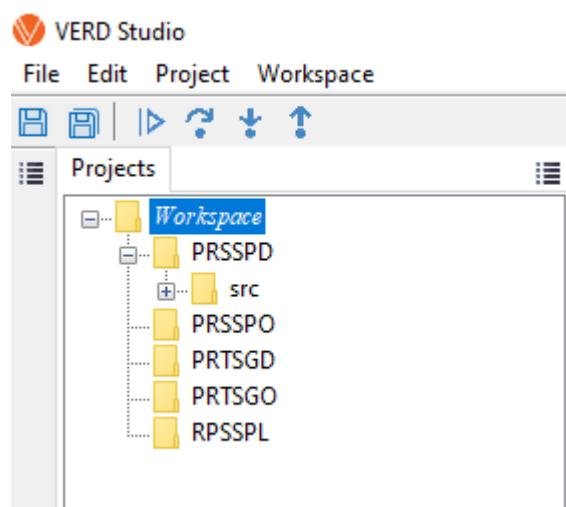
VERD Studio works with workspaces. A workspace is simply a collection of projects grouped together. In VERD Studio user can select from workspaces that are previously added or created in VERD Studio to work on it. Steps for selecting a workspace are listed below.

- Choose one of the workspace listed in workspace dropdown list in initial screen of VERD Studio and click on “Select” button. (*Screenshot 3.5*)



Screenshot 3.5 Initial screen with dropdown expanded

- VERD Studio reads the workspace file for selected workspace and loads the projects in workspace into VERD Studio. (*Screenshot 3.6*)

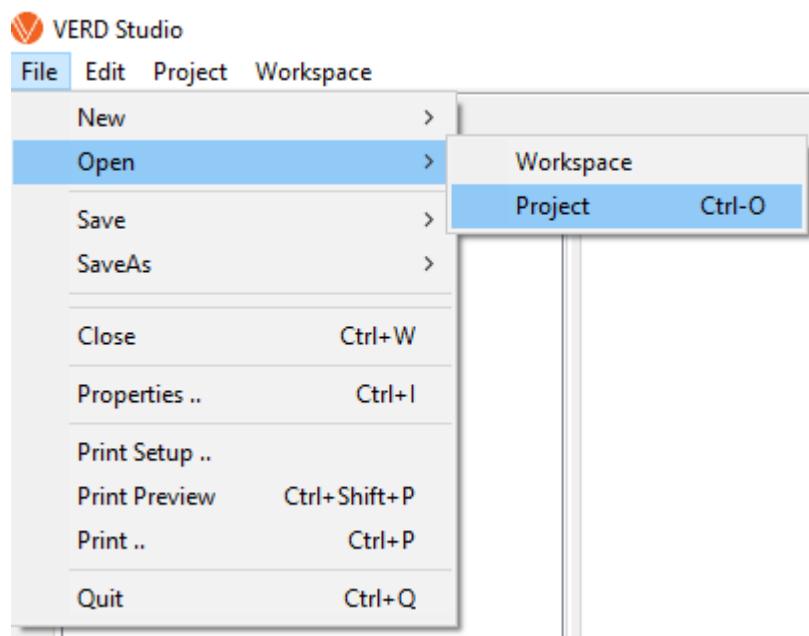


Screenshot 3.6 Main screen after loading workspace file

3.4. Import Existing Project

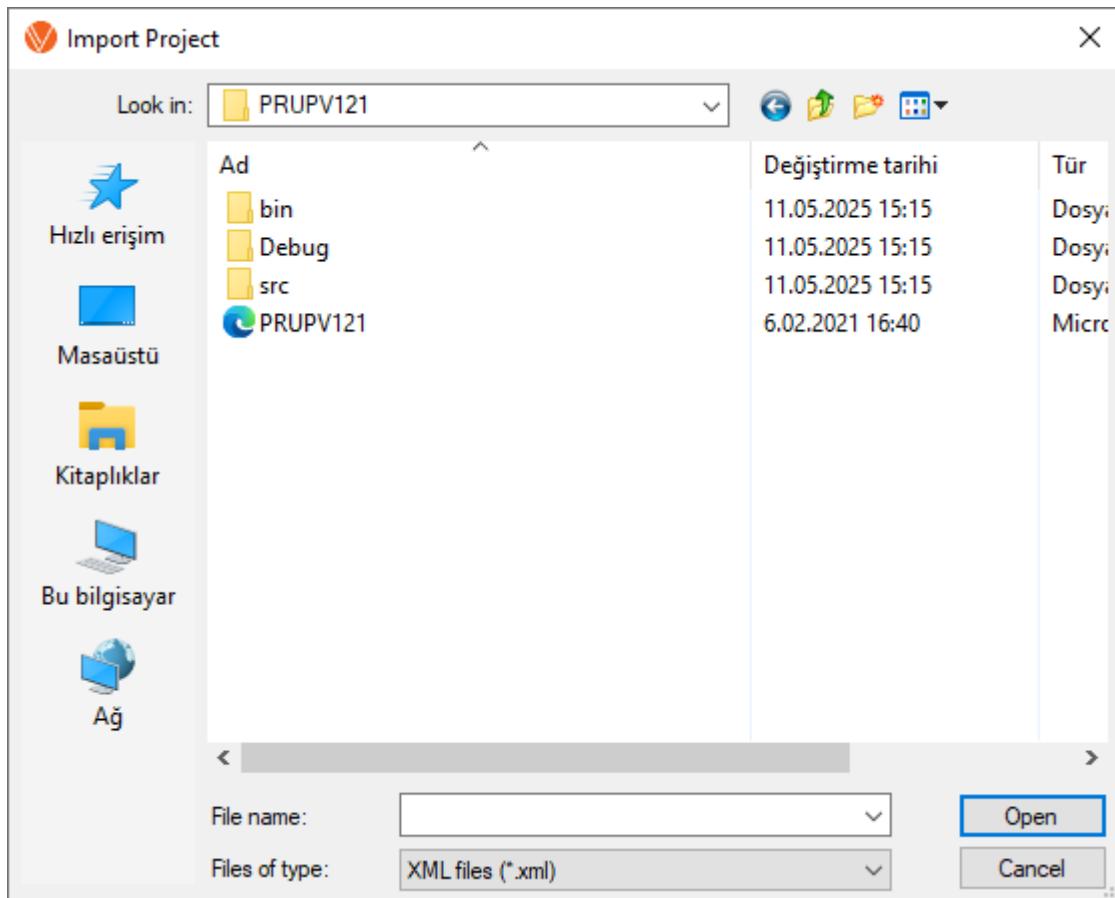
A pre-existing project can be imported to VERD Studio. The project files used to import to VERD Studio should be in XML format. Details about format of VERD project file is given in **Section 2**. The steps in the importing existing project are listed below, along with descriptions and screenshots:

- To import a project file, select **File->Open->Project**. (*Screenshot 3.7*)



Screenshot 3.7 Menu Path for Open Project

- On the next screen, select the VERD project file and press the Open button to import the project into workspace. (*Screenshot 3.8*)

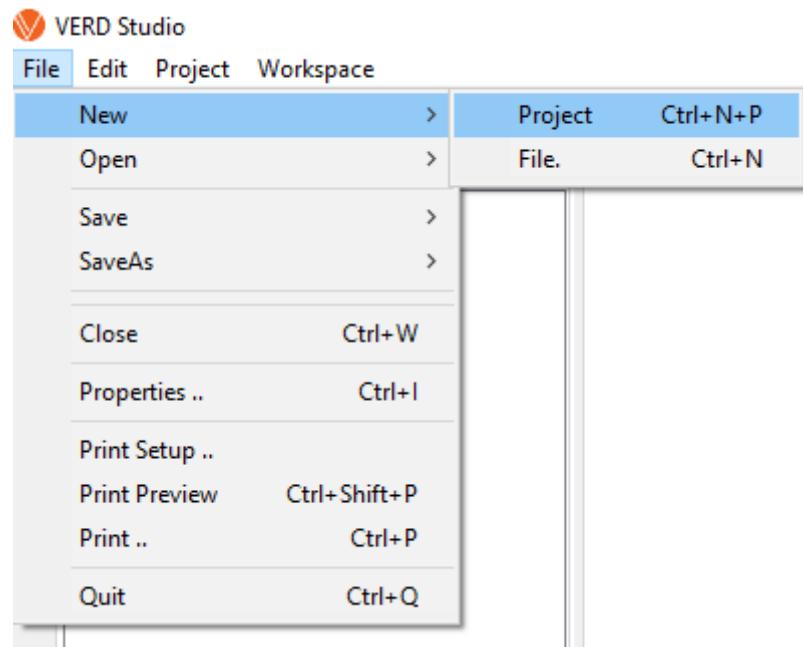


Screenshot 3.8 Import Project File Dialog

3.5. Create New VERD Project

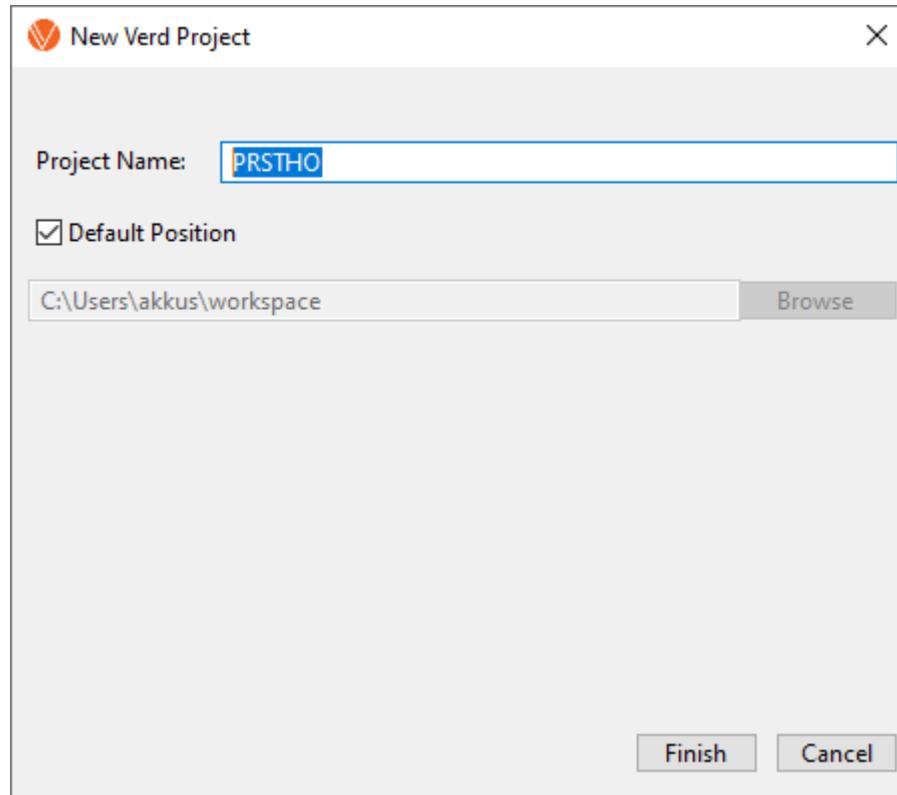
A new VERD project can be created in VERD Studio. The steps for creating a new VERD project are listed below, along with descriptions and screenshots.

- To create a project file, select **File->New->Project** menu path. (*Screenshot 3.9*)



Screenshot 3.9 Menu Path for New Project

- On the next screen, enter the project name and click the Finish button.
(Screenshot 3.10)

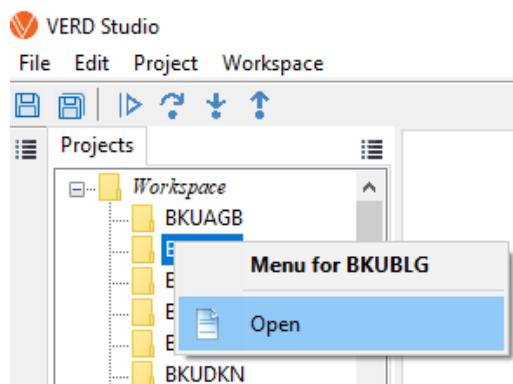


Screenshot 3.10 New Project Create Screen

3.6. Open a closed VERD Project

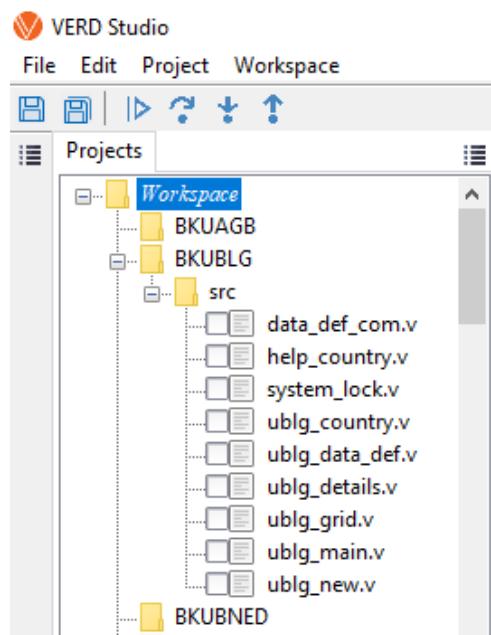
A new VERD project that is already closed can be opened VERD Studio. The steps for opening a VERD project are listed below, along with descriptions and screenshots.

- To open an existing VERD projects, right-click after selecting a project that is closed. In the menu that appears, click on **Open**. (*Screenshot 3.11*)



Screenshot 3.11 Menu Path for opening a closed project

- The project is opened in workspace, a user can click + button appeared near project name to display contents of a VERD project. (*Screenshot 3.12*)

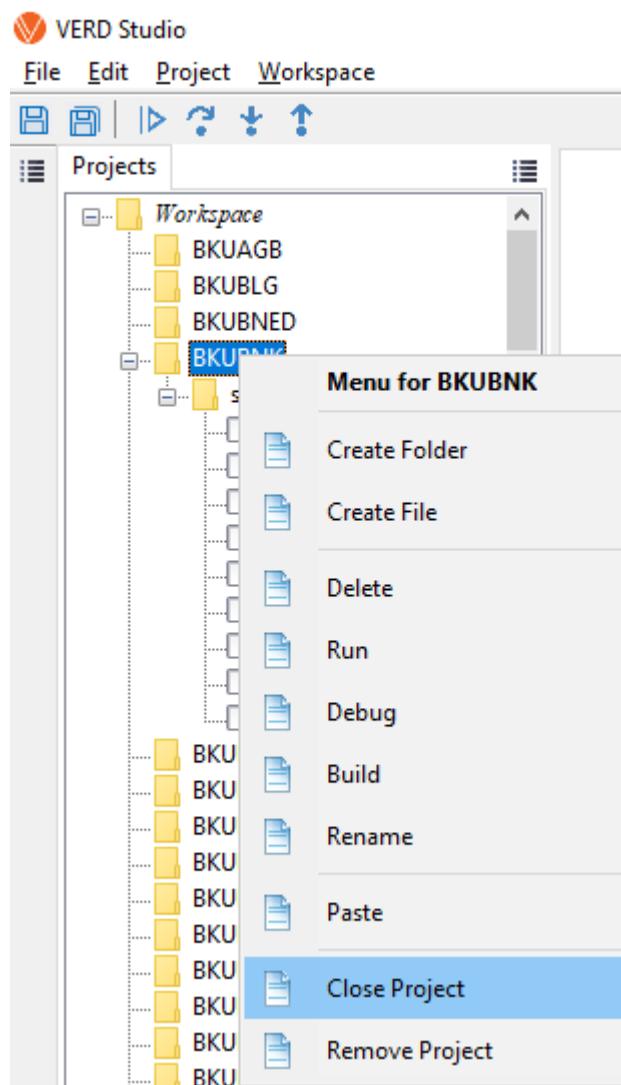


Screenshot 3.12 Main screen after the project is opened

3.7. Close an opened VERD Project

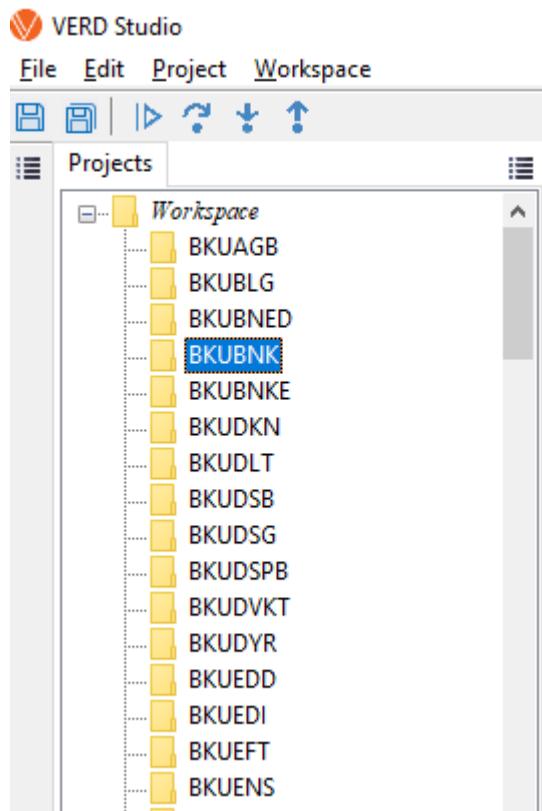
A new VERD project that is already opened can be closed VERD Studio. The steps for closing a VERD project are listed below, along with descriptions and screenshots.

- To close an existing VERD projects, right-click after selecting a project that is already opened. In the menu that appears, click on **Close**. (*Screenshot 3.13*)



Screenshot 3.13 Menu Path for closing an opened project

- The project is closed in workspace, + button previously appeared near project name disappeared. (**Screenshot 3.14**)

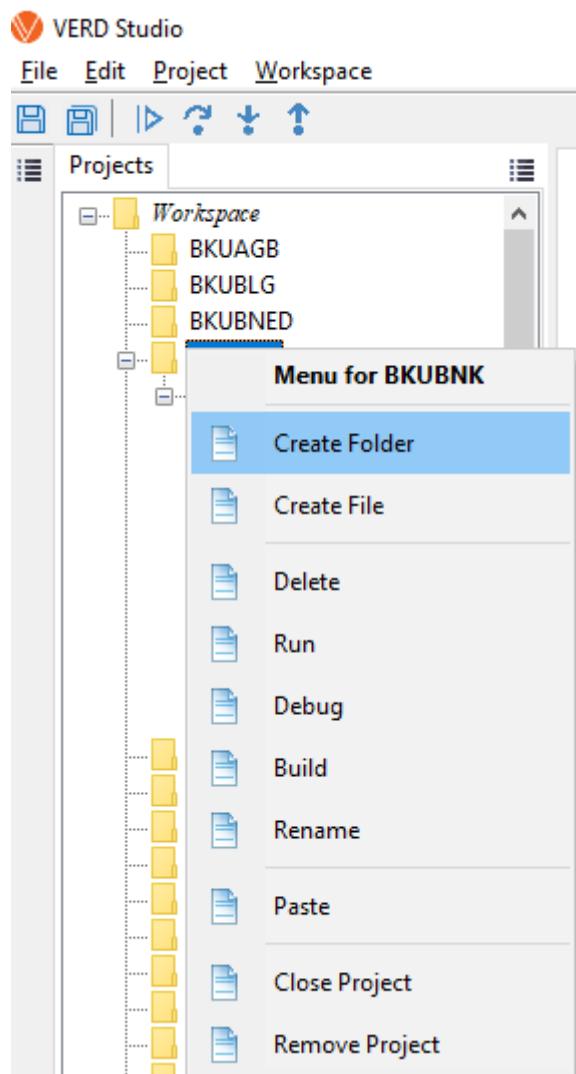


Screenshot 3.14 Main screen after the project is closed

3.8. Create New Folder under a Project

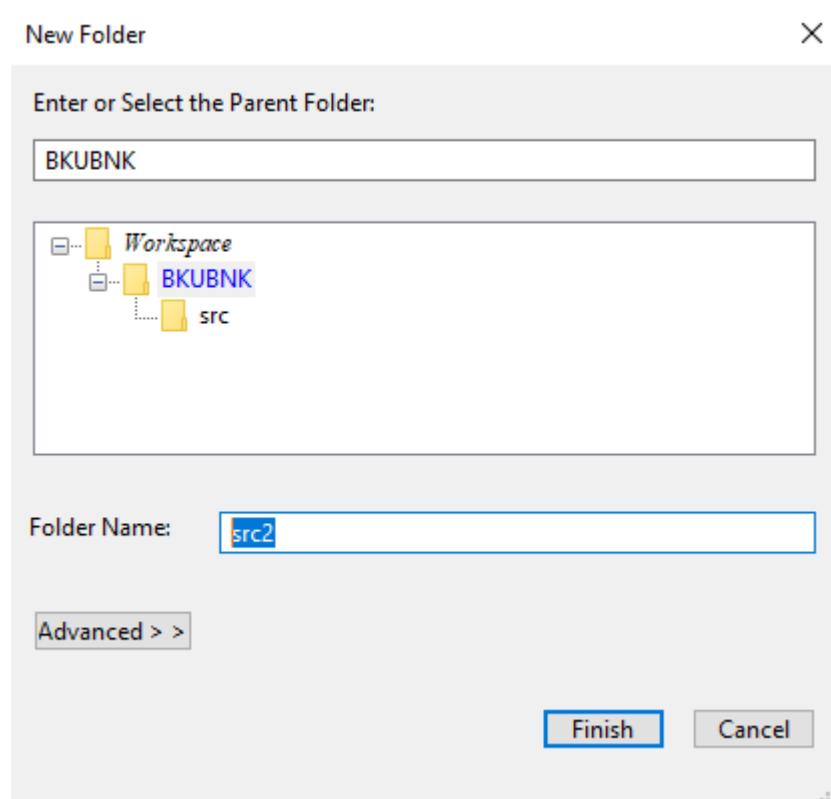
New directories can be created within VERD project in VERD Studio. The steps for creating a new directory under a project are listed below with descriptions and screenshots.

- To create a new folder, right-click after selecting a project. In the menu that appears, click on **Create Folder**. (*Screenshot 3.15*)



Screenshot 3.15 Menu Path for creating a folder under a project

- On the next screen, enter the directory name in the **Folder Name** field. Click the **Finish** button to create the new directory. (*Screenshot 3.16*)

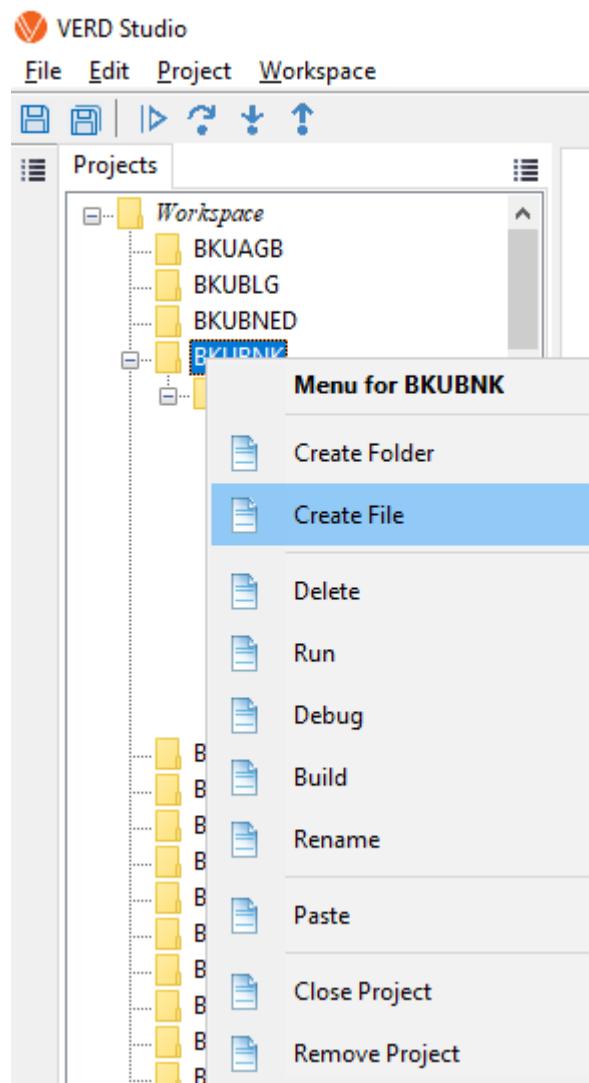


Screenshot 3.16 New Folder Dialog

3.9. Create New File under a Project

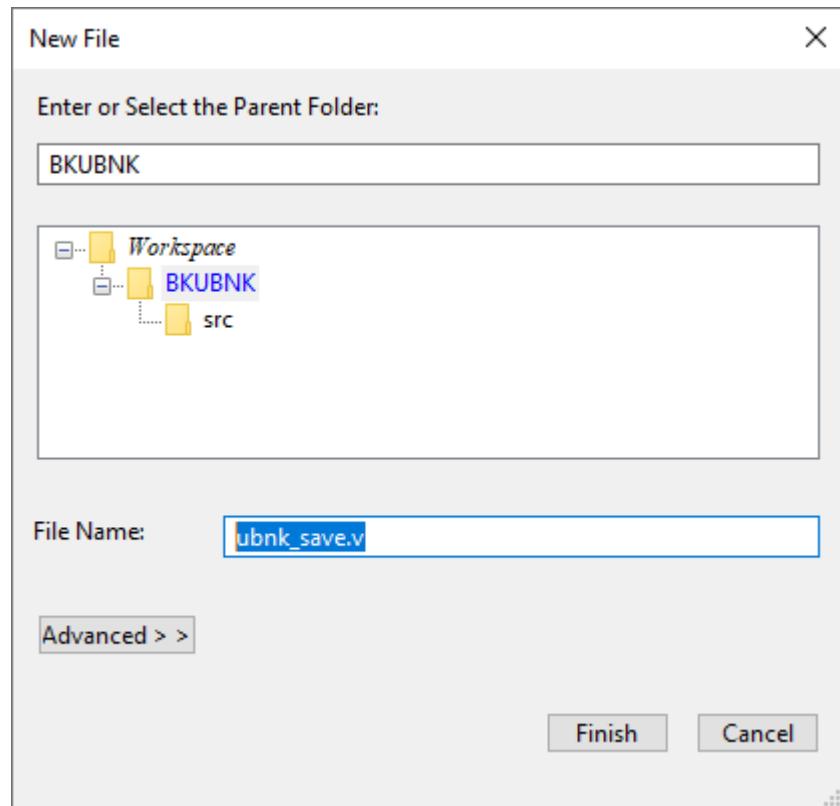
New files can be created under a project in VERD Studio. The steps for creating a new file under a project are listed below, along with descriptions and screenshots.

- To create a new file under a project, right-click after selecting a project. In the menu that appears, click on **Create File**. (*Screenshot 3.17*)



Screenshot 3.17 Menu Path for creating a file under a project

- On the next screen, enter the file name in the **File Name** field. Click the **Finish** button to create the new file. (*Screenshot 3.18*)

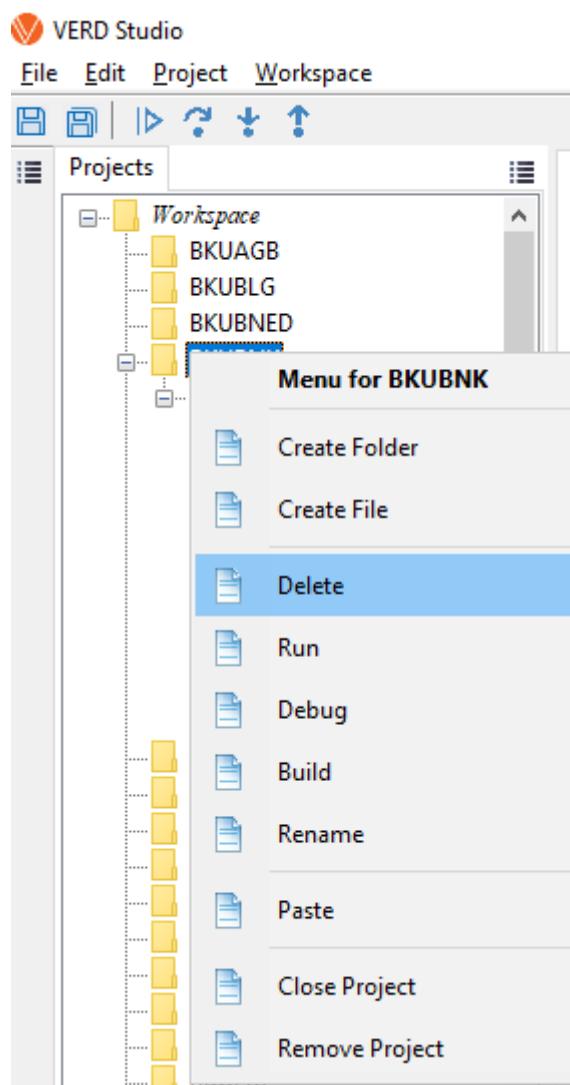


Screenshot 3.18 New File Dialog

3.10. Delete an existing VERD project

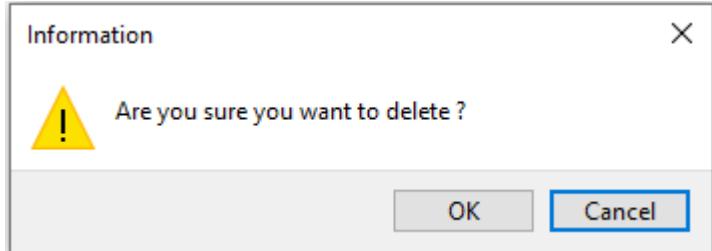
A VERD project can be deleted in VERD Studio. When a VERD project is deleted it is removed from workspace and all files associated with project is also deleted from file system. The steps for deleting a VERD project are listed below, along with descriptions and screenshots.

- To delete a VERD project, right-click after selecting a project. In the menu that appears, click on **Delete**. (*Screenshot 3.19*)



Screenshot 3.19 Menu Path for deleting a project

- On the next screen click on the **OK** button to remove VERD project from workspace and delete all files associated with VERD project from file system.
(Screenshot 3.20)

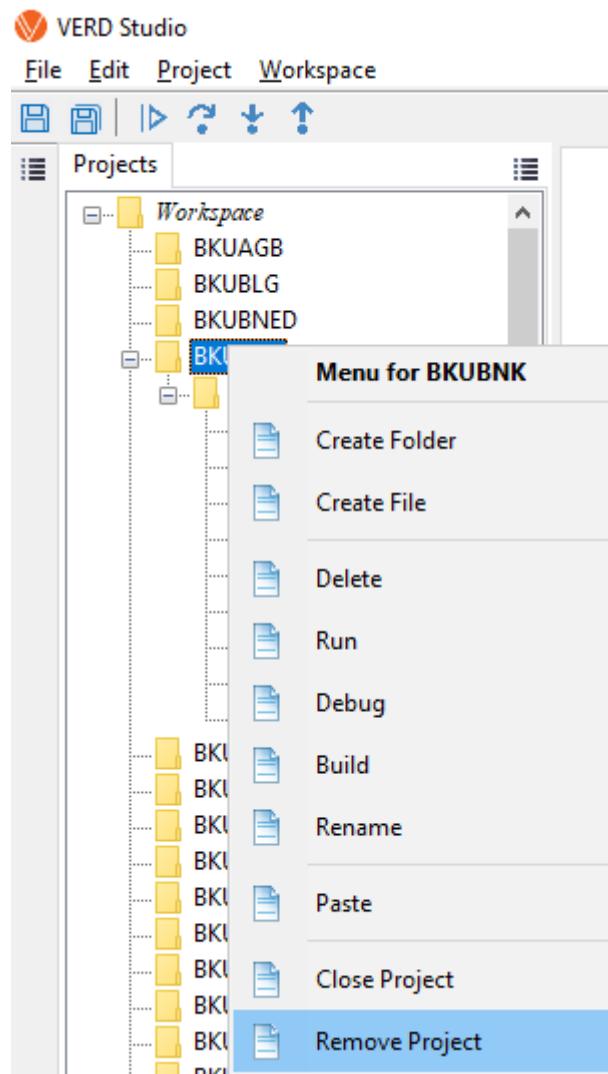


Screenshot 3.20 Confirmation screen for deleting a project

3.11. Remove an existing VERD project from workspace

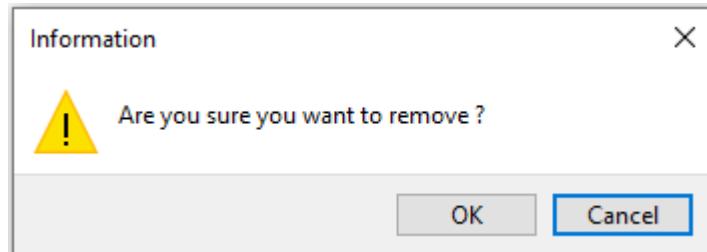
A VERD project can be removed from workspace in VERD Studio. When a VERD project is removed from workspace it is only removed from workspace, files associated with project are not deleted from file system. The steps for removing a VERD project from workspace are listed below, along with descriptions and screenshots.

- To remove a VERD project, right-click after selecting a project. In the menu that appears, click on **Remove**. (*Screenshot 3.21*)



Screenshot 3.21 Menu Path for removing a project

- On the next screen click on the **OK** button to remove VERD project from workspace. (*Screenshot 3.22*)

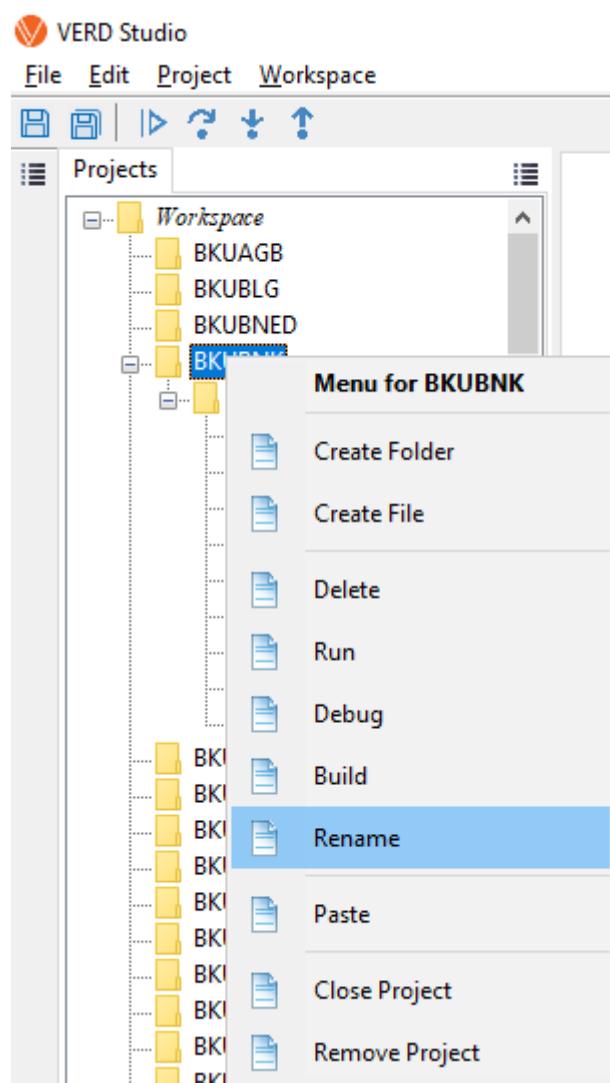


Screenshot 3.22 Confirmation screen for removing a project

3.12. Rename a VERD project

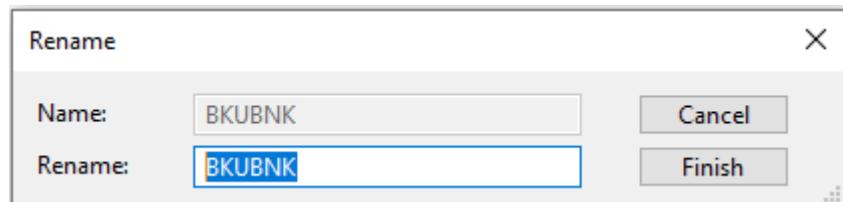
A VERD project can be renamed in VERD Studio. When a VERD project is renamed, workspace file and VERD project file is updated according to this change.. The steps for renaming a VERD project are listed below, along with descriptions and screenshots.

- To rename a VERD project, right-click after selecting a project. In the menu that appears, click on **Rename**. (*Screenshot 3.23*)



Screenshot 3.23 Menu Path for renaming a project

- On the next screen enter **Rename** field with new project name and click on **OK** button (*Screenshot 3.24*)

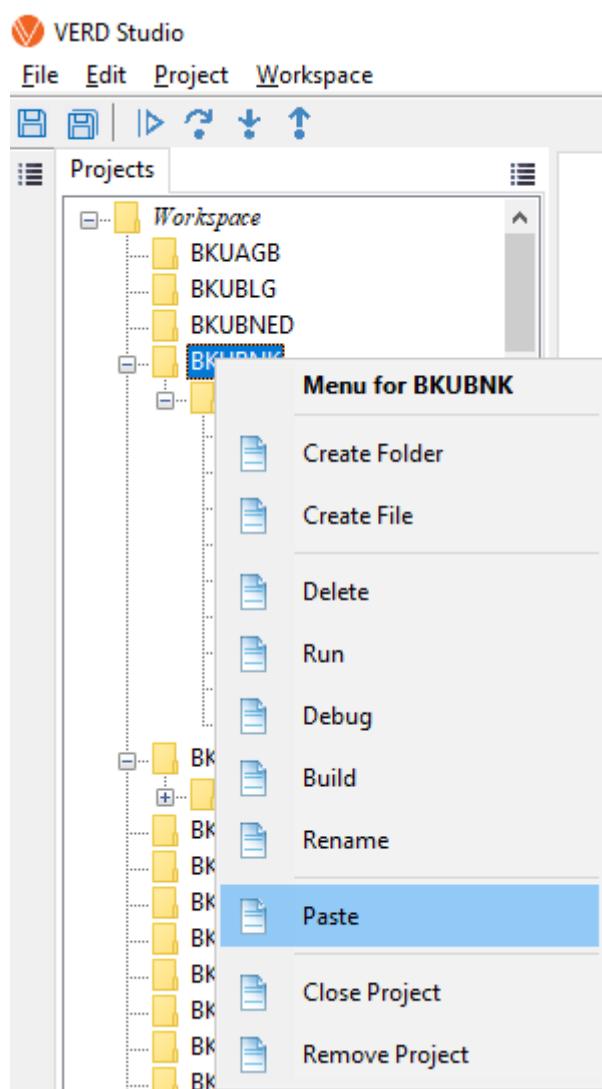


Screenshot 3.24 Rename Dialog for project

3.13. Paste a copied folder under a VERD project

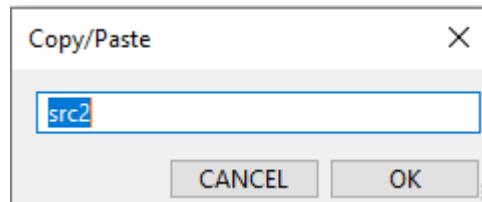
A folder that is copied can be pasted to another project in VERD Studio. Copied folder can be from a different VERD project than copied VERD project. The steps for pasting a folder under VERD project are listed below, along with descriptions and screenshots.

- To paste a folder under a VERD project, right-click after selecting a VERD project. In the menu that appears, click on **Paste**. (*Screenshot 3.25*)



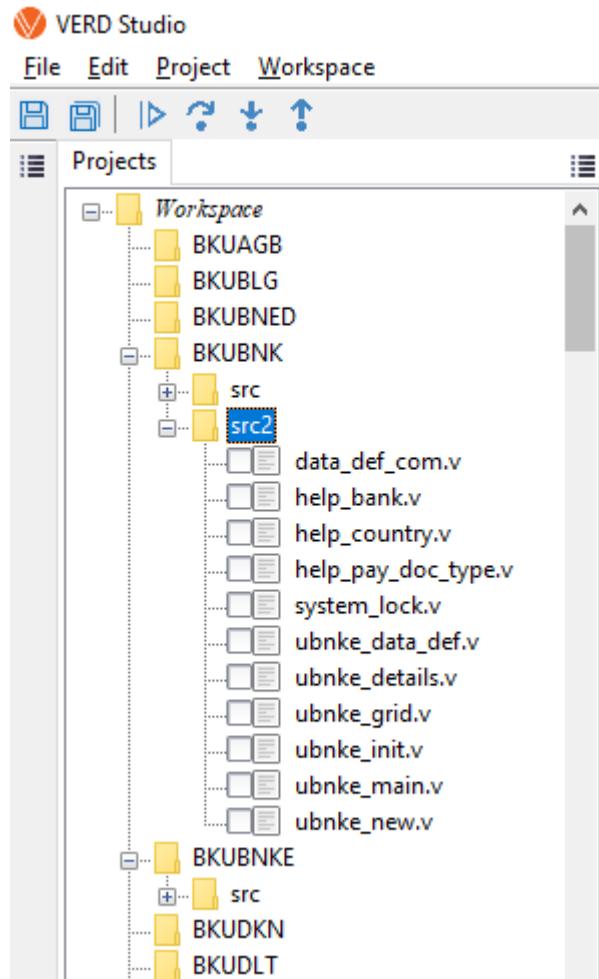
Screenshot 3.25 Menu Path for pasting a folder under a project

- On the next screen enter the new folder name of copied folder as it would appear under the pasted VERD project. (*Screenshot 3.26*)



Screenshot 3.26 Paste Dialog for pasting a folder under a project

- Previously copied folder with new folder name would be pasted to the VERD project that is selected. After pasting the folder, new folder with new folder name would appear under the VERD project. (*Screenshot 3.27*)

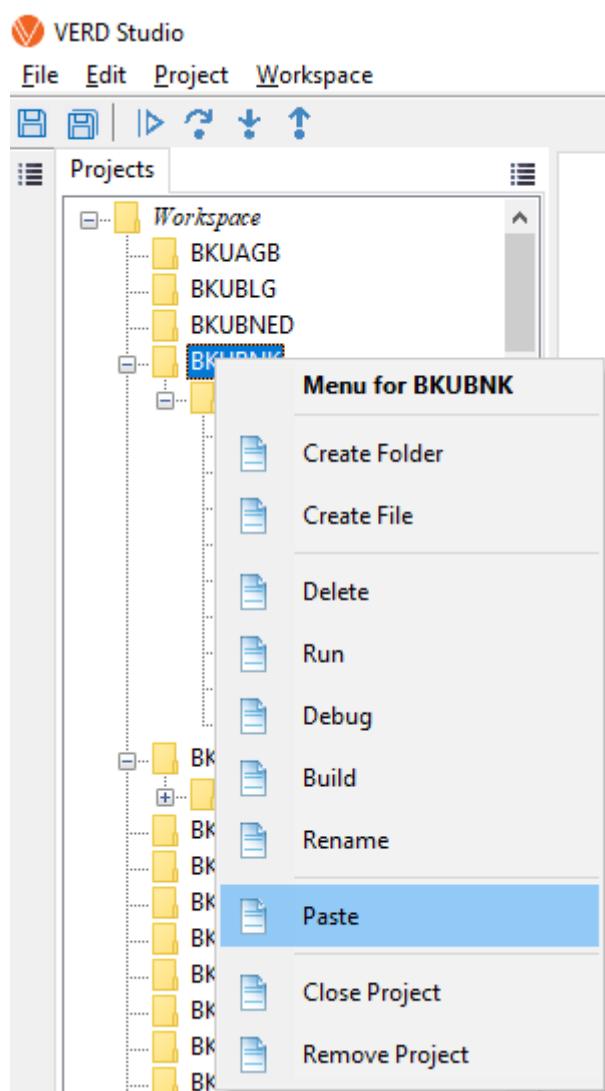


Screenshot 3.27 Main screen after pasting a folder under a project

3.14. Paste a copied file under a VERD project

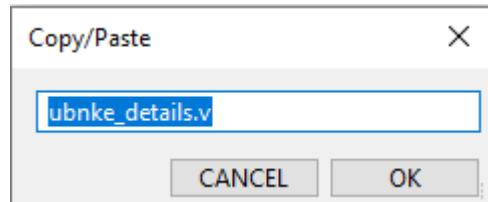
A file that is copied can be pasted to another project in VERD Studio. Copied file can be from a different VERD project than copied VERD project. The steps for pasting a file under VERD project are listed below, along with descriptions and screenshots.

- To paste a file under a VERD project, right-click after selecting a VERD project.
In the menu that appears, click on **Paste**. (*Screenshot 3.28*)



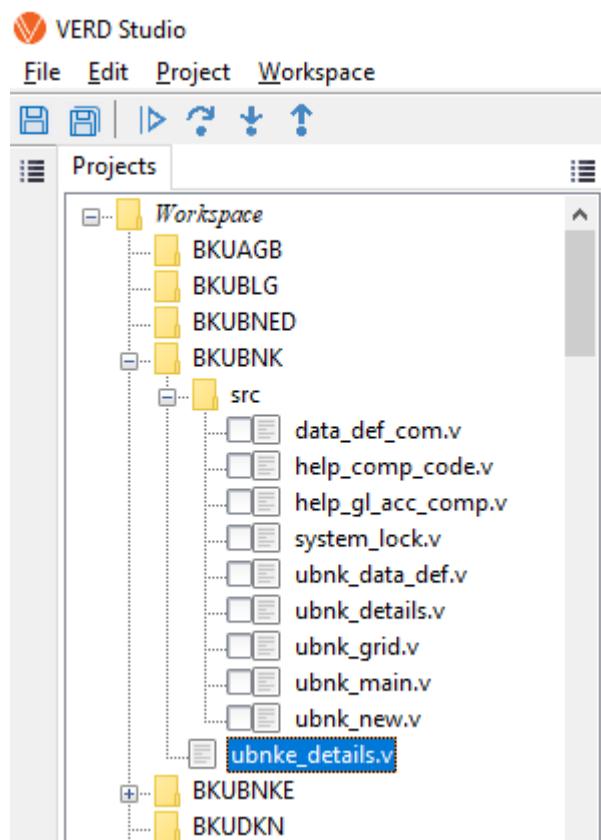
Screenshot 3.28 Menu Path for pasting a file under a project

- On the next screen enter the new file name of copied file as it would appear under the pasted VERD project. (*Screenshot 3.29*)



Screenshot 3.29 Paste Dialog for pasting a file under a project

- Previously copied file with new file name would be pasted to the VERD project that is selected. After pasting the file, new file with new file name would appear under the VERD project. (*Screenshot 3.30*)

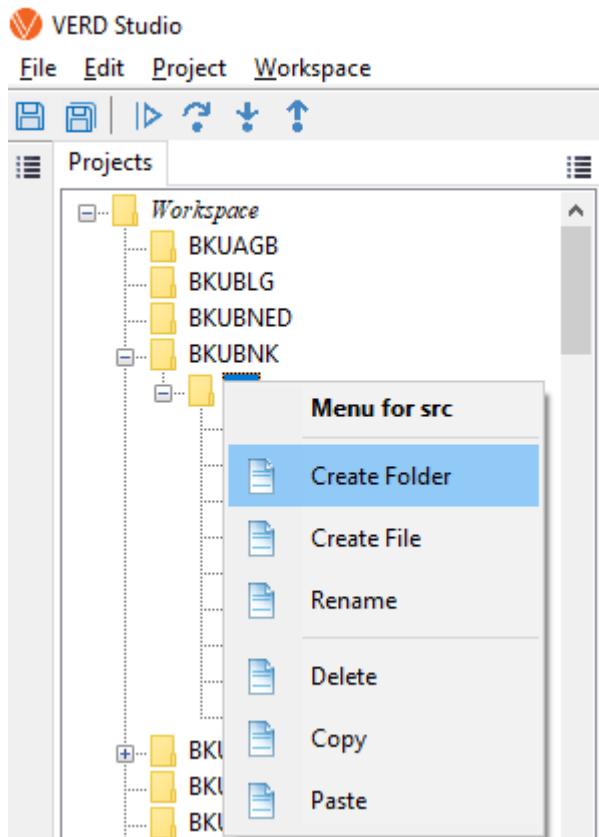


Screenshot 3.30 Main screen after pasting a file under a project

3.15. Create New Folder under a Folder

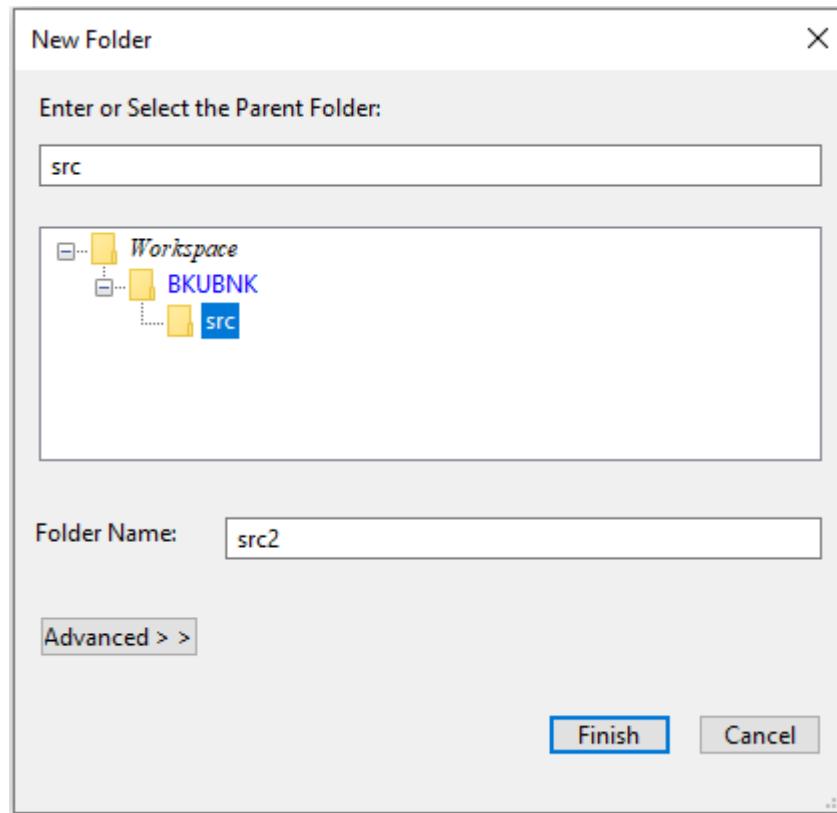
New directories can be created within a folder in VERD Studio. The steps for creating a new directory under a folder are listed below with descriptions and screenshots.

- To create a new folder, right-click after selecting a folder. In the menu that appears, click on **Create Folder**. (*Screenshot 3.31*)



Screenshot 3.31 Menu Path for creating a folder under a folder

- On the next screen, enter the directory name in the **Folder Name** field. Click the **Finish** button to create the new directory. (*Screenshot 3.32*)

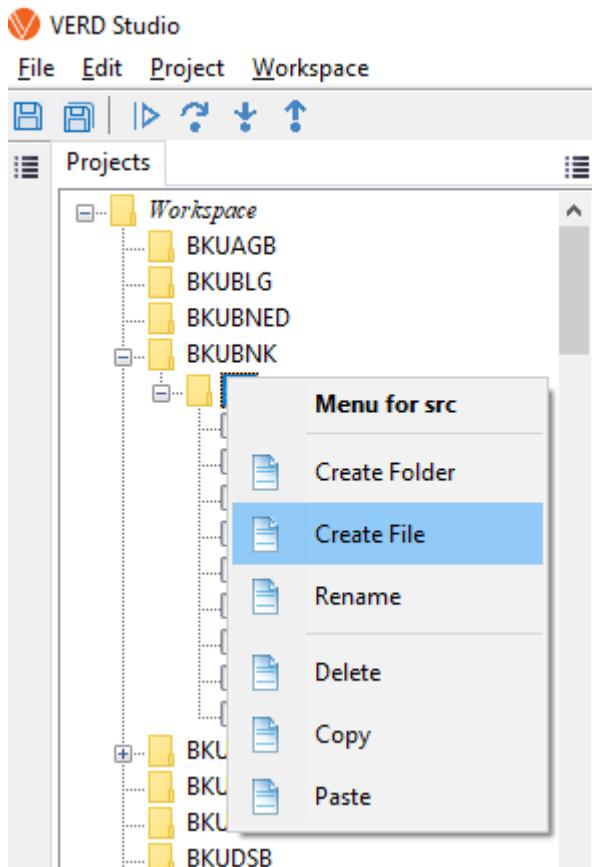


Screenshot 3.32 New Folder Dialog

3.16. Create New File under a Folder

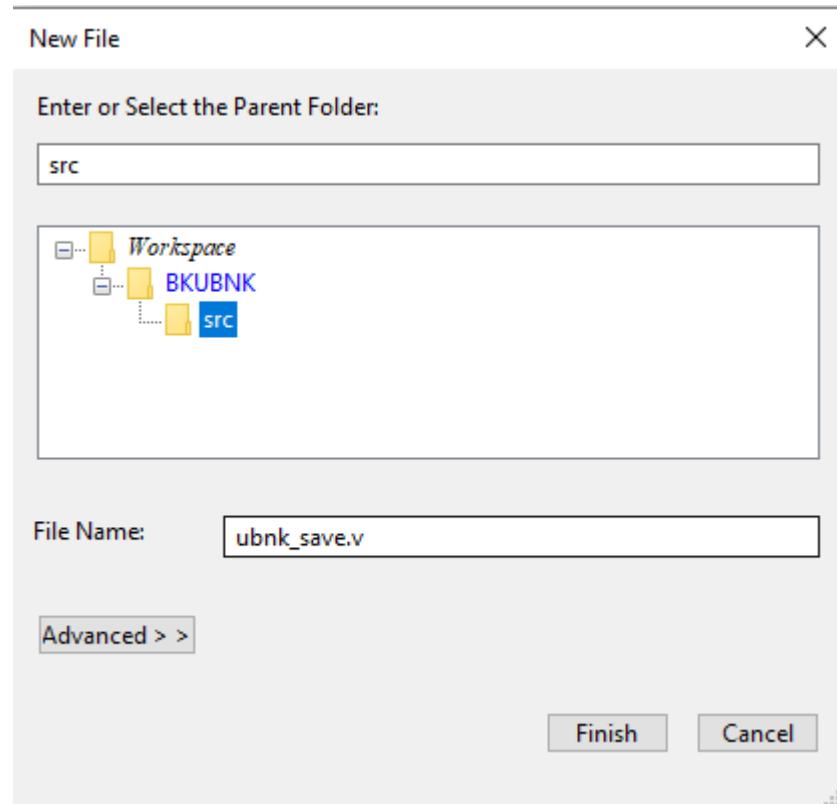
New files can be created under a project in VERD Studio. The steps for creating a new file under a project are listed below, along with descriptions and screenshots.

- To create a new file under a directory, right-click after selecting a directory. In the menu that appears, click on **Create File**. (*Screenshot 3.33*)



Screenshot 3.33 Menu Path for creating a file under a folder

- On the next screen, enter the file name in the **File Name** field. Click the **Finish** button to create the new file. (*Screenshot 3.34*)

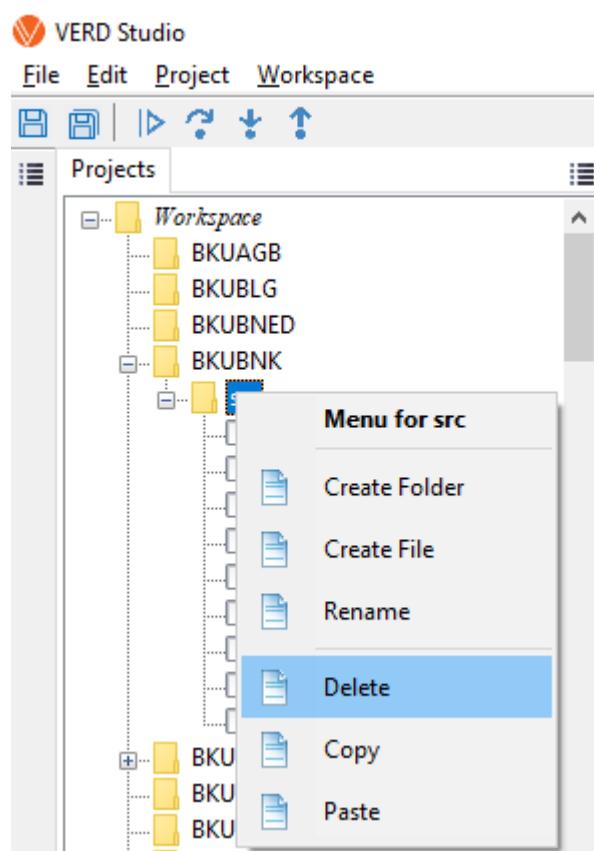


Screenshot 3.34 New File Dialog

3.17. Delete an existing folder under a VERD project

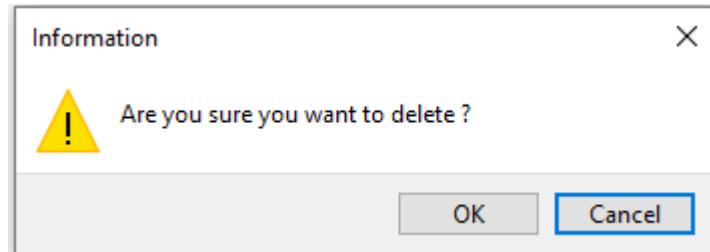
A folder under a VERD project can be deleted in VERD Studio. When the folder under a VERD project is deleted, all files associated with folder is also deleted from file system. The steps for deleting a folder under a VERD project are listed below, along with descriptions and screenshots.

- To delete a folder under a VERD project, right-click after selecting a folder. In the menu that appears, click on **Delete**. (*Screenshot 3.35*)



Screenshot 3.35 Menu Path for deleting a folder

- On the next screen click on the **OK** button to delete a folder under a VERD project and all files associated with the folder from file system. (*Screenshot 3.36*)

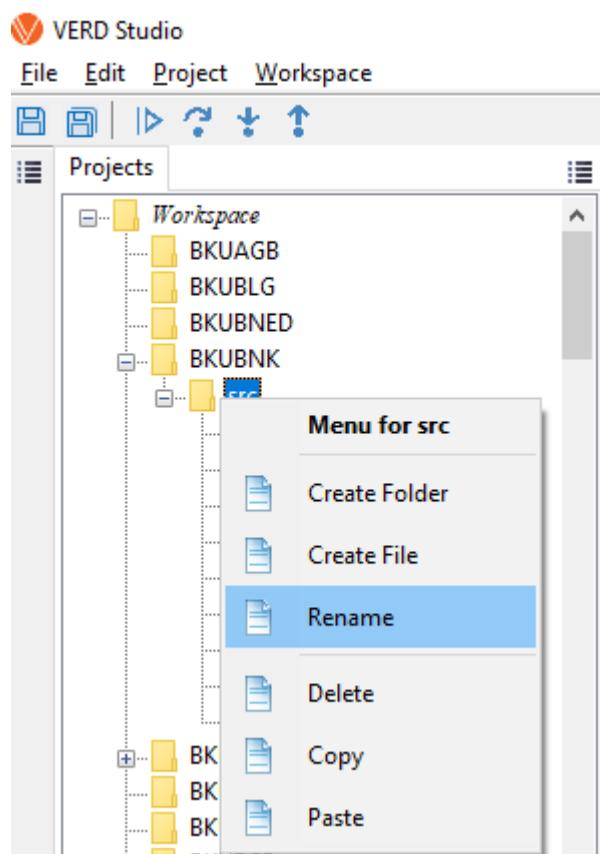


Screenshot 3.36 Confirmation screen for deleting a folder

3.18. Rename a folder under a VERD project

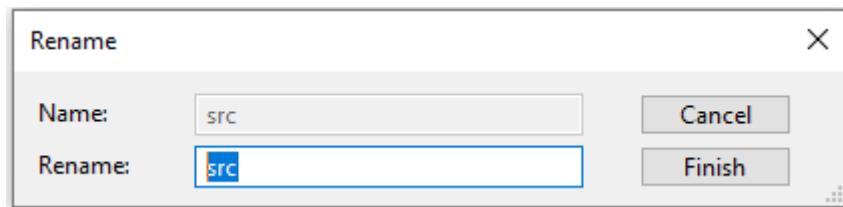
A folder under a VERD project can be renamed in VERD Studio. When a folder under a VERD project is renamed, VERD project file is updated according to this change. The steps for renaming a folder under a VERD project are listed below, along with descriptions and screenshots.

- To rename a folder under a VERD project, right-click after selecting a folder. In the menu that appears, click on **Rename**. (*Screenshot 3.37*)



Screenshot 3.37 Menu Path for renaming a folder

- On the next screen enter the **Rename** field with new folder name and click on **OK** button. (*Screenshot 3.38*)

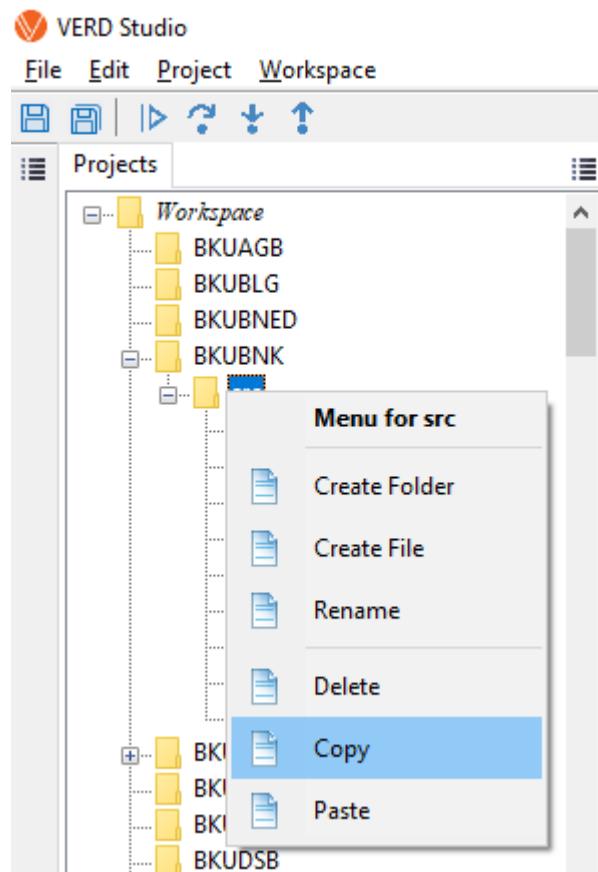


Screenshot 3.38 Rename Dialog for folder

3.19. Copy a folder under a VERD project

A folder under a VERD project can be copied in VERD Studio to be pasted afterwards. The steps for copying a folder under a VERD project are listed below, along with descriptions and screenshots.

- To copy a folder under a VERD project, right-click after selecting a folder. In the menu that appears, click on **Copy**. (*Screenshot 3.39*)

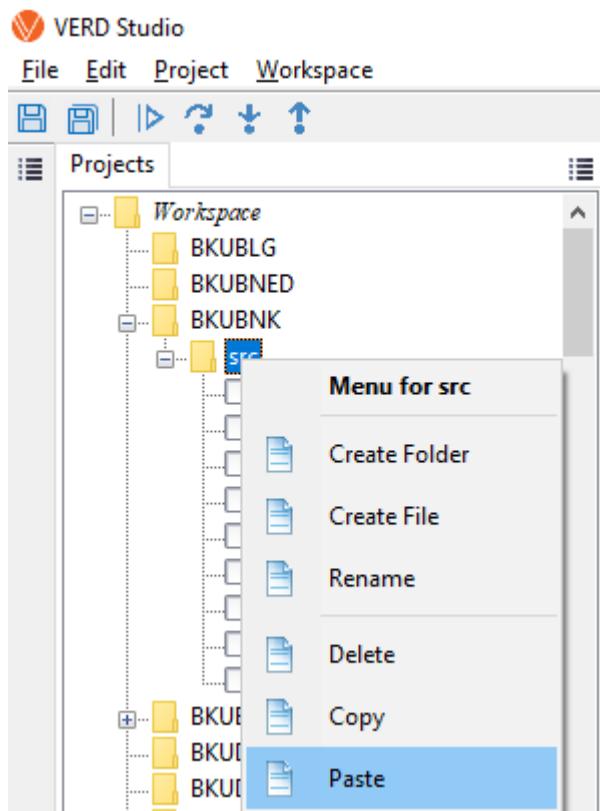


Screenshot 3.39 Menu Path for copying a folder

3.20. Paste a copied folder under a folder of VERD project

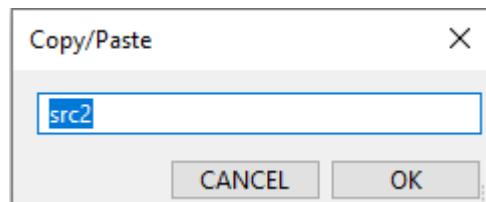
A folder that is copied can be pasted to another folder in VERD Studio. Copied and pasted folder can be from different projects. The steps for pasting a folder under a folder of VERD project are listed below, along with descriptions and screenshots.

- To paste a folder under a folder of VERD project, right-click after selecting a folder. In the menu that appears, click on **Paste**. (*Screenshot 3.40*)



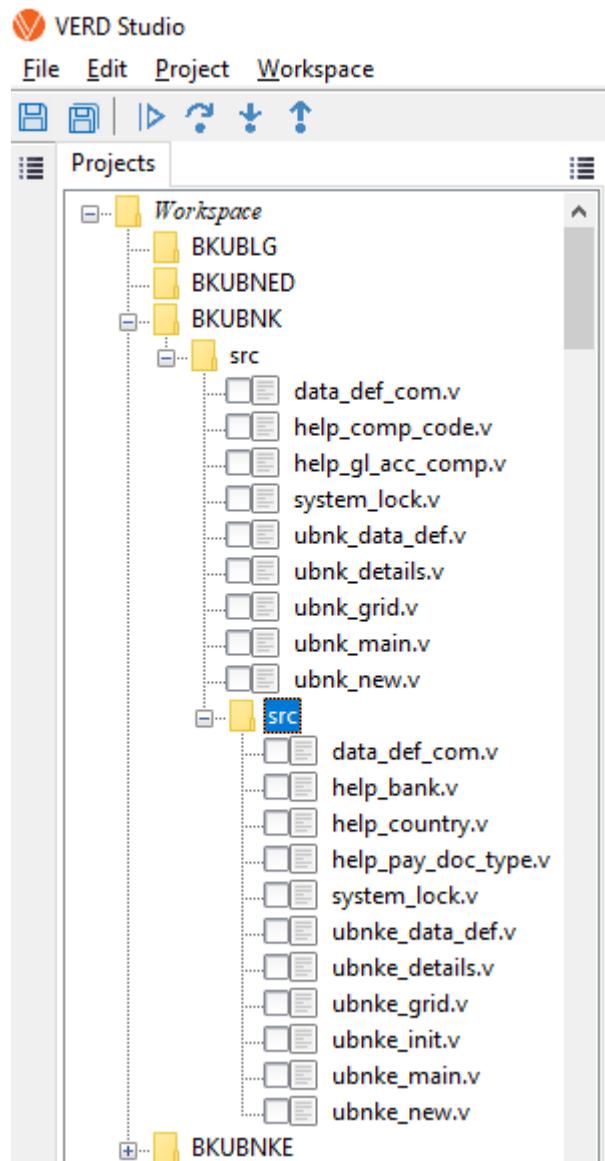
Screenshot 3.40 Menu Path for pasting under a folder

- On the next screen enter the new folder name of copied folder as it would appear under the pasted folder. (*Screenshot 3.41*)



Screenshot 3.41 Paste Dialog for pasting a folder under a folder

- Previously copied folder with new folder name would be pasted to the folder that is selected. After pasting the folder, new folder with new folder name would appear under the folder. (*Screenshot 3.42*)

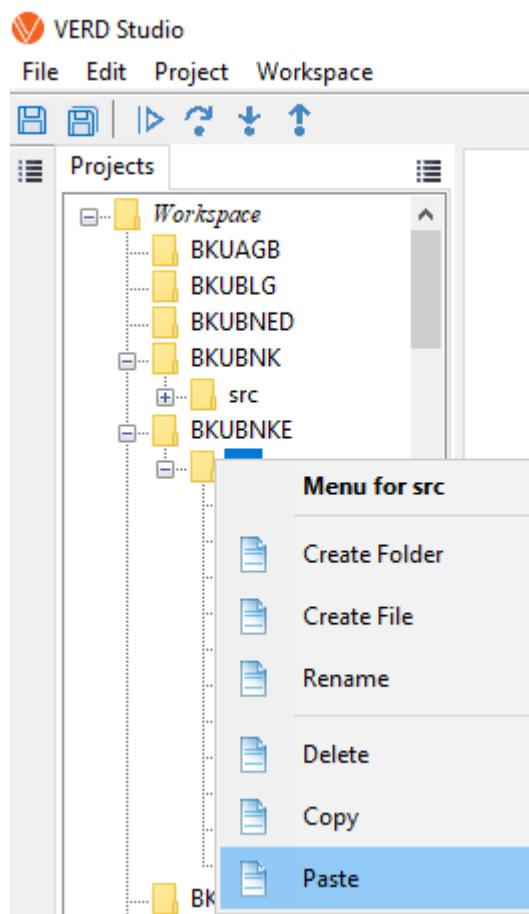


Screenshot 3.42 Main screen after pasting a folder under a folder

3.21. Paste a copied file under a folder of VERD project

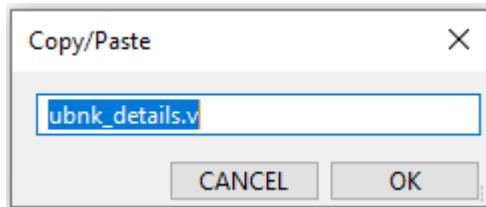
A file that is copied can be pasted to another folder in VERD Studio. Copied file and pasted folder can be from different projects. The steps for pasting a file under a folder of VERD project are listed below, along with descriptions and screenshots.

- To paste a file under a folder of VERD project, right-click after selecting a folder. In the menu that appears, click on **Paste**. (*Screenshot 3.43*)



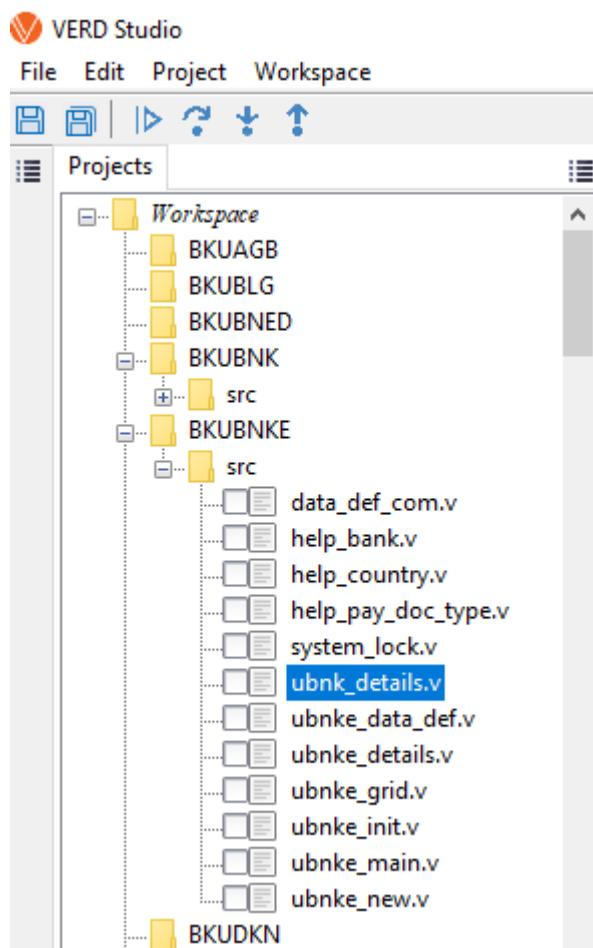
Screenshot 3.43 Menu Path for pasting a file under a folder

- On the next screen enter the new file name of copied file as it would appear under the pasted folder. (*Screenshot 3.44*)



Screenshot 3.44 Paste Dialog for pasting a file under a folder

- Previously copied file with new file name would be pasted to the folder that is selected. After pasting the file, new file with new file name would appear under the folder. (*Screenshot 3.45*)

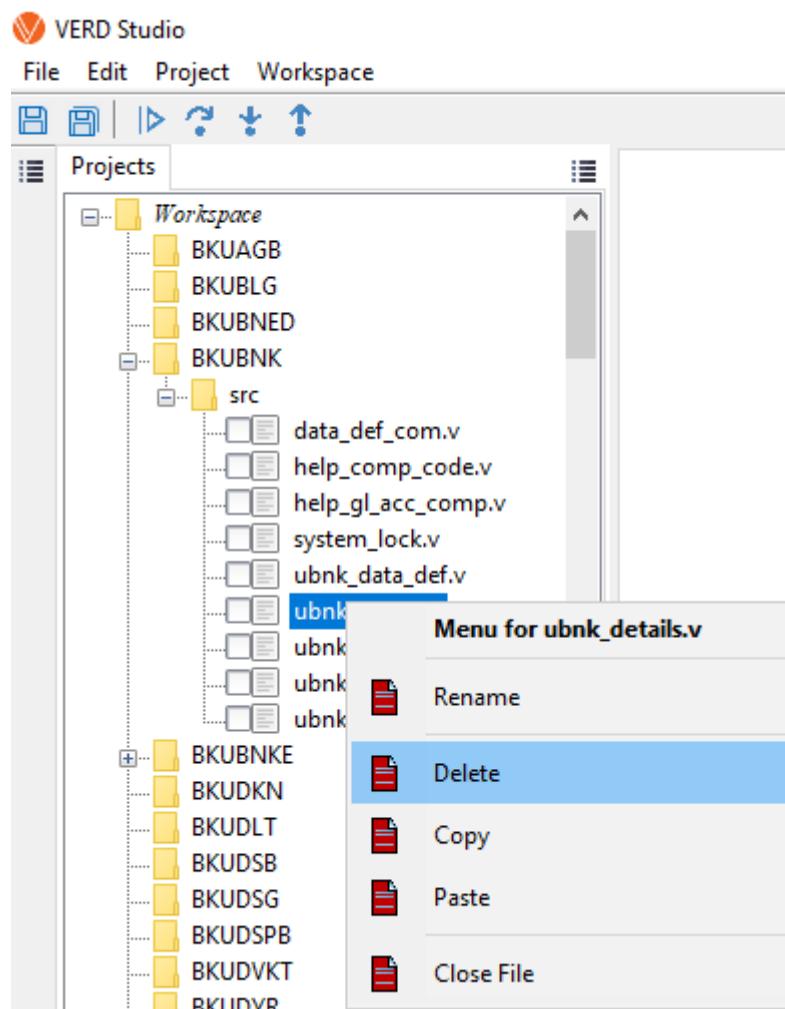


Screenshot 3.45 Main screen after pasting a file under a folder

3.22. Delete a file under a VERD project

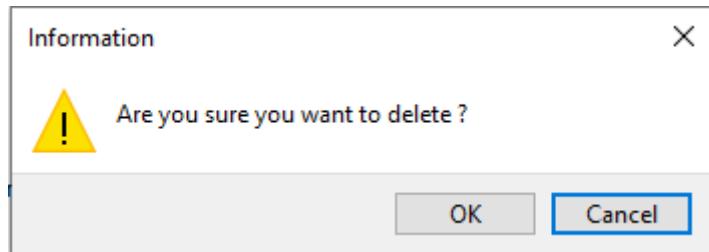
A file under a VERD project or under folders of a VERD project can be deleted in VERD Studio. When the file under a VERD project is deleted, file is also deleted from file system. The steps for deleting a folder under a VERD project are listed below, along with descriptions and screenshots.

- To delete a file under a VERD project or under folders of a VERD project, right-click after selecting a file. In the menu that appears, click on **Delete**.
(Screenshot 3.46)



Screenshot 3.46 Menu Path for deleting a file

- On the next screen click on the **OK** button to delete the file under a VERD project or under folders of a VERD project. The file would also be deleted from file system. (*Screenshot 3.47*)

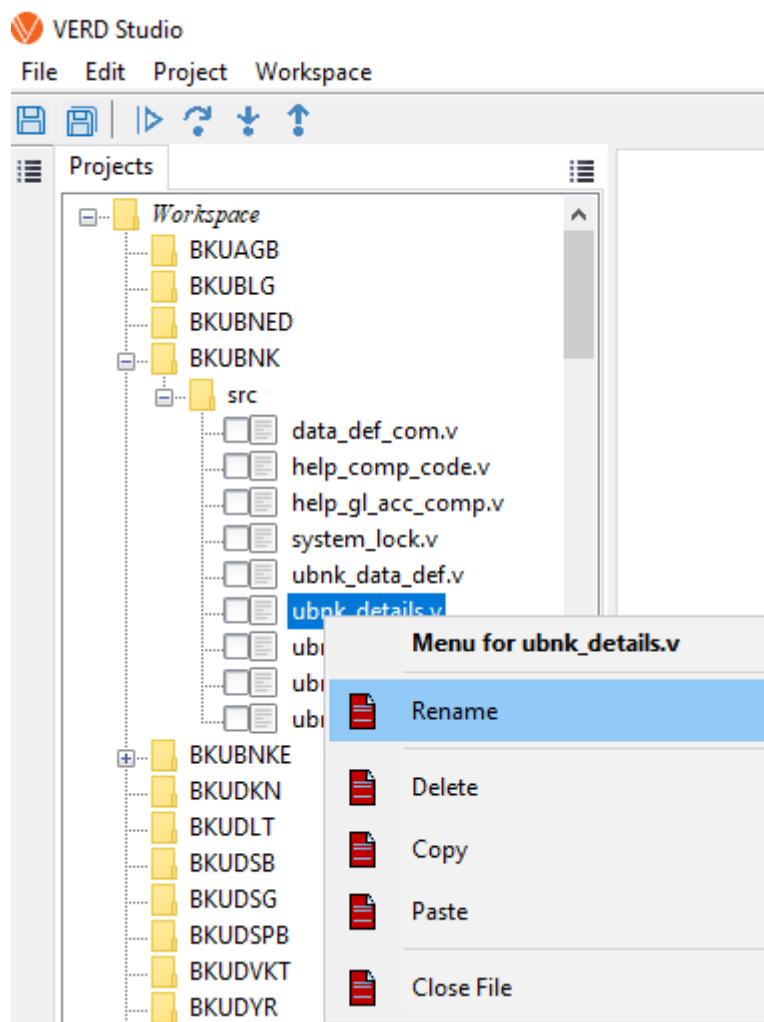


Screenshot 3.47 Confirmation screen for deleting a file

3.23. Rename a file under a VERD project

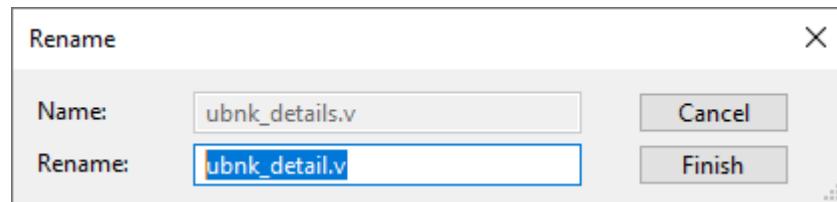
A file under a VERD project or under folders of a VERD project can be renamed in VERD Studio. When a file under a VERD project is renamed, VERD project file is updated according to this change. The steps for renaming a file under a VERD project are listed below, along with descriptions and screenshots.

- To rename a folder under a VERD project or under folders of a VERD project, right-click after selecting a file. In the menu that appears, click on **Rename**.
(Screenshot 3.48)



Screenshot 3.48 Menu path for renaming a file

- On the next screen enter the **Rename** field with new file name and click on **OK** button. (*Screenshot 3.49*)

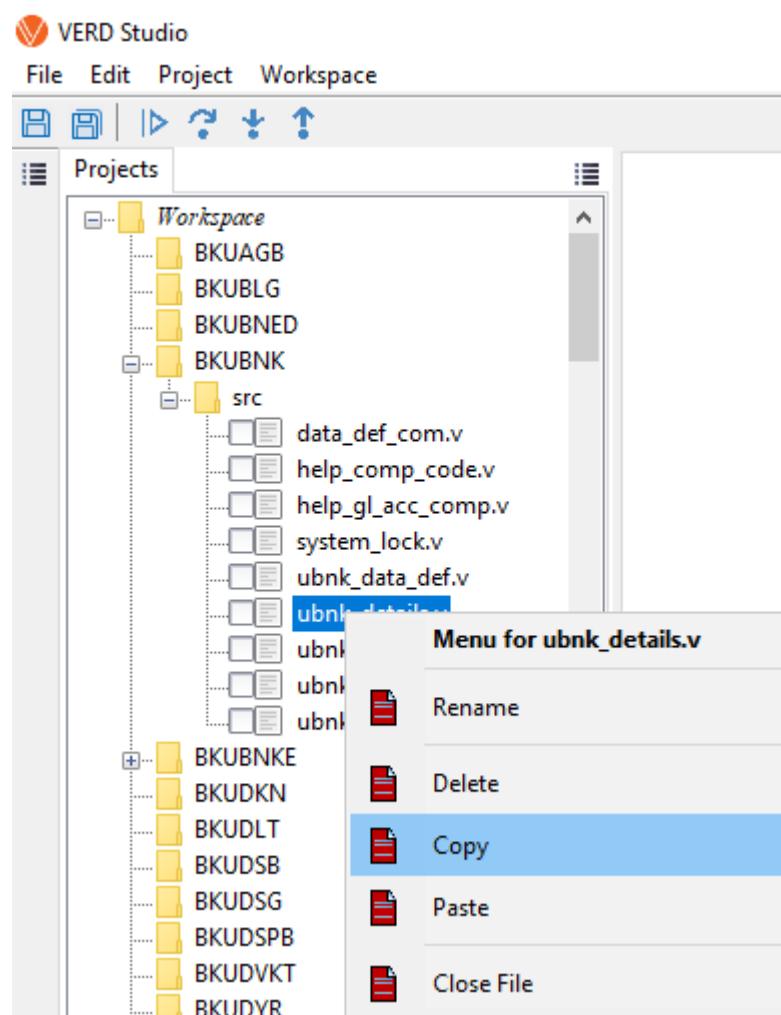


Screenshot 3.49 Rename Dialog for file

3.24. Copy a file under a VERD project

A file under a VERD project or under folders of a VERD project can be copied in VERD Studio to be pasted afterwards. The steps for copying a file under a VERD project are listed below, along with descriptions and screenshots.

- To copy a file under a VERD project or under folders of a VERD project, right-click after selecting a file. In the menu that appears, click on **Copy**. (*Screenshot 3.50*)

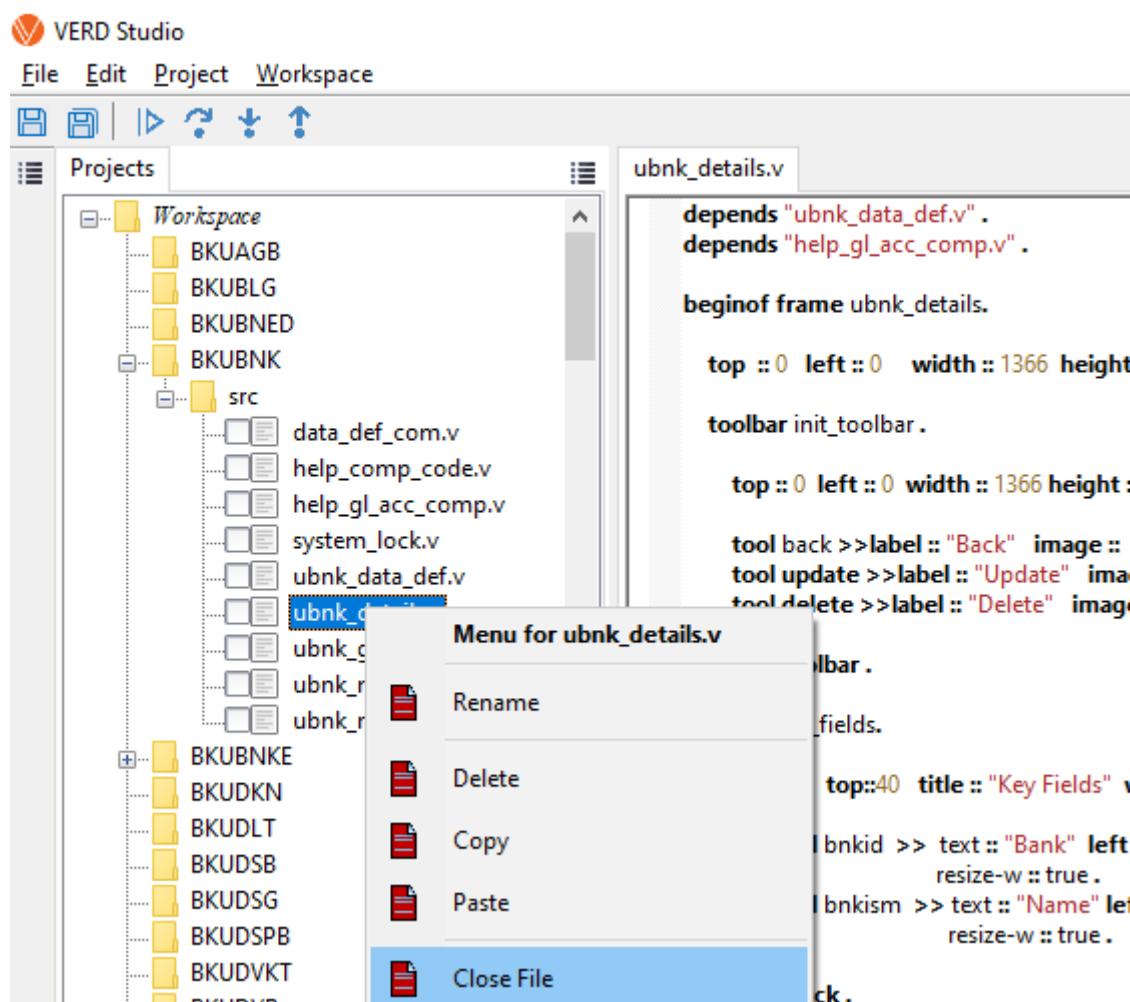


Screenshot 3.50 Menu Path for copying a file

3.25. Close an already opened file under a VERD project

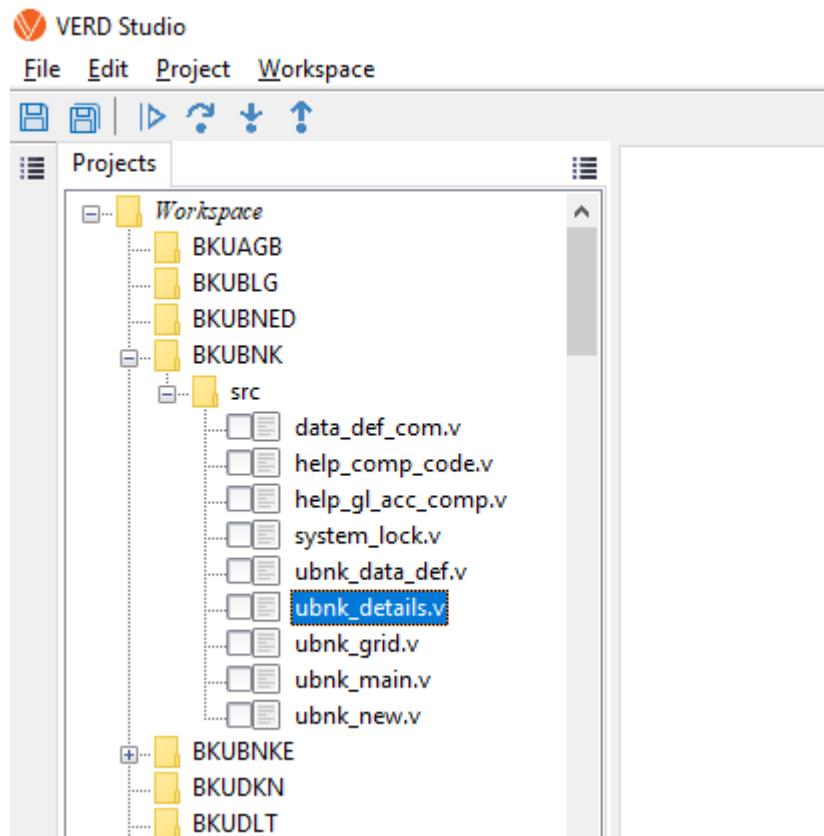
A file under a VERD project or under folders of a VERD project can be closed in VERD Studio if it is already opened. The steps for closing a file under a VERD project are listed below, along with descriptions and screenshots.

- To close a file already opened under a VERD project or under folders of a VERD project, right-click after selecting a file. In the menu that appears, click on **Close File**. (*Screenshot 3.51*)



Screenshot 3.51 Menu Path for closing a file

- If the file is already opened it would be closed and deleted from files tab.
(Screenshot 3.52)

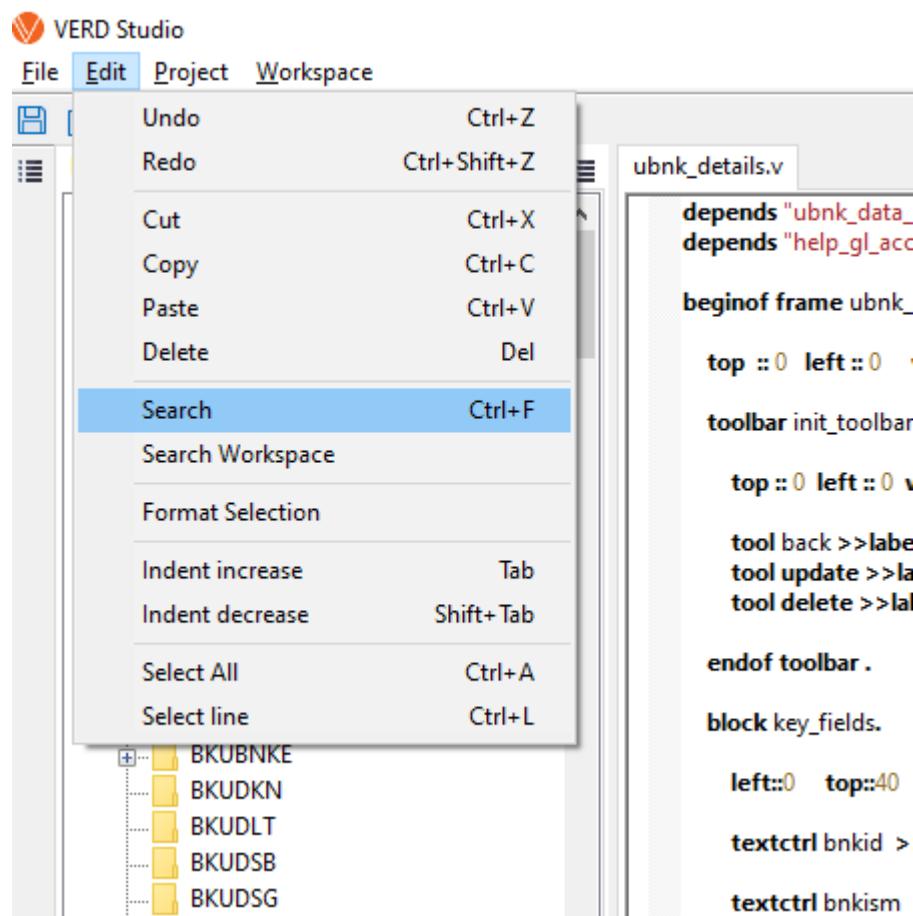


Screenshot 3.52 Main screen after closing a file

3.26. Search and Replace in the current file.

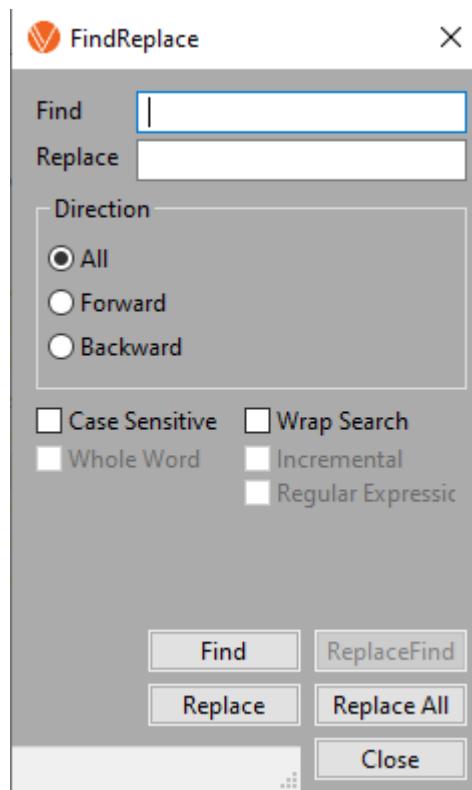
Search and replace in current file being processed is possible in VERD Studio. The steps for search and replace in current file being processed are listed below, along with descriptions and screenshots..

- For find and replace in current file being processed select **Edit->Search** menu path. (*Screenshot 3.53*)



Screenshot 3.53 Menu Path for searching in current file

- On the next screen, the text to be searched is entered, and the search is performed by clicking on the desired options. (*Screenshot 3.54*) Several options are available for the search and replace in current file being processed. The options and their purposes are listed in **Table 3.1**.



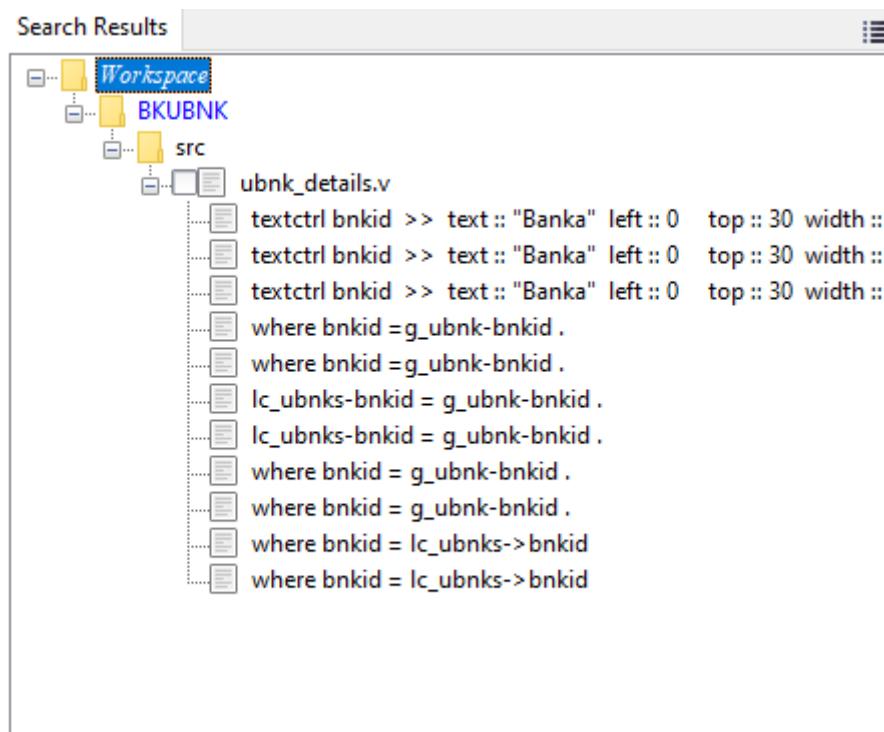
Screenshot 3.54 Search and Replace Dialog for current file

Table 3.1 Options for Search and Replace for current file

Option Name	Type	Purpose
All	Radio Button	This option must be selected when searching the entire document.
Forward	Radio Button	This option should be selected when searching beyond the point where the mouse was last clicked in the document
Backward	Radio Button	This option should be selected when searching prior to the point where the mouse was last clicked in the document
Case Sensitive	Check Box	This option should be clicked if the search needs to be

		case-sensitive. If this option is checked, the words "loop" and "Loop" are treated as different words; if it is not checked, they are treated as the same word
Find	Button	When this option is selected, only a search will be performed for the desired text in the file; no replacement would be made
Replace	Button	When this option is selected, the first match found in the file for the text being searched would be replaced with the text entered in replace field. To change the second match found, the Replace option must be selected again.
Replace All	Button	When this option is selected, all matches found in the file for the text being searched would be replaced with the text entered in replace field.

- After a search or replace operation, the results are listed in a tree structure. When the user clicks on any tree element, the found file and row are opened by VERD Studio. (*Screenshot 3.55*)

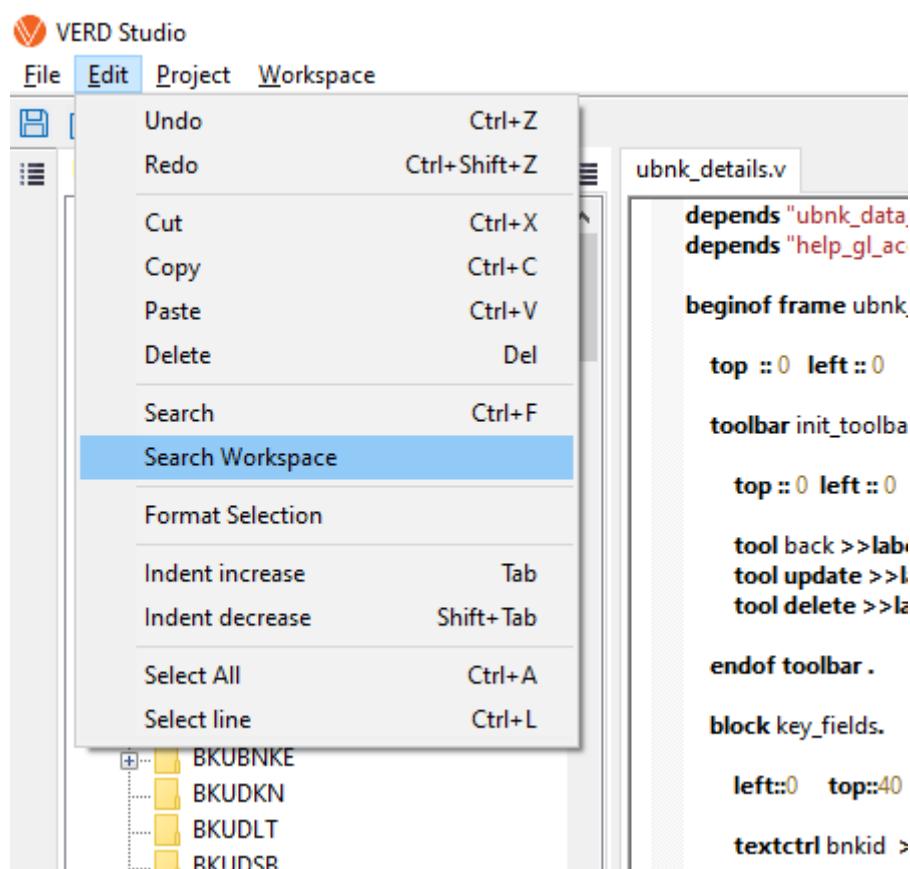


Screenshot 3.55 Listing search results in a tree

3.27. Search and Replace in the Workspace.

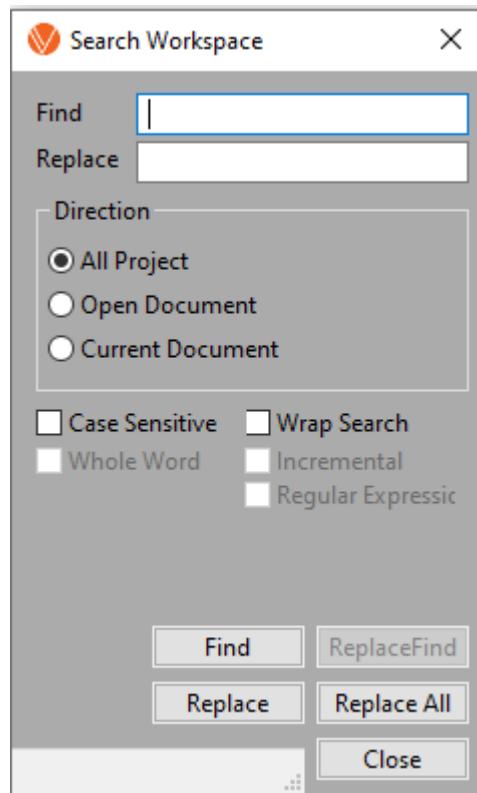
Search and replace in workspace is supported in VERD Studio. In this functionality of VERD Studio, search and replace operations can be performed in all open projects in workspace. The steps for search and replace in workspace are listed below, along with descriptions and screenshots..

- For search and replace in all open projects in workspace select **Edit->Search Workspace** menu path. (*Screenshot 3.56*)



Screenshot 3.56 Menu Path for search and replace in workspace

- On the next screen, the text to be searched is entered, and the search is performed by clicking on the desired options (*Screenshot 3.57*). Several options are available for the search and replace in current file being processed. The options and their purposes are listed in **Table 3.2**.



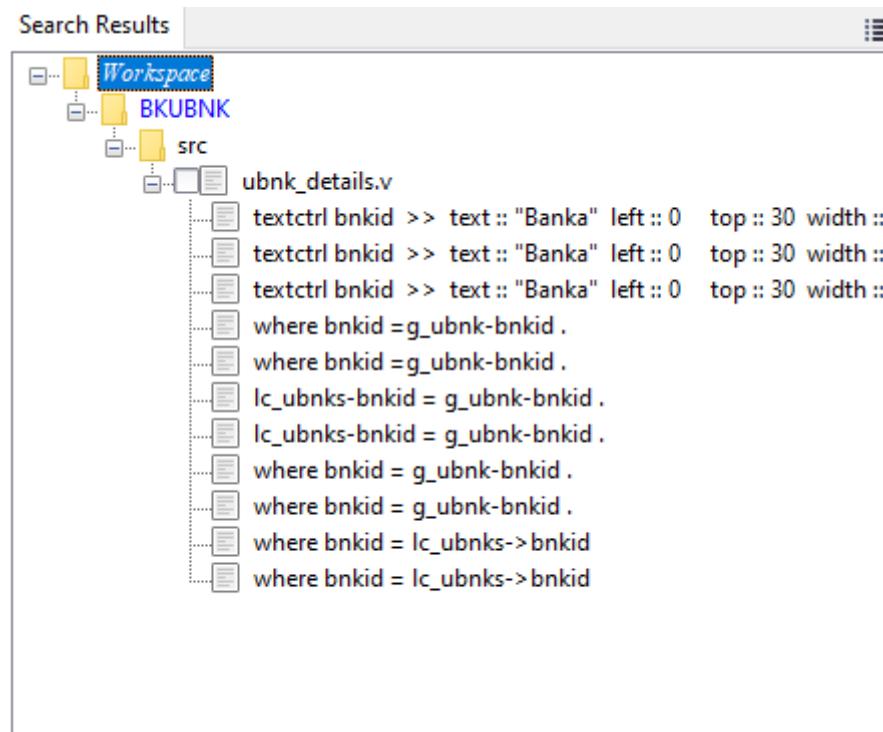
Screenshot 3.57 Search and Replace Dialog for workspace

Table 3.2 Options for Search and Replace for workspace

Option Name	Type	Purpose
All Project	Radio Button	This option must be selected when searching all open projects in workspace.
Open Document	Radio Button	This option should be selected when searching only in open source code files.
Current Document	Radio Button	This option should be selected when searching only in currently processing open source code file.
Case Sensitive	Check Box	This option should be clicked if the search needs to be case-sensitive. If this option is checked, the words "loop" and "Loop" are treated as different words; if it

		is not checked, they are treated as the same word
Find	Button	When this option is selected, only a search will be performed for the desired text in the files; no replacement would be made
Replace	Button	When this option is selected, the first match found in all files for the text being searched would be replaced with the text entered in replace field. To change the second match found, the Replace option must be selected again.
Replace All	Button	When this option is selected, all matches found in all files for the text being searched would be replaced with the text entered in replace field.

- After a search or replace operation, the results are listed in a tree structure. When the user clicks on any tree element, the found file and row are opened by VERD Studio. (*Screenshot 3.58*)

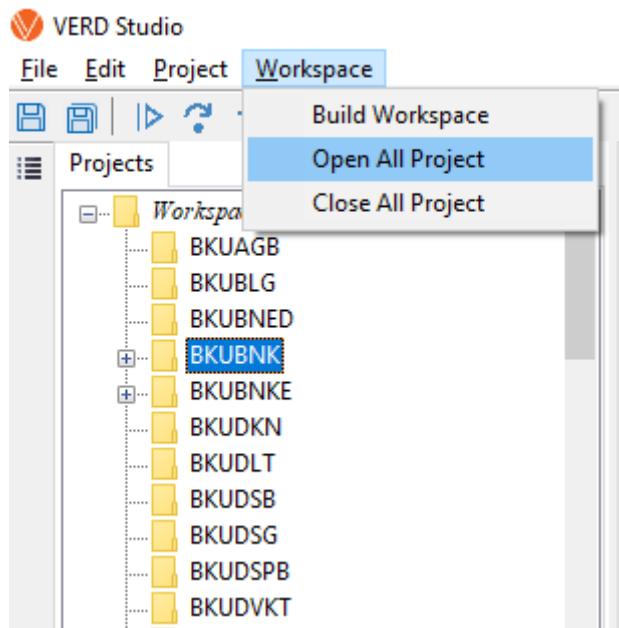


Screenshot 3.58 Listing search results in a tree

3.28. Open all projects in a Workspace

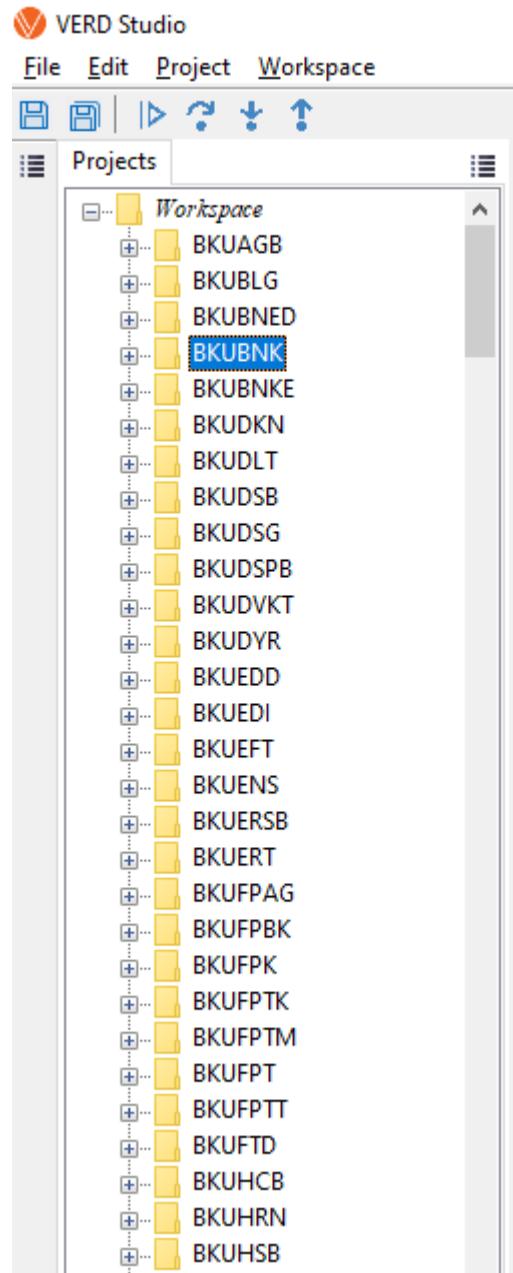
All projects in a workspace can be opened with a single menu command. The steps for opening all projects in a workspace are listed below, along with descriptions and screenshots.

- To open all projects in workspace select **Workspace->Open All Projects** menu path. (*Screenshot 3.59*)



Screenshot 3.59 Menu Path for opening all projects

- All projects are opened in VERD Studio as it can be seen from screenshot below. (*Screenshot 3.60*)

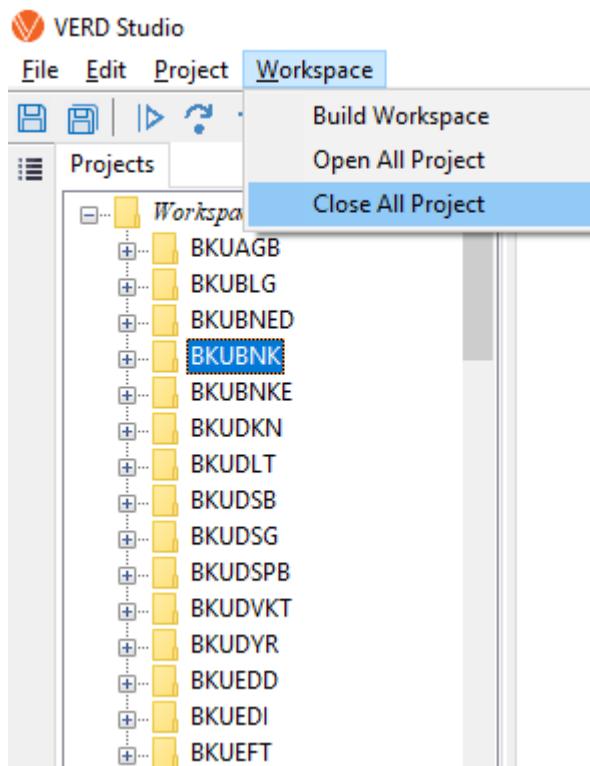


Screenshot 3.60 Main Screen after opening all projects

3.29. Close all projects in a Workspace

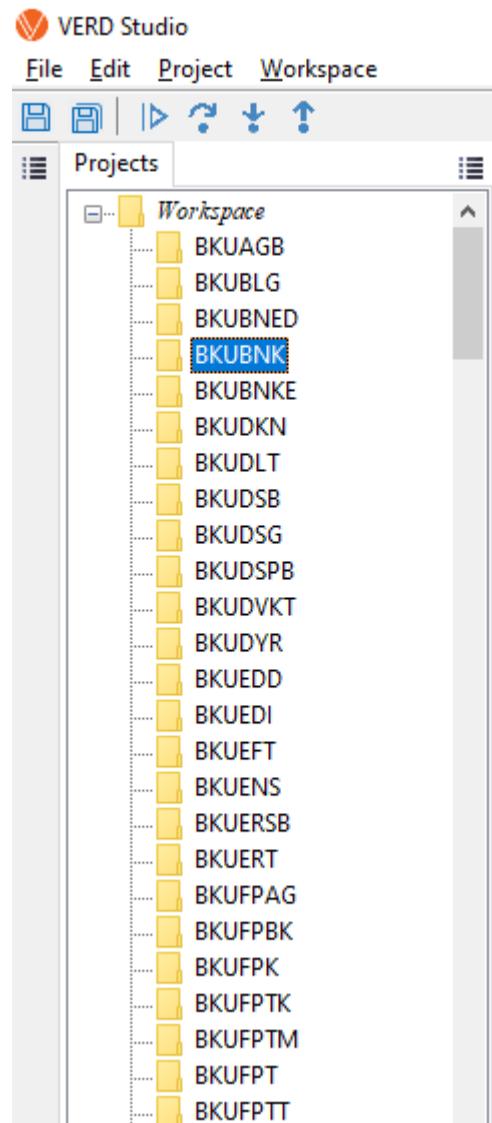
All open projects in a workspace can be closed with a single menu command. The steps for closing all open projects in a workspace are listed below, along with descriptions and screenshots..

- To close all open projects in workspace select **Workspace->Close All Projects** menu path. (*Screenshot 3.61*)



Screenshot 3.61 Menu Path for closing all open projects

- All open projects are closed in VERD Studio as it can be seen from screenshot below. (*Screenshot 3.62*)



Screenshot 3.62 Main Screen after closing all open projects

3.30. Format Selected Text

In VERD Studio, text in source code files can be formatted for better readability. Formatting in VERD Studio includes correcting indentation and formatting text for VERD language constructs such as **select**, **loop** and **read**.

Text formatting in VERD Studio can only be done for the selected text in currently processing source file .The steps for formatting selected text are listed below, along with descriptions and screenshots.

- Select a portion of text in already opened source file. (*Screenshot 3.63*)

```

endof frame .

routine init_ubnk_details .

data lc_ubnks type ubnks_ca .
data lc_ubnks_ref type weak ref to ubnks_ca .

data lc_count type unsigned short .

if g_ubnk-ubnks_tab is initial .

select * from ubnks into
corresponding fields of
    table g_ubnk-ubnks_tab
    where bnid = g_ubnk-bnid .

loop at g_ubnk-ubnks_tab
into lc_ubnks_ref .

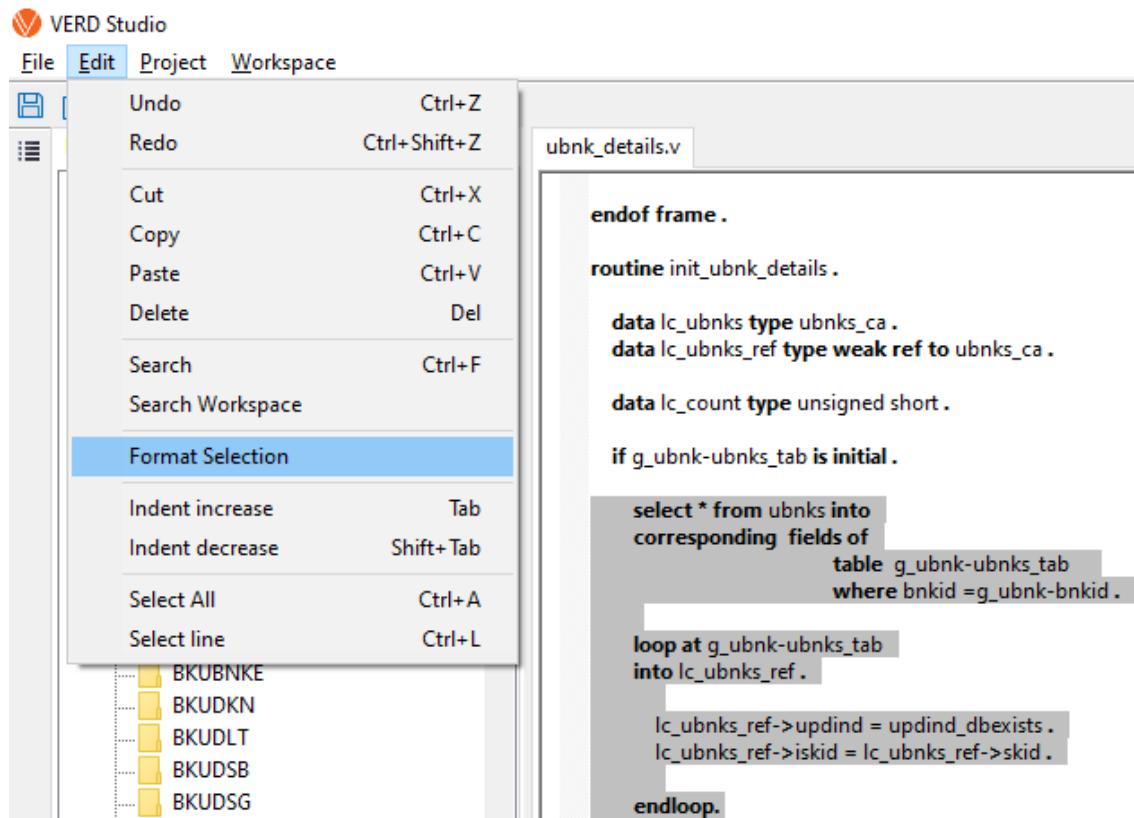
lc_ubnks_ref->updind = updind_dbexists .
lc_ubnks_ref->iskid = lc_ubnks_ref->skid .

endloop.
...

```

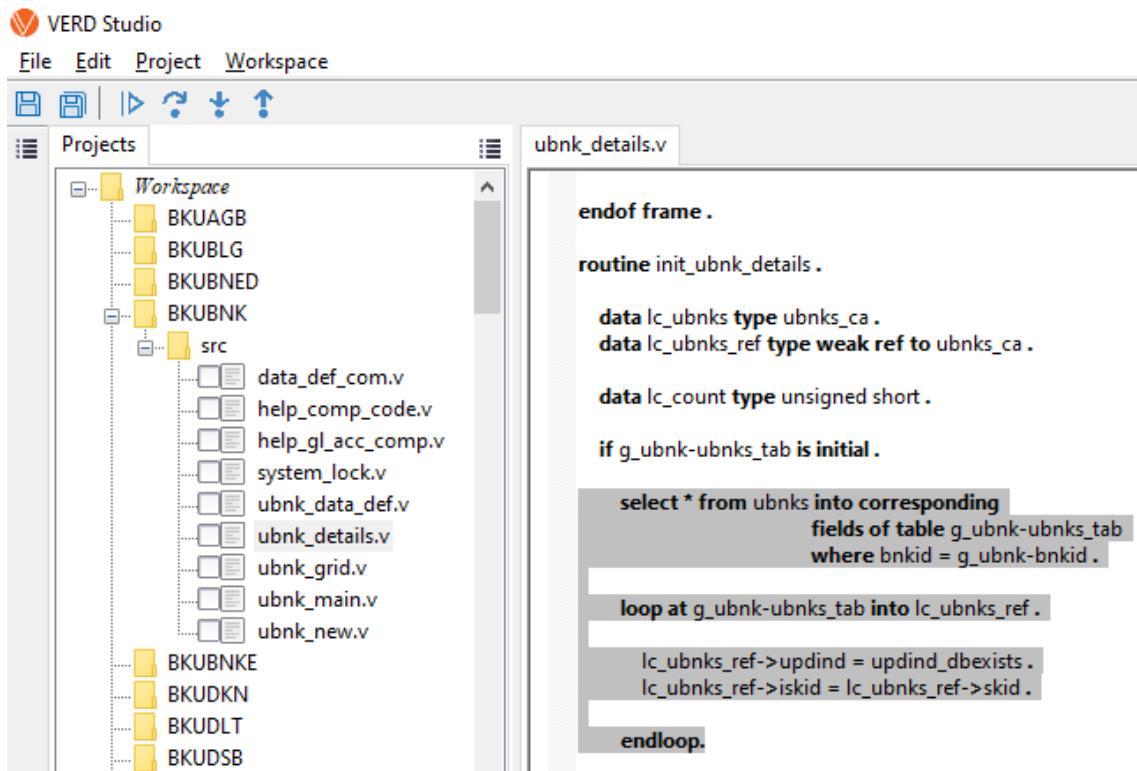
Screenshot 3.63 Selecting a portion of text to be formatted

- To format selected text, select **Edit->Format Selection** menu path. (*Screenshot 3.64*)



Screenshot 3.64 Menu Path for formatting selected text

- Selected text has been formatted by VERD Studio as it can be seen from screenshot below. (*Screenshot 3.65*)



The screenshot shows the VERD Studio interface. The top menu bar includes File, Edit, Project, and Workspace. Below the menu is a toolbar with icons for file operations. The left pane is a 'Projects' browser showing a hierarchy of workspace projects: Workspace, BKUAGB, BKUBLG, BKUBNED, BKUBNK, and several sub-folders like src containing various Verilog files such as data_def_com.v, help_comp_code.v, etc. The right pane displays the content of the file 'ubnk_details.v'. The code is a snippet of SystemVerilog or similar language:

```

endof frame .

routine init_ubnk_details .

data lc_ubnks type ubnks_ca .
data lc_ubnks_ref type weak ref to ubnks_ca .

data lc_count type unsigned short .

if g_ubnk-ubnks_tab is initial .

    select * from ubnks into corresponding
        fields of table g_ubnk-ubnks_tab
        where bnkid = g_ubnk-bnkid .

    loop at g_ubnk-ubnks_tab into lc_ubnks_ref .
        lc_ubnks_ref->updind = updind_dbexists .
        lc_ubnks_ref->iskid = lc_ubnks_ref->skid .
    endloop.

```

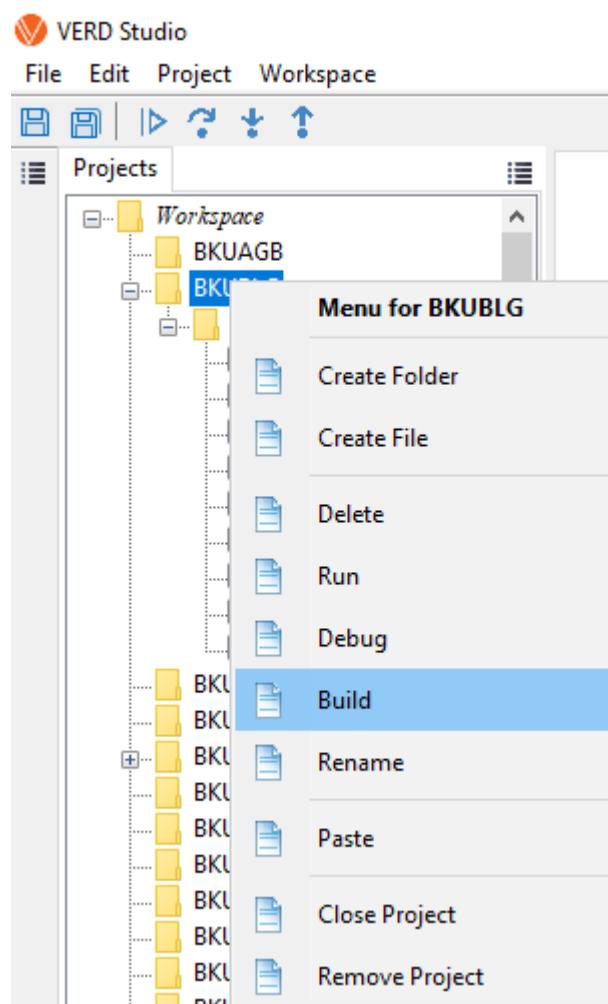
Screenshot 3.65 Final text after being formatted

3.31. Build VERD Project

The project can be compiled within VERD Studio. During the project compilation process, VERD Studio sends information about the project to be compiled to the compiler, and the compiler compiles the project based on this information. Any errors that may occur as a result of the compilation are returned by the compiler to VERD Studio.

The steps in the project build process are listed below, along with descriptions and screenshots.

- To build a project, right-click after selecting a project. In the menu that appears, click on **Build**. (*Screenshot 3.66*)



Screenshot 3.66 Menu Path for building a project

- After the project is compiled, the compiler output is shown to the user.
(Screenshot 3.67)

File	LineNo	Type	Message
\src\system_lock.v	15	⚠	Possible loss of data since SY-USRNAME is bigger than size of LC_SKLT-KLID
\src\system_lock.v	60	⚠	Possible loss of data since SY-USRNAME is bigger than size of LC_SKLT-KLID
\src\system_lock.v	88	⚠	Possible loss of data since SY-USRNAME is bigger than size of LC_SKLT-KLID
\src\system_lock.v	114	⚠	Possible loss of data since SY-USRNAME is bigger than size of LC_SKLT-KLID
\src\ublg_details.v	31	✗	Parameter LC_UBLGX is not defined
\src\ublg_details.v	35	✗	Parameter G_UBLGX is not defined

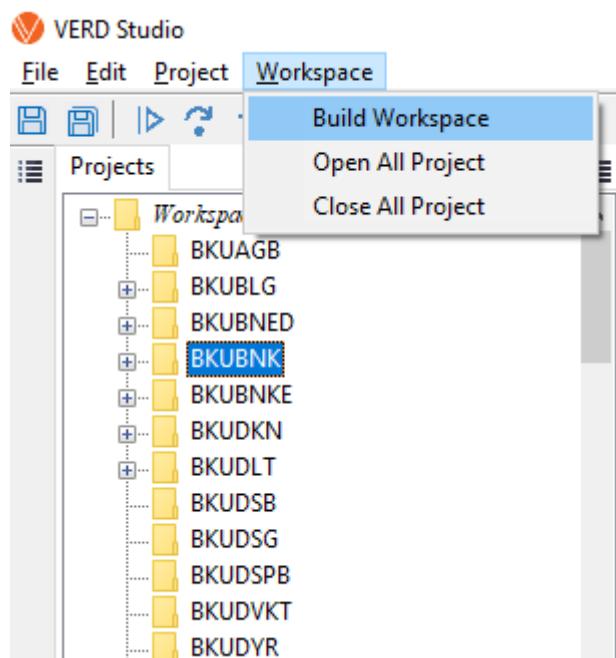
Build Failed :2 errors, 4 warnings

Screenshot 3.67 Build Log after building a project

3.32. Build Workspace

All open projects in a workspace can be built with a single menu command. If any build errors occur in any project during compilation, VERD Studio stops processing remaining open projects and displays the build errors from the project in question. The steps for building all open projects in a workspace are listed below, along with descriptions and screenshots.

- To build all open projects in workspace select **Workspace->Build Workspace** menu path. (*Screenshot 3.68*)



Screenshot 3.68 Menu Path for building all open projects

- All open projects are built in VERD Studio by using VERD compiler. If build errors occur in any project, VERD Studio displays these error messages and stops processing remaining open projects as it can be seen from screenshot below. (**Screenshot 3.69**)

File	LineNo	Type	Message
\src\system_lock.v	15	⚠	Possible loss of data since SY-USRNAME is bigger than size of LC_SKLT-KLID
\src\system_lock.v	60	⚠	Possible loss of data since SY-USRNAME is bigger than size of LC_SKLT-KLID
\src\system_lock.v	88	⚠	Possible loss of data since SY-USRNAME is bigger than size of LC_SKLT-KLID
\src\system_lock.v	114	⚠	Possible loss of data since SY-USRNAME is bigger than size of LC_SKLT-KLID
\src\ubnk_grid.v	59	✗	Parameter G_UBNKX_TAB is not defined
\src\ubnk_grid.v	61	✗	Parameter LC_UBNKX is not defined
\src\ubnk_grid.v	72	✗	Parameter LC_UBNKX is not defined
\src\ubnk_grid.v	73	✗	Parameter LC_UBNKX is not defined
\src\ubnk_grid.v	74	✗	Parameter LC_UBNKX is not defined
Build Failed :5 errors, 4 warnings			

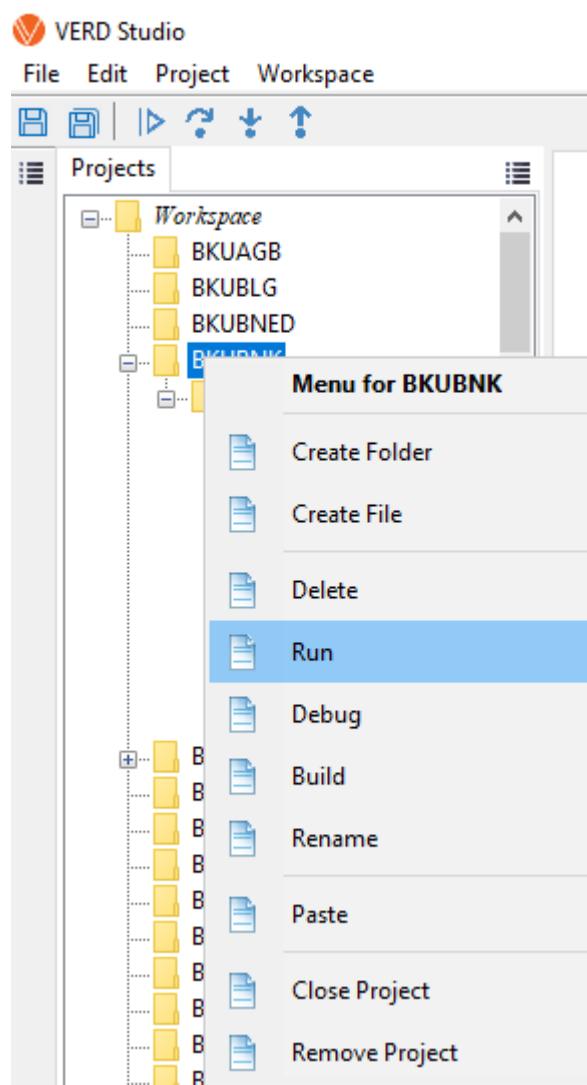
Screenshot 3.69 Build Log after building all open projects

3.33. Running VERD Project

The project can be run within VERD Studio. During the project compilation process, VERD Studio sends information about the outputs of compiled project to the runtime, and the runtime runs the project based on this information.

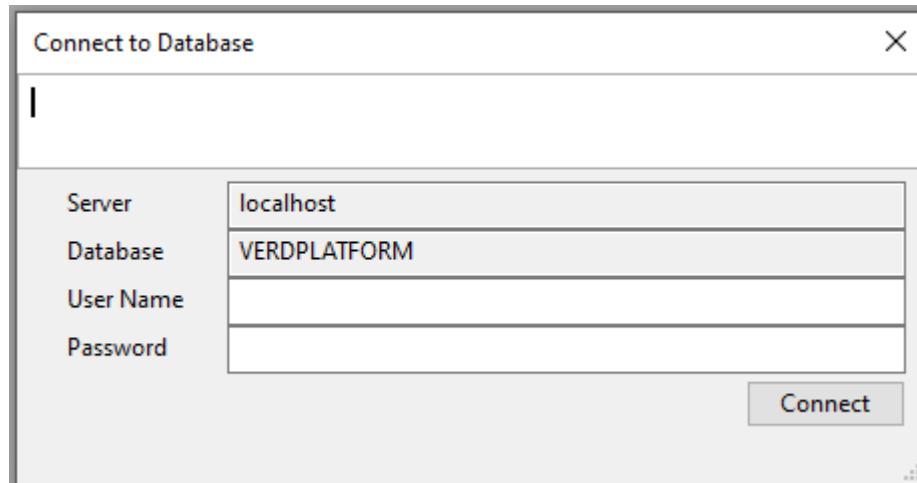
The steps in running the project are listed below, along with descriptions and screenshots.

- To build a project, right-click after selecting a project. In the menu that appears, click on **Run**. (*Screenshot 3.70*)



Screenshot 3.70 Menu Path for running a project

- When the project is run, the initial screen of program is displayed to the user.
(Screenshot 3.71)



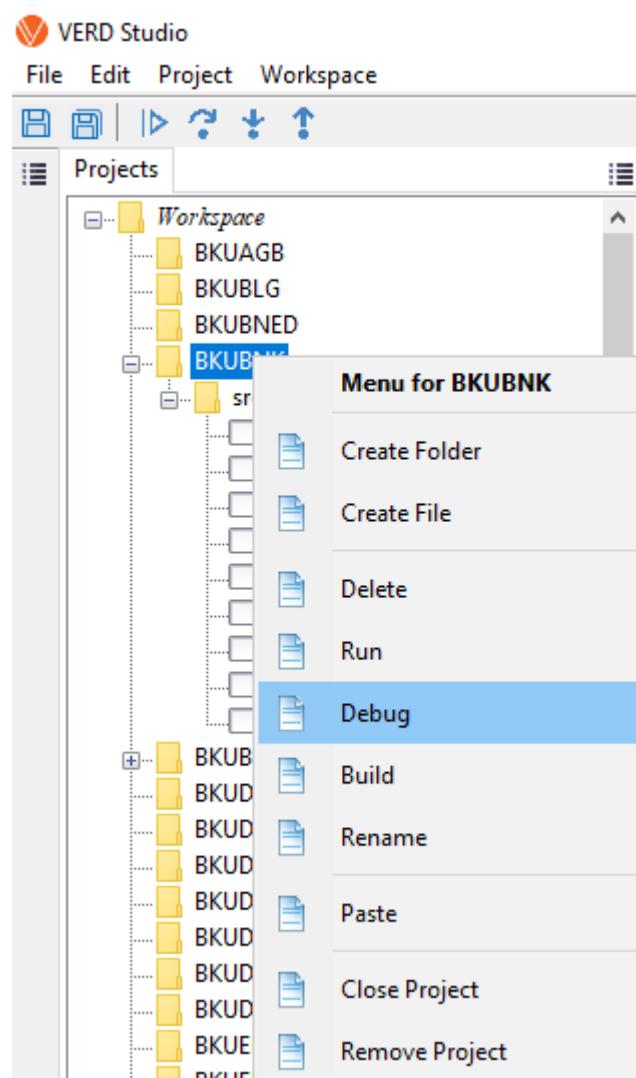
Screenshot 3.71 Initial screen of program that is run

3.34. Debug VERD Project

Errors that may occur in the project can be debugged within VERD Studio using the VERD debugger. During project debugging, VERD Studio sends information about the project to be debugged to the VERD debugger, which then runs the project based on this information and waits for user requests.

The steps involved in debugging the VERD project are listed below, along with descriptions and screenshots.

- To debug a project, right-click after selecting a project. In the menu that appears, click on **Debug**. (*Screenshot 3.72*)



Screenshot 3.72 Menu Path for debugging project

- When the debugger runs and stops at a breakpoint, the response from the debugger is processed and displayed to the user. (*Screenshot 3.73*)

The screenshot shows the VERD Studio interface. The top menu bar includes File, Edit, Project, and Workspace. The left sidebar displays a project tree under 'Workspace' with several subfolders like BKUAGB, BKUBLG, BKUBND, BKUBNK, etc., and a 'src' folder containing various source files such as data_def_com.v, help_comp_code.v, help_gl_acc_comp.v, system_lock.v, ubnk_data_def.v, ubnk_details.v, ubnk_grid.v, ubnk_main.v, and ubnk_new.v. The right pane shows the code editor for 'ubnk_grid.v'. The code contains several lines of Verilog-like pseudocode, including imports, conditionals, loops, and table operations. A callout box highlights a specific line: 'select * from ubnk into corresponding fields of table g_ubnk_tab .'. Below the code editor is a 'Call Stack' table:

Function	Path	Line no
init_ubnk_grid	\src\ubnk_grid.v	58
main	\src\ubnk_main.v	6

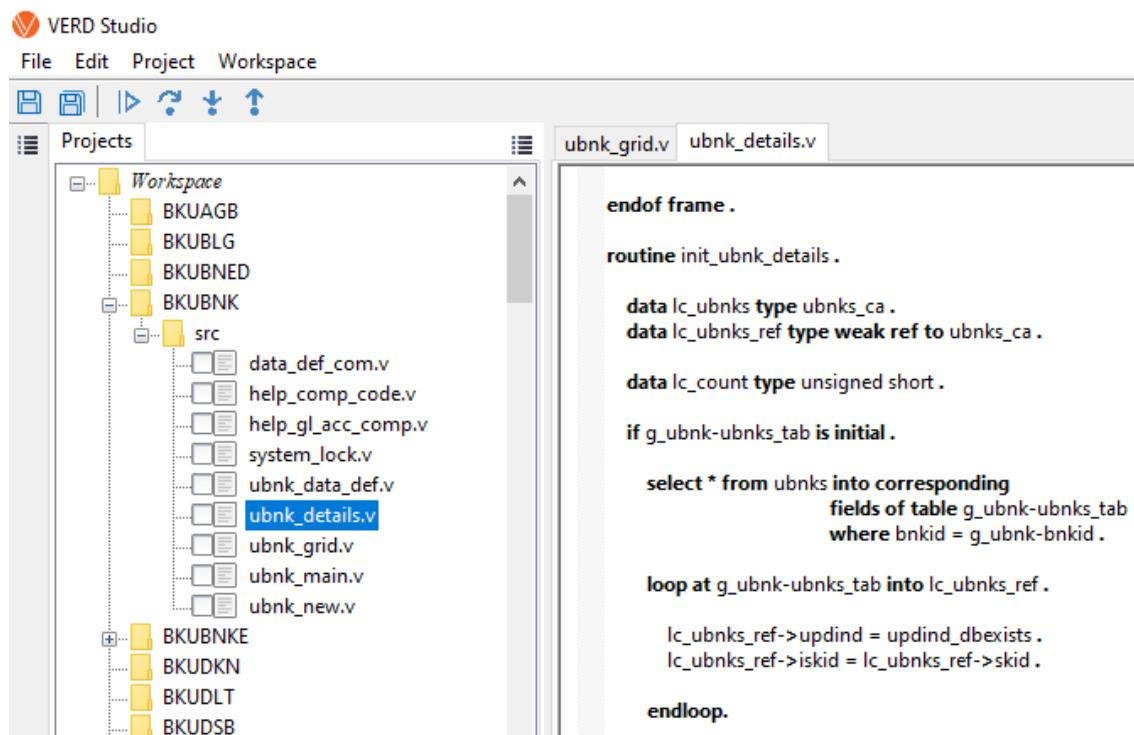
Screenshot 3.73 Main screen after debugger stops at a breakpoint

The VERD Studio supports some of user actions for debugging VERD programs. List of user actions that VERD Studio supports are listed below.

3.34.1. Add Breakpoint

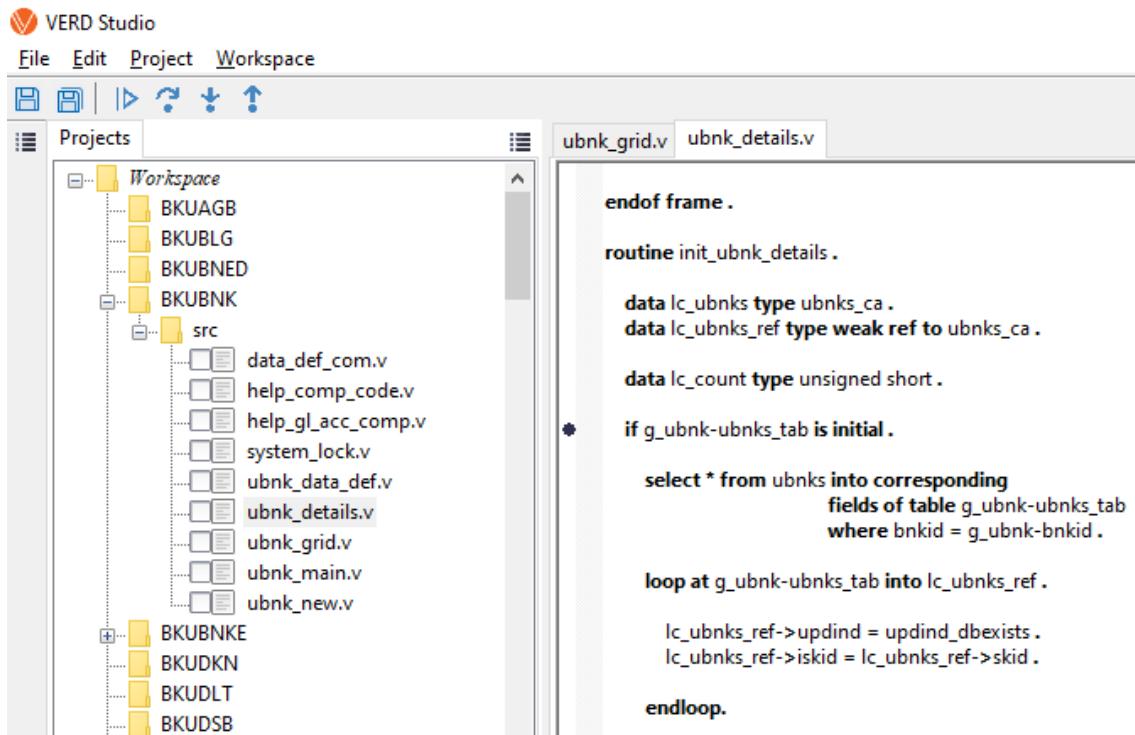
User can add breakpoint to a line in any of the source files within VERD Studio. When a user adds a breakpoint to a line in any of the source files, VERD Studio sends a message to the debugger for that breakpoint. Steps for adding a breakpoint to a source file are listed below.

- Open a source file by double click on it. (*Screenshot 3.74*)



Screenshot 3.74 Opening a file to add a breakpoint

- Double click on the left of the line that you want to put breakpoint. Breakpoint is set and displayed with a blue circle. (*Screenshot 3.75*)



The screenshot shows the VERD Studio interface. The top menu bar includes File, Edit, Project, and Workspace. The toolbar has icons for file operations like Open, Save, and Run. The left pane is a 'Projects' view showing a 'Workspace' folder containing several subfolders (BKUAGB, BKUBLG, BKUBNED, BKUBNK, BKUBNKE, BKUDKN, BKUDLT, BKUDSB) and a 'src' folder containing various Verilog files (data_def_com.v, help_comp_code.v, help_gl_acc_comp.v, system_lock.v, ubnk_data_def.v, ubnk_details.v, ubnk_grid.v, ubnk_main.v, ubnk_new.v). The right pane displays the content of the 'ubnk_grid.v' file. The code is a VHDL-like script with comments and pragmas. A blue circle, indicating a breakpoint, is visible on the line starting with 'select * from ubnks into corresponding'. The window title bar shows 'ubnk_grid.v' and 'ubnk_details.v'.

```

endof frame .

routine init_ubnk_details .

data lc_ubnks type ubnks_ca .
data lc_ubnks_ref type weak ref to ubnks_ca .

data lc_count type unsigned short .

if g_ubnk-ubnks_tab is initial .

select * from ubnks into corresponding
    fields of table g_ubnk-ubnks_tab
    where bnid = g_ubnk-bnid .

loop at g_ubnk-ubnks_tab into lc_ubnks_ref .

lc_ubnks_ref->updind = updind_dbexists .
lc_ubnks_ref->iskid = lc_ubnks_ref->skid .

endloop.

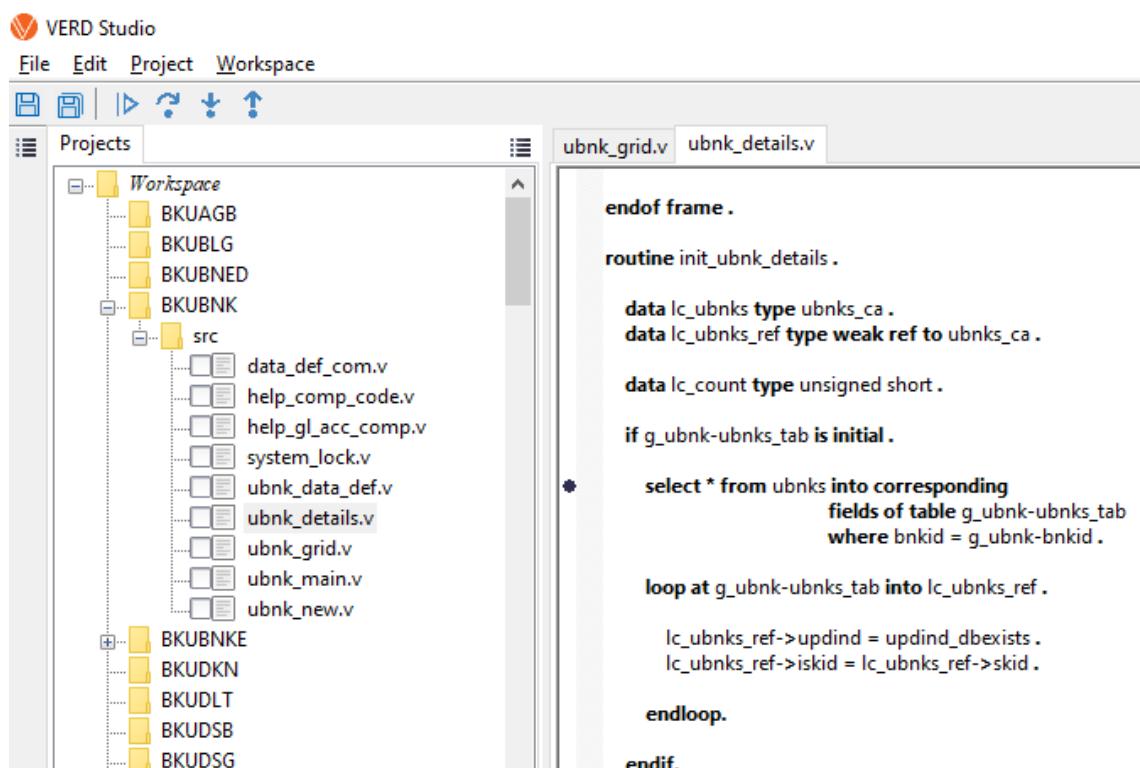
```

Screenshot 3.75 Breakpoint is displayed with a blue circle.

3.34.2. Remove Breakpoint

User can remove a breakpoint that exists in a line of source file within VERD Studio. If a user deletes a breakpoint on a line in any of the source files, VERD Studio sends a message to the debugger for that deleted breakpoint. Steps for deleting a breakpoint from a source file are listed below.

- Open a source file by double click on it. (*Screenshot 3.76*)



The screenshot shows the VERD Studio interface. The top menu bar includes File, Edit, Project, and Workspace. The left sidebar displays the 'Projects' tree, which contains a 'Workspace' folder with several subfolders like BKUAGB, BKUBLG, BKUBNED, BKUBNK, and BKUBNKE, along with a 'src' folder containing various Verilog files such as data_def_com.v, help_comp_code.v, help_gl_acc_comp.v, system_lock.v, ubnk_data_def.v, ubnk_details.v, ubnk_grid.v, ubnk_main.v, and ubnk_new.v. The right pane shows the content of the 'ubnk_grid.v' file. A red asterisk (*) is placed before the line 'select * from ubnks into corresponding', indicating a breakpoint has been set or is being removed.

```

endof frame .

routine init_ubnk_details .

data lc_ubnks type ubnks_ca .
data lc_ubnks_ref type weak ref to ubnks_ca .

data lc_count type unsigned short .

if g_ubnk-ubnks_tab is initial .

    select * from ubnks into corresponding
        fields of table g_ubnk-ubnks_tab
        where bnkid = g_ubnk-bnkid .

    loop at g_ubnk-ubnks_tab into lc_ubnks_ref .

        lc_ubnks_ref->updind = updind_dbexists .
        lc_ubnks_ref->iskid = lc_ubnks_ref->skid .

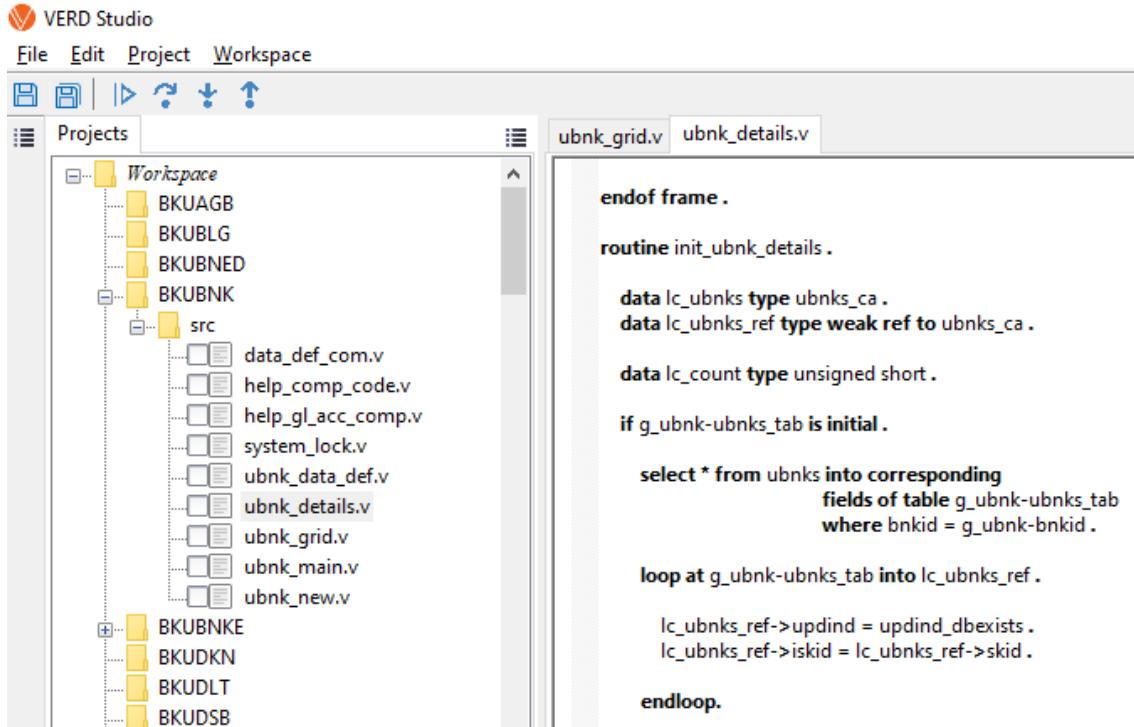
    endloop .

endif.

```

Screenshot 3.76 Opening a file to remove a breakpoint

- Double click on the breakpoint that you want to remove. Breakpoint is already displayed with a blue circle in left margin. Breakpoint is removed and blue circle in left margin is removed from line. (*Screenshot 3.77*)



```

endof frame .

routine init_ubnk_details .

data lc_ubnks type ubnks_ca .
data lc_ubnks_ref type weak ref to ubnks_ca .

data lc_count type unsigned short .

if g_ubnk-ubnks_tab is initial .

select * from ubnks into corresponding
fields of table g_ubnk-ubnks_tab
where bnkid = g_ubnk-bnkid .

loop at g_ubnk-ubnks_tab into lc_ubnks_ref .

lc_ubnks_ref->updind = updind_dbexists .
lc_ubnks_ref->iskid = lc_ubnks_ref->skid .

endloop.

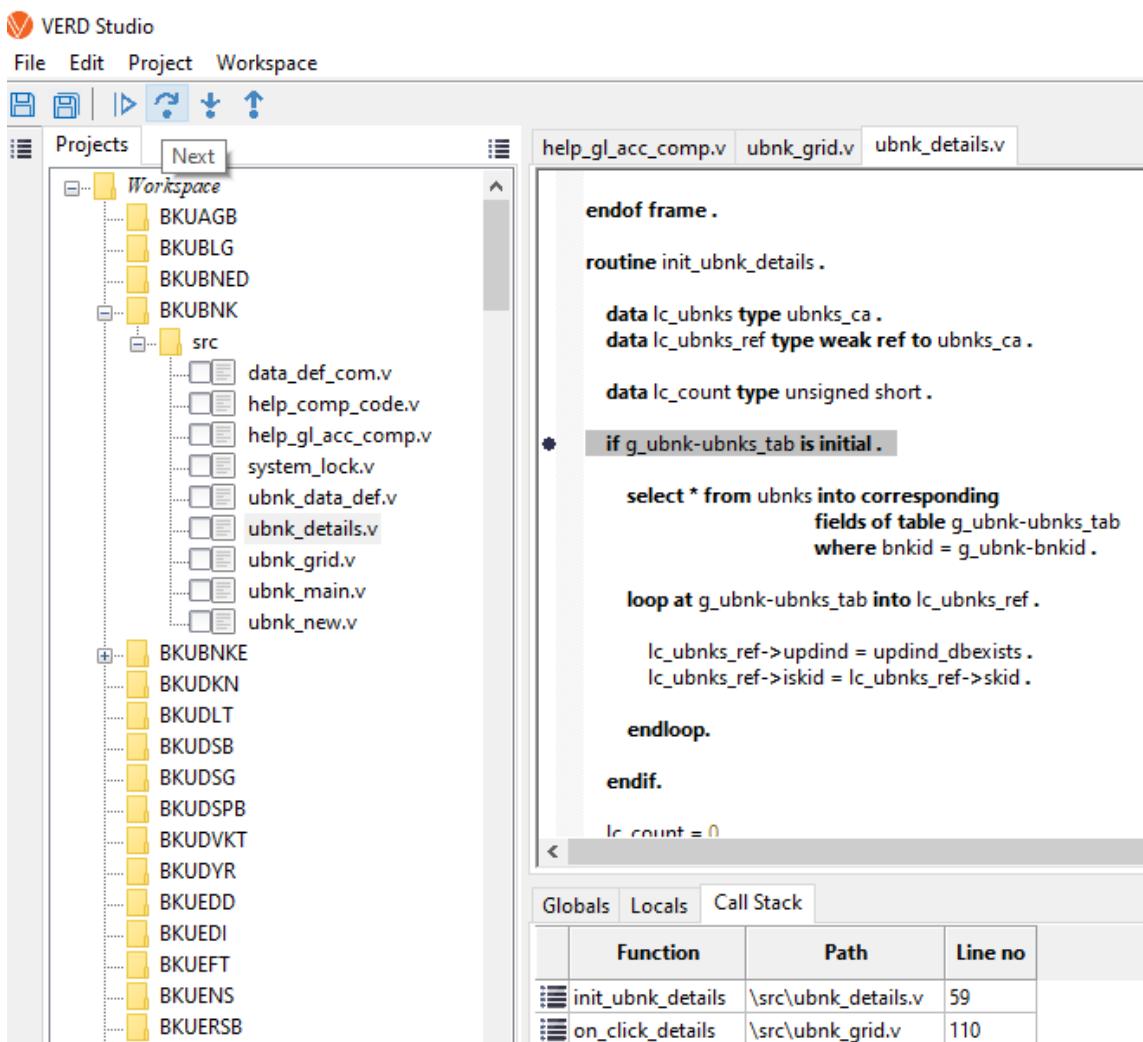
```

Screenshot 3.77 Blue circle for breakpoint is removed from line.

3.34.3. Step Over to Next Instruction

When the debugger reaches a breakpoint, it stops the program flow and waits for requests from the user. One such request is a “**step over to next instruction**” request. Step over to next instruction request allows the debugger to perform the operation it is currently halted on and then stop the program again on the next instruction. Steps for stepping over to next instruction are listed below.

- Click on “**Next**” icon  in toolbar when debugger stops the program at a breakpoint. (*Screenshot 3.78*)



```

VerD Studio
File Edit Project Workspace
Projects | Next
Workspace
  BKUAGB
  BKUBLG
  BKUBNED
  BKUBNK
    src
      data_def_com.v
      help_comp_code.v
      help_gl_acc_comp.v
      system_lock.v
      ubnk_data_def.v
      ubnk_details.v
      ubnk_grid.v
      ubnk_main.v
      ubnk_new.v
  BKUBNKE
  BKUDKN
  BKUDLT
  BKUDSB
  BKUDSG
  BKUDSPB
  BKUDVKT
  BKUDYR
  BKUEDD
  BKUEDI
  BKUEFT
  BKUENS
  BKUERSB

help_gl_acc_comp.v  ubnk_grid.v  ubnk_details.v

endof frame .

routine init_ubnk_details .

  data lc_ubnks type ubnks_ca .
  data lc_ubnks_ref type weak ref to ubnks_ca .

  data lc_count type unsigned short .

  if g_ubnk-ubnks_tab is initial .

    select * from ubnks into corresponding
      fields of table g_ubnk-ubnks_tab
      where bnkid = g_ubnk-bnkid .

    loop at g_ubnk-ubnks_tab into lc_ubnks_ref .

      lc_ubnks_ref->updind = updind_dbexists .
      lc_ubnks_ref->iskid = lc_ubnks_ref->skid .

    endloop .

  endif .

  lc_count = 0

Globals Locals Call Stack
Function Path Line no
init_ubnk_details \src\ubnk_details.v 59
on_click_details \src\ubnk_grid.v 110

```

Screenshot 3.78 Click “Next” when debugger stopped at breakpoint.

- The debugger iterates to next instruction and stop at that instruction. Source code is updated with new line where debugger stopped. (*Screenshot 3.79*)

VERD Studio

File Edit Project Workspace

Projects

help_gl_acc_comp.v ubnk_grid.v ubnk_details.v

```

endof frame .

routine init_ubnk_details .

data lc_ubnks type ubnks_ca .
data lc_ubnks_ref type weak ref to ubnks_ca .

data lc_count type unsigned short .

if g_ubnk-ubnks_tab is initial .

select * from ubnks into corresponding
    fields of table g_ubnk-ubnks_tab
    where bnkid = g_ubnk-bnkid .

loop at g_ubnk-ubnks_tab into lc_ubnks_ref .

    lc_ubnks_ref->updind = updind_dbexists .
    lc_ubnks_ref->iskid = lc_ubnks_ref->skid .

    endloop.

endif.

lc_count = 0

```

Globals Locals Call Stack

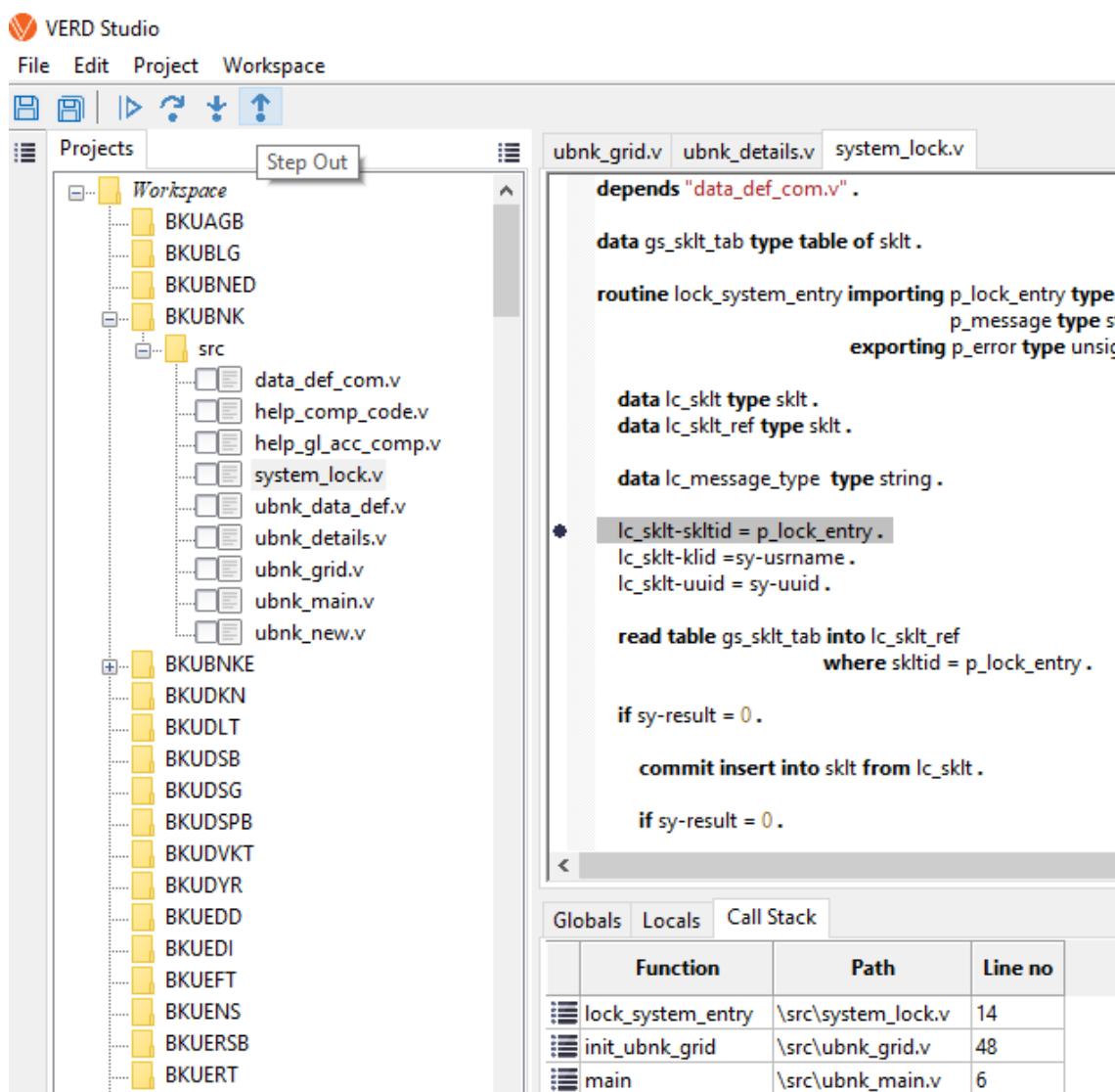
	Function	Path	Line no
	init_ubnk_details	\src\ubnk_details.v	61
	on_click_details	\src\ubnk_grid.v	110

Screenshot 3.79 Source code is updated after “Next” instruction

3.34.4. Step Out from Current Function

When the debugger reaches a breakpoint, it stops the program's execution and waits for user requests. One such request is “**step out from current function**” request. This request allows the debugger to complete all operations in the function that it is currently in, and to stop again in the next instruction in the calling function Steps for stepping out from current function are listed below.

- Click on “**Step Out**” icon  in toolbar when debugger stops the program at a breakpoint. (*Screenshot 3.80*)



```

VERD Studio
File Edit Project Workspace
Projects Step Out
Workspace
  BKUAGB
  BKUBLG
  BKUBNED
  BKUBNK
    src
      data_def_com.v
      help_comp_code.v
      help_gl_acc_comp.v
      system_lock.v
      ubnk_data_def.v
      ubnk_details.v
      ubnk_grid.v
      ubnk_main.v
      ubnk_new.v
  BKUBNKE
  BKUDKN
  BKUDLT
  BKUDSB
  BKUDSG
  BKUDSPB
  BKUDVKT
  BKUDYR
  BKUEDD
  BKUEDI
  BKUEFT
  BKUENS
  BKUERSB
  BKUERT
ubnk_grid.v ubnk_details.v system_lock.v
depends "data_def_com.v".

data gs_sklt_tab type table of sklt .

routine lock_system_entry importing p_lock_entry type
  p_message type st
  exporting p_error type unsig

data lc_sklt type sklt .
data lc_sklt_ref type sklt .

data lc_message_type type string .

lc_sklt-skltid = p_lock_entry .
lc_sklt-klid = sy-username .
lc_sklt-uuid = sy-uuid .

read table gs_sklt_tab into lc_sklt_ref
  where skltid = p_lock_entry .

if sy-result = 0 .

  commit insert into sklt from lc_sklt .

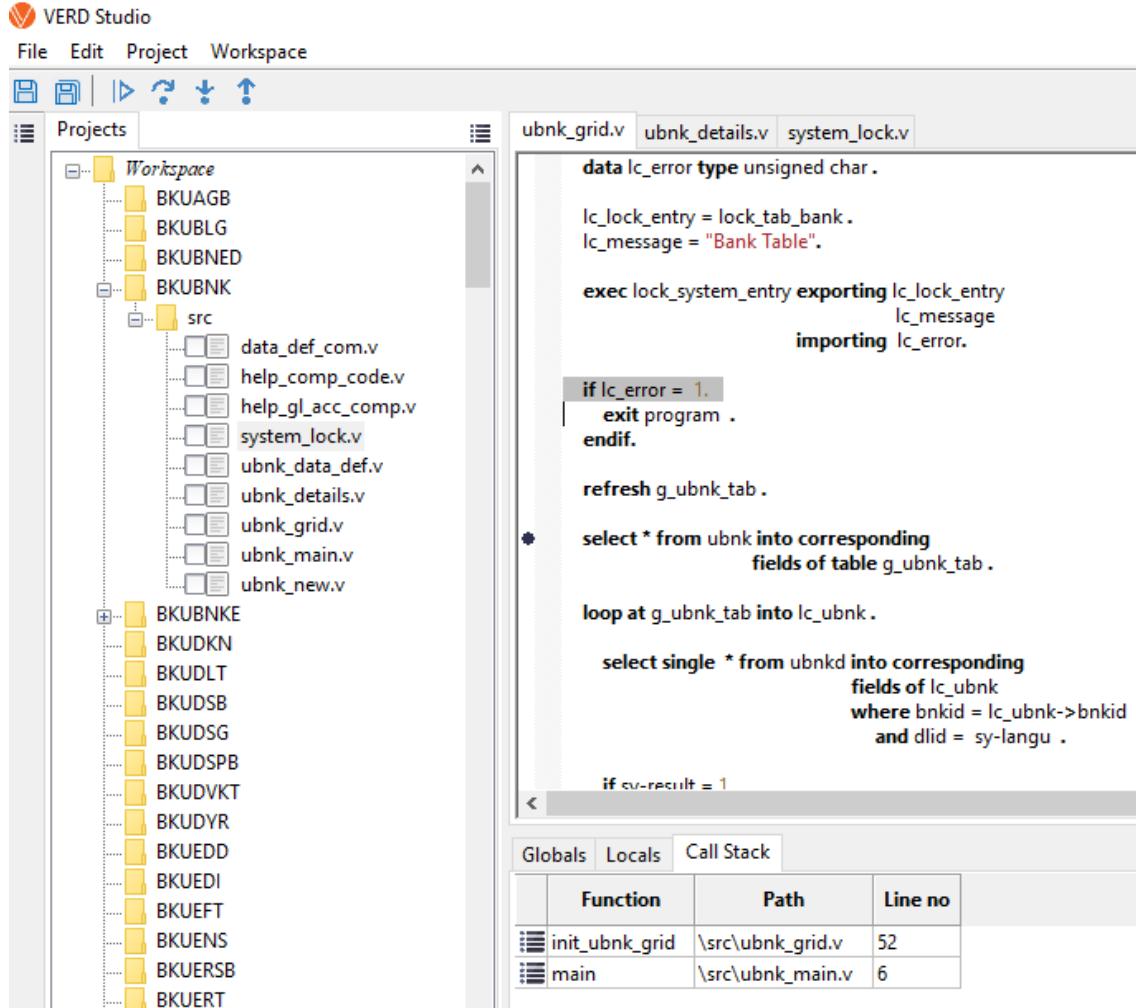
  if sy-result = 0 .

Call Stack
Function Path Line no
lock_system_entry \src\system_lock.v 14
init_ubnk_grid \src\ubnk_grid.v 48
main \src\ubnk_main.v 6

```

Screenshot 3.80 Click “Step Out” when debugger stopped at breakpoint

- The debugger completes all operations in the function that it is currently in, and to stop again in the next instruction in the calling function. The new line where debugger stopped is selected by VERD Studio. (*Screenshot 3.81*)



VERD Studio

File Edit Project Workspace

Projects

Workspace

- BKUAGB
- BKUBLG
- BKUBNED
- BKUBNK**
 - src
 - data_def_com.v
 - help_comp_code.v
 - help_gl_acc_comp.v
 - system_lock.v
 - ubnk_data_def.v
 - ubnk_details.v
 - ubnk_grid.v
 - ubnk_main.v
 - ubnk_new.v
- BKUBNKE
- BKUDKN
- BKUDLT
- BKUDSB
- BKUDSG
- BKUDSPB
- BKUDVKT
- BKUDYR
- BKUEDD
- BKUEDI
- BKUEFT
- BKUENS
- BKUERSB
- BKUERT

ubnk_grid.v ubnk_details.v system_lock.v

```

data lc_error type unsigned char.

lc_lock_entry = lock_tab_bank.
lc_message = "Bank Table".

exec lock_system_entry exporting lc_lock_entry
                           lc_message
                           importing lc_error.

if lc_error = 1.
  exit program .
endif.

refresh g_ubnk_tab.

select * from ubnk into corresponding
               fields of table g_ubnk_tab .

loop at g_ubnk_tab into lc_ubnk.

  select single * from ubnkd into corresponding
                             fields of lc_ubnk
                             where bnkid = lc_ubnk->bnkid
                               and dlid = sy-langu .

  if sy-result = 1

```

Globals Locals Call Stack

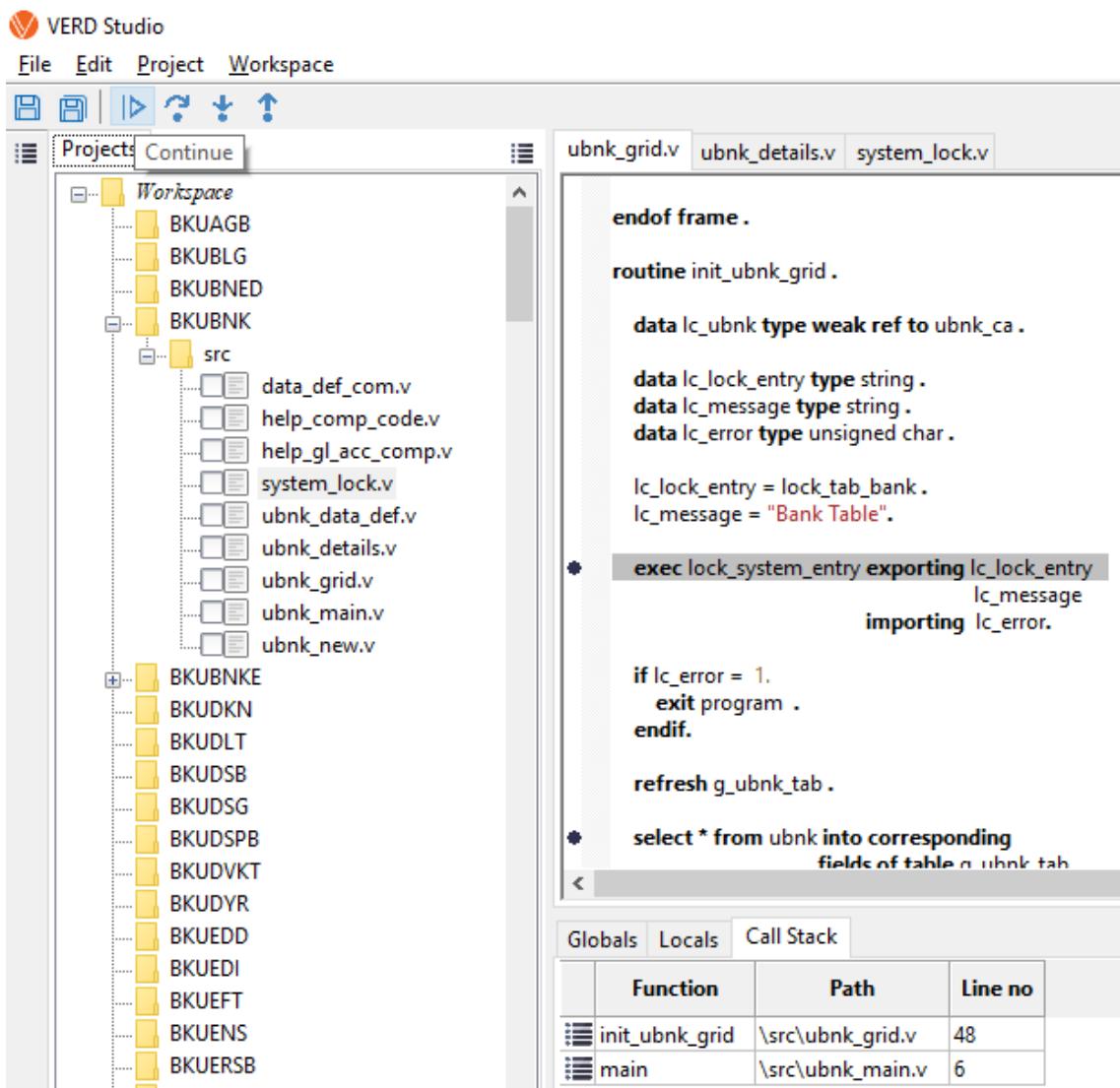
	Function	Path	Line no
init_ubnk_grid	\src\ubnk_grid.v	52	
main	\src\ubnk_main.v	6	

Screenshot 3.81 Source code is updated after “Step Out” instruction

3.34.5. Continue

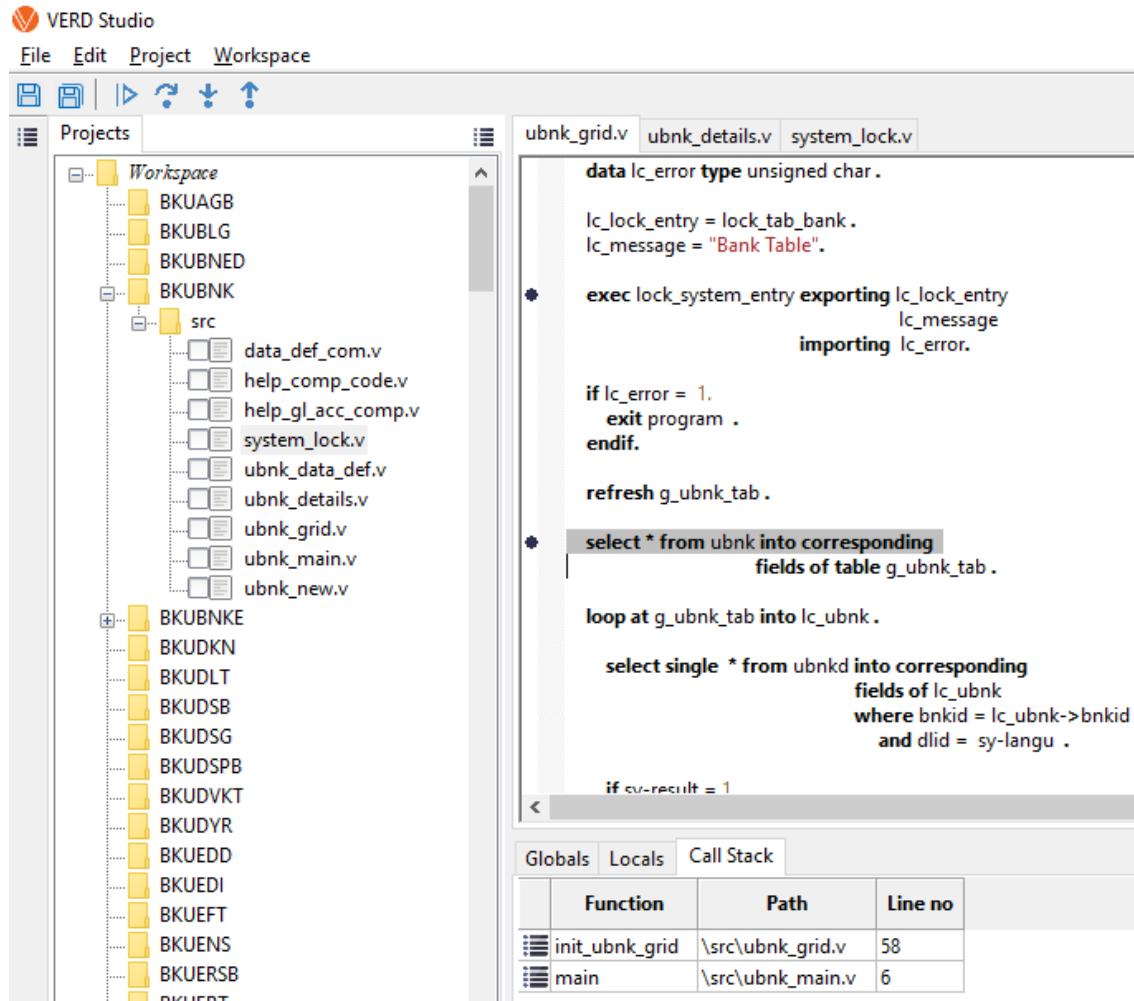
When the debugger reaches a breakpoint, it stops the program flow and waits for requests from the user. One of these requests is the “**continue**” request. This request allows the debugger to continue the process until the program reaches the next breakpoint. Steps for “**continue**” request are listed below.

- Click on “**Continue**” icon  in toolbar when debugger stops the program at a breakpoint. (*Screenshot 3.82*)



Screenshot 3.82 Click “Continue” when debugger stopped at breakpoint

- The debugger continues the process until the program reaches the next breakpoint if exists. The new line where debugger stopped is selected by VERD Studio. (*Screenshot 3.83*)



VERD Studio

File Edit Project Workspace

Projects

Workspace

- BKUAGB
- BKUBLG
- BKUBNED
- BKUBNK**
- src

 - data_def_com.v
 - help_comp_code.v
 - help_gl_acc_comp.v
 - system_lock.v
 - ubnk_data_def.v
 - ubnk_details.v
 - ubnk_grid.v
 - ubnk_main.v
 - ubnk_new.v

- BKUBNKE
- BKUDKN
- BKUDLT
- BKUDSB
- BKUDSG
- BKUDSPB
- BKUDVKT
- BKUDYR
- BKUEDD
- BKUEDI
- BKUEFT
- BKUENS
- BKUERSB
- RKIERT

ubnk_grid.v ubnk_details.v system_lock.v

```

data lc_error type unsigned char.

lc_lock_entry = lock_tab_bank .
lc_message = "Bank Table".

exec lock_system_entry exporting lc_lock_entry
                           lc_message
                           importing lc_error.

if lc_error = 1.
  exit program .
endif.

refresh g_ubnk_tab .

select * from ubnk into corresponding
               fields of table g_ubnk_tab .

loop at g_ubnk_tab into lc_ubnk .

  select single * from ubnkd into corresponding
               fields of lc_ubnk
               where bnkid = lc_ubnk->bnkid
                     and dlid = sy-langu .

  if sy-result = 1

```

Globals Locals Call Stack

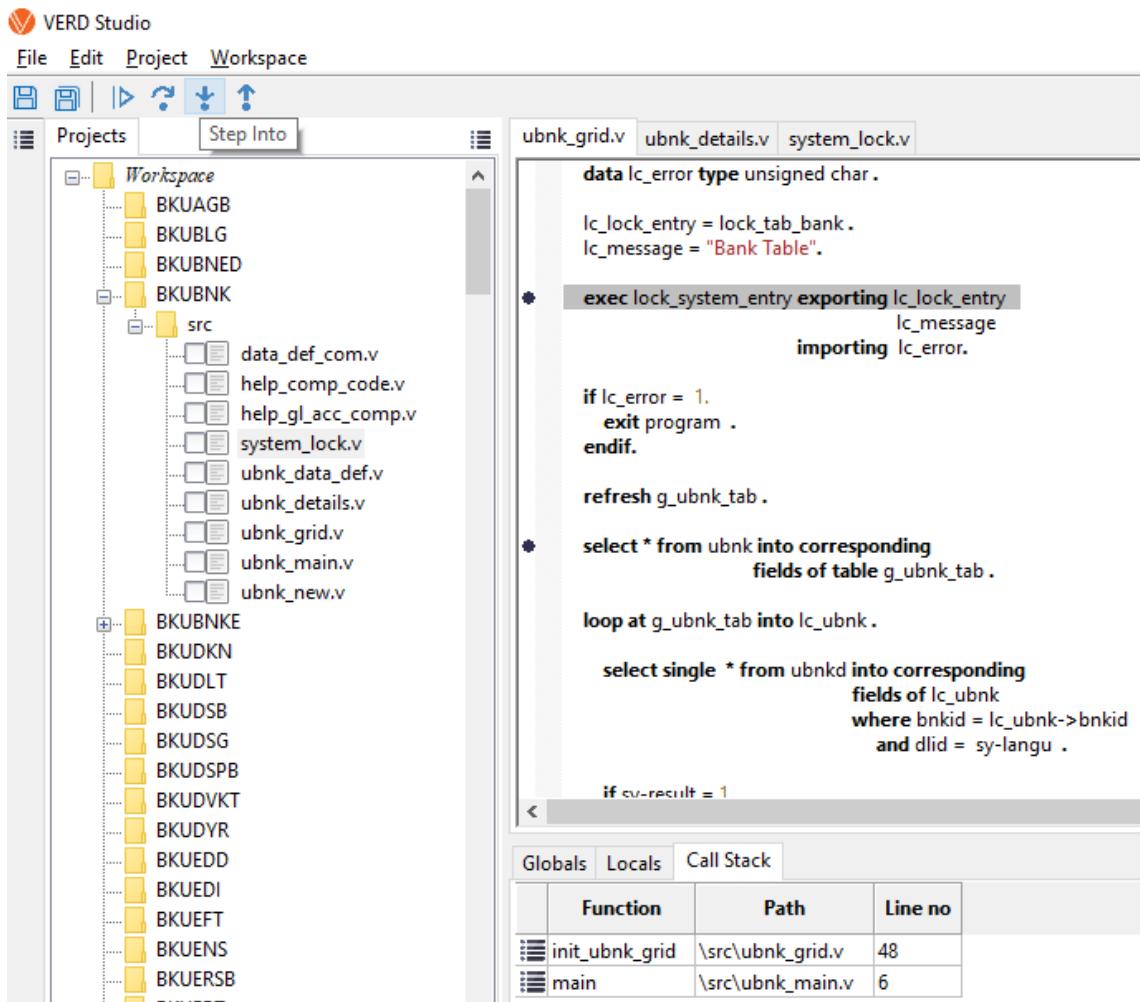
	Function	Path	Line no
init_ubnk_grid	\src\ubnk_grid.v	58	
main	\src\ubnk_main.v	6	

Screenshot 3.83 Source code is updated after “Continue” instruction

3.34.6. Step into Function

When the debugger reaches a breakpoint, it stops the program flow and waits for user requests. One of the requests is the "step into function" request. This request allows the debugger to stop at the first instruction of the "exec" function if debugger already stopped at "exec" instruction. If debugger hasn't stopped at "exec" instruction, it would stop at the next instruction in the current function in this case "step into function" request performs the same function as the "step over to next instruction" request. Steps for stepping into function are listed below.

- Click on "Step Into" icon  in toolbar when debugger stops the program at a breakpoint. (*Screenshot 3.84*)



Screenshot 3.84 Click "Step Into" when debugger stopped at breakpoint

- The debugger stops at the first instruction of the “exec” function if debugger already stopped at “exec” instruction. If debugger hasn't stopped at “exec” instruction, it would stop at the next instruction. The new line where debugger stopped is selected by VERD Studio. (*Screenshot 3.85*)

VERD Studio

File Edit Project Workspace

Projects

Workspace

ubnk_grid.v ubnk_details.v system_lock.v

```

depends "data_def_com.v".

data gs_sklt_tab type table of sklt.

routine lock_system_entry importing p_lock_entry type string
                           p_message type string
                           exporting p_error type unsigned char.

data lc_sklt type sklt .
data lc_sklt_ref type sklt .

data lc_message_type type string .

lc_sklt-skltid = p_lock_entry .
lc_sklt-klid = sy-username .
lc_sklt-uuid = sy-uuid .

read table gs_sklt_tab into lc_sklt_ref
      where skltid = p_lock_entry .

if sy-result = 0 .

  commit insert into sklt from lc_sklt .

  if sy-result = 0 .

```

Globals Locals Call Stack

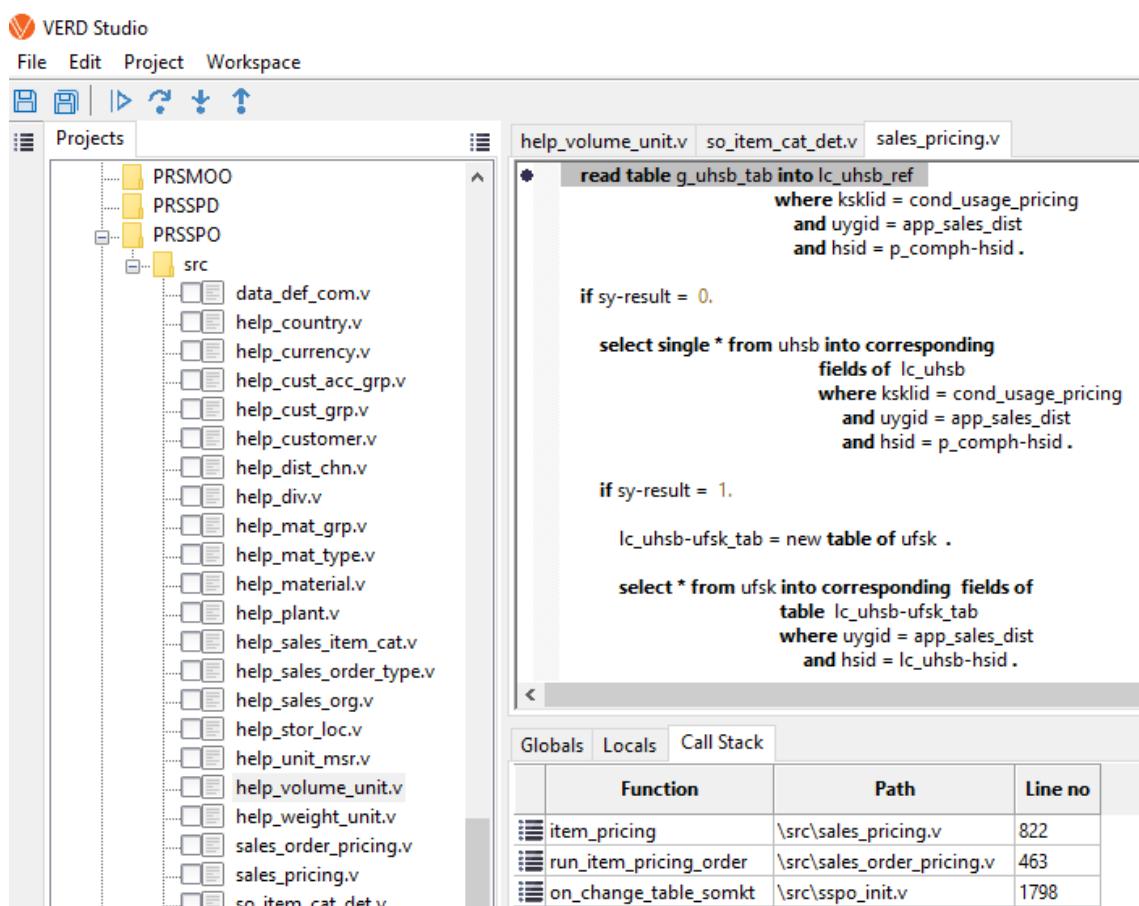
	Function	Path	Line no
lock_system_entry	\src\system_lock.v	14	
init_ubnk_grid	\src\ubnk_grid.v	48	
main	\src\ubnk_main.v	6	

Screenshot 3.85 Source code is updated after “Step Into” instruction

3.34.7. Navigate through Call Stack

When the debugger reaches a breakpoint, it stops the program flow and sends call stack info to VERD Studio. User can navigate through call stack in VERD Studio. Steps for navigating through call stack are listed below.

- Double click on “**Details**” icon  in any row of call stack that is displayed on “Call Stack” tab when debugger stops the program at a breakpoint. (*Screenshot 3.86*)



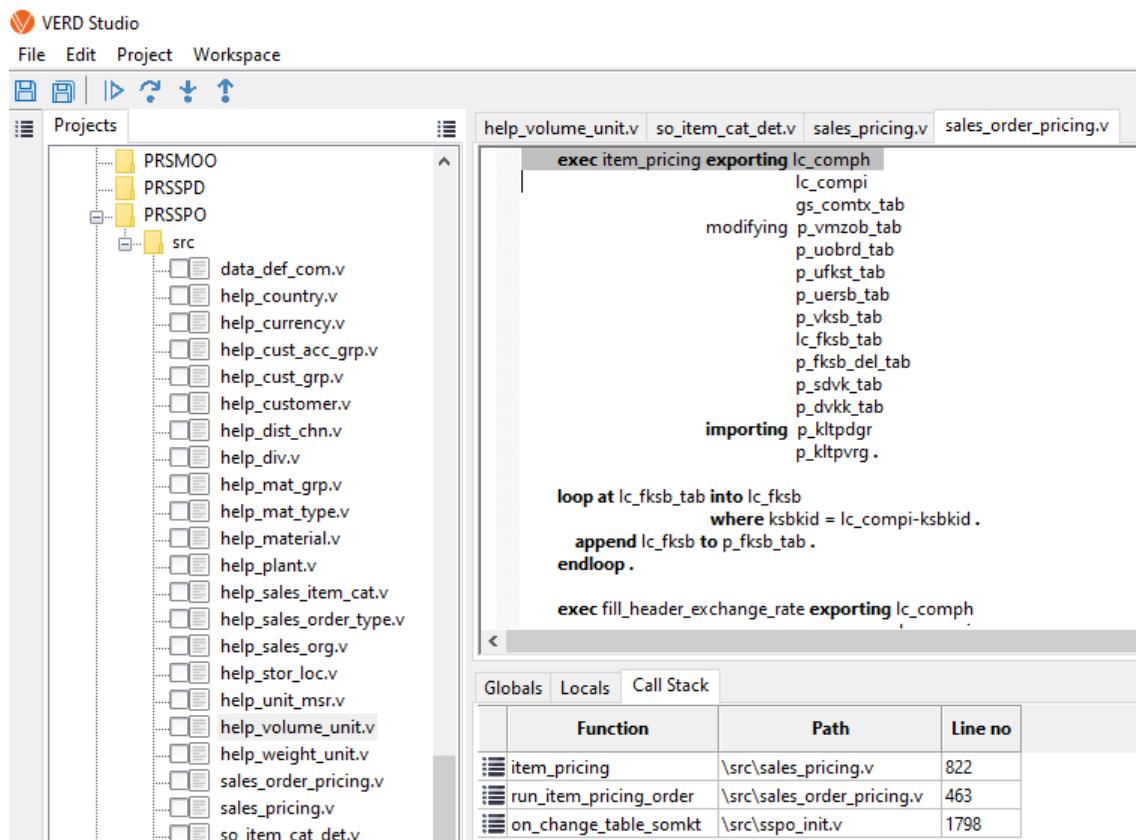
The screenshot shows the VERD Studio interface with the following details:

- Projects View:** Shows several projects: PRSMOO, PRSSPD, PRSSPO, and their subfolders and files like help_country.v, help_currency.v, etc.
- Code Editor:** Displays ABAP code with syntax highlighting for SQL statements and comments.
- Call Stack Tab:** Shows the call stack with three entries:

	Function	Path	Line no
	item_pricing	\src\sales_pricing.v	822
	run_item_pricing_order	\src\sales_order_pricing.v	463
	on_change_table_somkt	\src\sspo_init.v	1798

Screenshot 3.86 Double Click “Details” icon of a call stack row

- The VERD Studio then navigates to source file and line no for the requested call stack info. (*Screenshot 3.87*)



The screenshot shows the VERD Studio interface. The menu bar includes File, Edit, Project, and Workspace. The toolbar has icons for opening files, saving, and navigating. The left pane displays the 'Projects' tree, which contains three main projects: PRSMOO, PRSSPD, and PRSSPO. PRSSPO is expanded, showing a 'src' folder containing numerous source files such as data_def_com.v, help_country.v, help_currency.v, etc. The right pane shows a code editor with several tabs: help_volume_unit.v, so_item_cat_det.v, sales_pricing.v, and sales_order_pricing.v. The current tab is sales_pricing.v, displaying the following VBA-like pseudocode:

```

exec item_pricing exporting lc_comph
| lc_compi
| gs_comtx_tab
modifying p_vmzob_tab
p_uobrd_tab
p_ufkst_tab
p_uersb_tab
p_vksb_tab
lc_fksb_tab
p_fksb_del_tab
p_sdvk_tab
p_dvkk_tab
importing p_kltpdgr
p_kltpvrg .

loop at lc_fksb_tab into lc_fksb
  where ksbkid = lc_compi-ksbkid .
  append lc_fksb to p_fksb_tab .
endloop .

exec fill_header_exchange_rate exporting lc_comph

```

Below the code editor is a table titled 'Call Stack' with three tabs: Globals, Locals, and Call Stack. The Call Stack tab is selected, showing the following data:

	Function	Path	Line no
item_pricing	\src\sales_pricing.v	822	
run_item_pricing_order	\src\sales_order_pricing.v	463	
on_change_table_somkt	\src\sspo_init.v	1798	

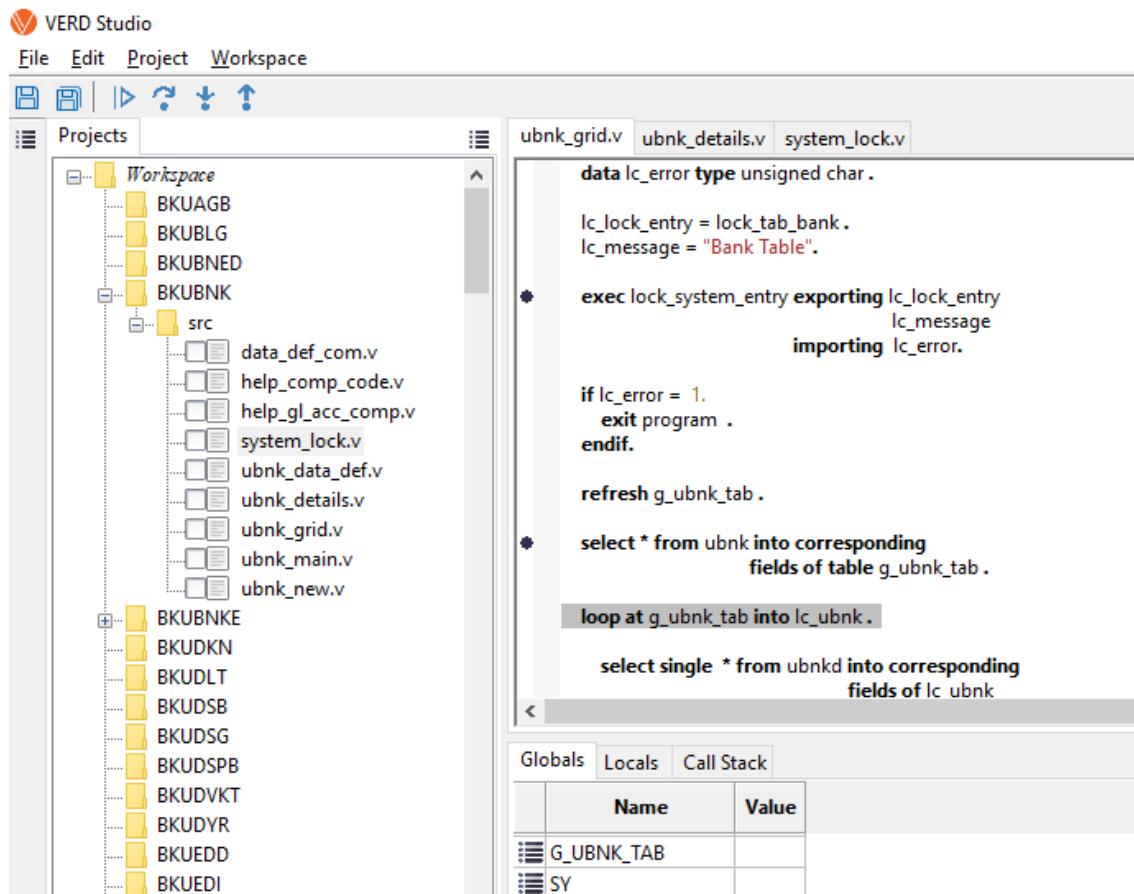
Screenshot 3.87 VERD Studio navigates to source file for the requested call stack.

3.34.8. Send Content of Variable

When the debugger reaches a breakpoint, it stops the program flow and waits for requests from the user. One such request is "**send content of variable**". This request is sent to the debugger by VERD Studio whenever the user clicks on any local or global variable.

In response to this request, the debugger sends the variable's field information and content to the VERD Studio. VERD Studio then displays the content of variable by using the data sent by debugger. Steps for this request are listed below.

- Double click on “**Details**” icon  in any global or local variables that are displayed on “**Globals**” and “**Locals**” tab respectively when debugger stops the program at a breakpoint. (*Screenshot 3.88*)



The screenshot shows the VERD Studio interface. The menu bar includes File, Edit, Project, and Workspace. The toolbar has icons for file operations like Open, Save, and Run. The left pane is the Projects view, showing a hierarchy of workspace projects: Workspace, BKUAGB, BKUBLG, BKUBND, BKUBNK, and src. Under src, there are several files: data_def_com.v, help_comp_code.v, help_gl_acc_comp.v, system_lock.v, ubnk_data_def.v, ubnk_details.v, ubnk_grid.v, ubnk_main.v, and ubnk_new.v. The right pane shows the code editor with the file ubnk_grid.v open. The code contains VBA-like pseudocode:

```

data lc_error type unsigned char.

lc_lock_entry = lock_tab_bank.
lc_message = "Bank Table".

* exec lock_system_entry exporting lc_lock_entry
  lc_message
  importing lc_error.

if lc_error = 1.
  exit program .
endif.

refresh g_ubnk_tab .

* select * from ubnk into corresponding
  fields of table g_ubnk_tab .

loop at g_ubnk_tab into lc_ubnk .

  select single * from ubnkd into corresponding
    fields of lc_ubnk

```

Below the code editor is a table titled "Globals" showing variable information:

	Name	Value
	G_UBNK_TAB	
	SY	

Screenshot 3.88 Double Click “Details” icon of a program variable

- The debugger sends the variable's field information and content to the VERD Studio. VERD Studio then displays the content of variable by using the data sent by debugger. (*Screenshot 3.89*)

G_UBNK_TAB							
	BNKID	IBNKD	BNKISM	UBNKS_TAB	UBNKS_DEL_TAB	UPDIND_LN	UPDIND
AK	00046			NULL	NULL	0	0
GR	00062			NULL	NULL	0	0
IS	00064			NULL	NULL	0	0
HLK	00012			NULL	NULL	0	0
VKF	00015			NULL	NULL	0	0
ZR	00010			NULL	NULL	0	0
YPK	00067			NULL	NULL	0	0
TEB	00032			NULL	NULL	0	0
CT	00092			NULL	NULL	0	0
QNF	00111			NULL	NULL	0	0

Screenshot 3.89 Content of variable displayed by VERD Studio

3.34.9. Send Content of Variable that Mouse is over

When the debugger reaches a breakpoint, it stops the program flow and waits for requests from the user. One such request is the "**send contents of the variable that mouse is over**" request. This request is sent when the user keeps the mouse cursor over a variable in the source code for more than 500 milliseconds.

In response to this request, the debugger sends both the variable's field information and its contents to the VERD Studio. VERD Studio then displays the content of variable by using the data sent by debugger. Steps for this request are listed below.

- Keep mouse cursor a program variable that you want to display contents of, for more than 500 milliseconds when debugger stops the program at a breakpoint.
- (Screenshot 3.90)**

The screenshot shows the VERD Studio interface. The top menu bar includes File, Edit, Project, and Workspace. The left sidebar is titled 'Projects' and shows a tree structure of workspace projects: Workspace, BKUAGB, BKUBLG, BKUBNED, BKUBNK, and BKUBNKE. Under the BKUBNK project, there is a 'src' folder containing several files: data_def_com.v, help_comp_code.v, help_gl_acc_comp.v, system_lock.v, ubnk_data_def.v, ubnk_details.v, ubnk_grid.v, ubnk_main.v, and ubnk_new.v. The main workspace area displays a portion of the 'ubnk_grid.v' file:

```

refresh g_ubnk_tab .

* select * from ubnk into corresponding
  fields of table g_ubnk_tab .

loop at g_ubnk_tab into lc_ubnk .

  select single * from ubnkd into corresponding
    fields of lc_ubnk
    where bnid = lc_ubnk->bnid
      and dlid = sy-langu .

  if sy-result = 1 .
    lc_ubnk->updind_ln = updind_dbexists .
  endif.

  lc_ubnk->ubnks_tab = new table of ubnks_ca .
  lc_ubnk->ubnks_del_tab = new table of ubnks_ca .
  lc_ubnk->updind = updind_dbexists .

endloop.

endroutine .

```

At the bottom of the workspace area, there are tabs for Globals, Locals, and Call Stack. The Locals tab is selected, showing a table with two rows:

	Name	Value
G_UBNK_TAB		
SY		

Screenshot 3.90 Keep mouse cursor a program variable for more than 500 ms

- The debugger sends the variable's field information and content to the VERD Studio. VERD Studio then displays the content of variable by using the data sent by debugger. (*Screenshot 3.91*)

The screenshot shows the VERD Studio interface. The left pane displays a project tree under 'Workspace' with various source files like 'data_def_com.v', 'ubnk_grid.v', etc. The right pane shows a code editor with a SQL-like script and a data grid. The data grid, titled 'G_UBNK_TAB', contains the following data:

	BNKID	IBNKD	BNKISM	UBNKS_TAB	UBNKS_DEL_TAB	UPDIND_LN	UPDIND
	AK	00046		NULL	NULL	0	0
	GR	00062		NULL	NULL	0	0
if sy-re	IS	00064		NULL	NULL	0	0
endif.	HLK	00012		NULL	NULL	0	0
	VKF	00015		NULL	NULL	0	0
lc_ubn	ZR	00010		NULL	NULL	0	0
lc_ubn	YPK	00067		NULL	NULL	0	0
endloop.	TEB	00032		NULL	NULL	0	0
	CT	00092		NULL	NULL	0	0
endroutine.	QNF	00111		NULL	NULL	0	0

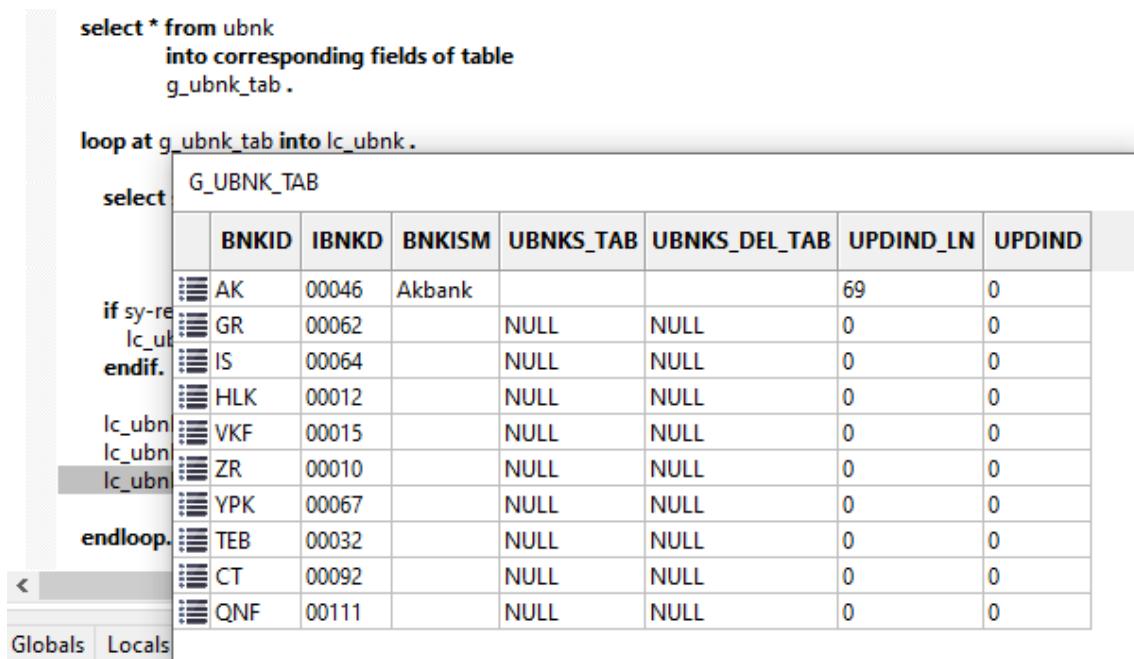
Below the grid, there are tabs for 'Globals', 'Locals', and 'Call Stack'. The 'Globals' tab shows entries for 'G_UBNK_TAB' and 'SY'.

Screenshot 3.91 Content of variable displayed by VERD Studio

3.34.10. Display a single row in a table

In VERD debugger, a single row in a table whose content is already displayed can be displayed in detail. Steps for displaying a single row in a table whose content is already displayed, are listed below

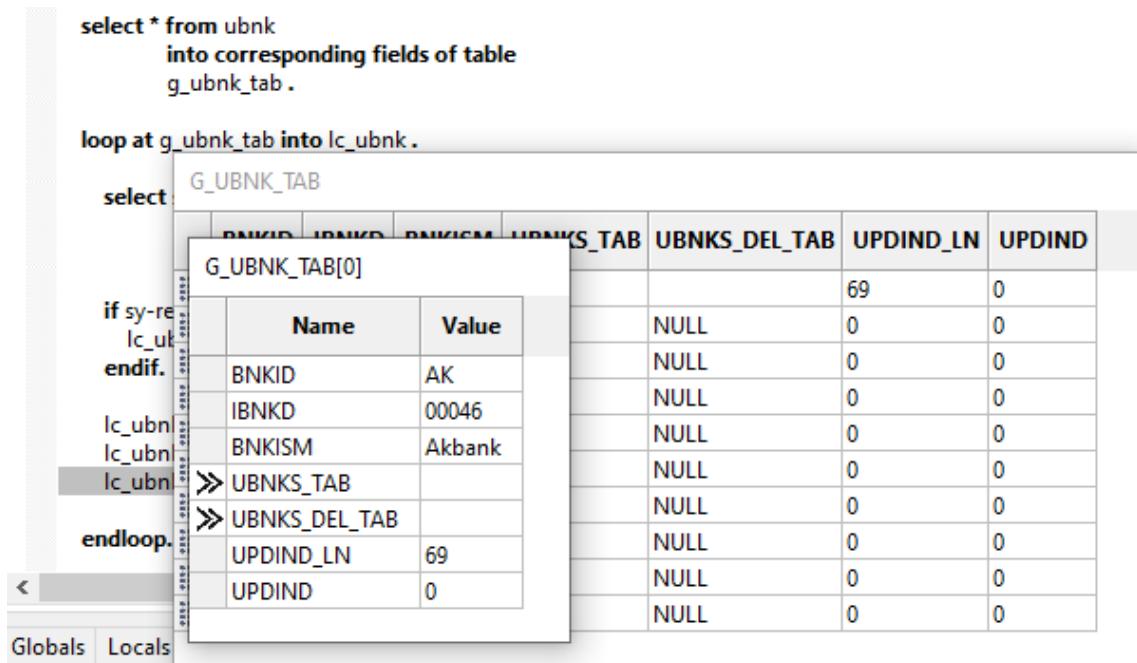
- Double click on “**Details**” icon  in any row in a table whose content is already displayed. VERD Studio sends a request to debugger to display a single row.(Screenshot 3.92)



	BNKID	IBNKD	BNKISM	UBNKS_TAB	UBNKS_DEL_TAB	UPDIND_LN	UPDIND
AK	00046	Akbank				69	0
GR	00062		NULL	NULL		0	0
IS	00064		NULL	NULL		0	0
HLK	00012		NULL	NULL		0	0
VKF	00015		NULL	NULL		0	0
ZR	00010		NULL	NULL		0	0
YPK	00067		NULL	NULL		0	0
TEB	00032		NULL	NULL		0	0
CT	00092		NULL	NULL		0	0
QNF	00111		NULL	NULL		0	0

*Screenshot 3.92 Double Click “**Details**” icon of a table row*

- The debugger sends the contents of row that is being requested. VERD Studio then displays the content of row by using the data sent by debugger. (*Screenshot 3.93*)



The screenshot shows a debugger interface with a code editor and a data viewer. The code editor contains the following ABAP code:

```

select * from ubnk
  into corresponding fields of table
  g_ubnk_tab .

loop at g_ubnk_tab into lc_ubnk .
  select
    BNKID  IBNKD   BNKISM  UBNKS_TAB  UBNKS_DEL_TAB  UPDIND_LN  UPDIND
    from G_UBNK_TAB
    into G_UBNK_TAB[0]
  if sy-recl = 100.
    lc_ubnk = g_ubnk_tab.
  endif.
  lc_ubnk = lc_ubnk + 1.
  lc_ubnk = lc_ubnk - 1.
endloop.

```

The data viewer shows a table with the following data:

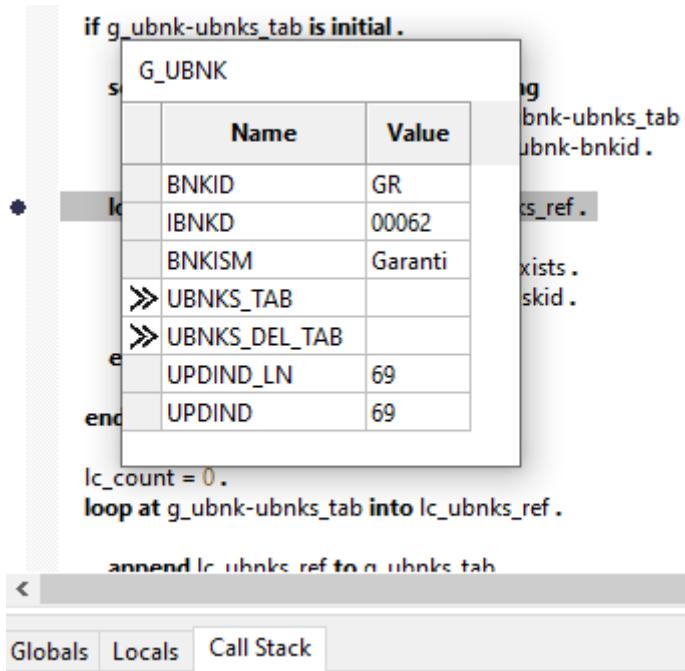
	Name	Value
BNKID	AK	
IBNKD	00046	
BNKISM	Akbank	
»> UBNKS_TAB		
»> UBNKS_DEL_TAB		
UPDIND_LN	69	
UPDIND	0	

Screenshot 3.93 Content of table row displayed by VERD Studio

3.34.11. Display a type ref to field within structure

In VERD debugger, a type ref to field in a structure whose contents is already displayed can be displayed in detail. Steps for displaying a type ref field in a structure whose contents is already displayed, are listed below

- Double click on “**Details**” icon  in a type ref to field in a structure whose content is already displayed. “**Details**” icon  is only displayed for type ref to fields, other character and numeric fields does not have such icon. VERD Studio sends a request to debugger to display the details of type ref to field.(Screenshot 3.94)



Screenshot 3.94 Double Click “**Details**” icon of a type ref to field

- The debugger sends the contents of type ref to field that is being requested. VERD Studio then displays the content of type ref to field by using the data sent by debugger. (*Screenshot 3.95*)

```

if g_ubnk-ubnks_tab is initial .
  G_UBNK
  |
  +----+-----+-----+
  |      Name       Value   |
  +----+-----+-----+
  |      BNKID     GR      |
  +----+-----+-----+
  |      IBNKD    00062   |
  +----+-----+-----+
  G_UBNK->UBNKS_TAB
  |
  +----+-----+-----+-----+-----+-----+-----+
  |      BNKID | SKID | THKID | KHKID | ISKID | UPDIND |
  +----+-----+-----+-----+-----+-----+
  |      GR    | 1000 | 1010000001 |        |        | 0       |
  +----+-----+-----+-----+-----+-----+
end

lc_count = 0.
loop at g_ubnk-ubnks_tab into lc_ubnks_ref .

append lc_ubnks_ref to a_ubnks_tab

```

Screenshot 3.95 Content of type ref to field displayed by VERD Studio