

VectorCAST IBM Rational Doors Integration

Version 1.0

2023-07-13

Application Note AN-ACT-1-012

Author Yue, Jeanne

Restrictions Public Document

Abstract Information on the IBM Rational Doors and VectorCAST Integration.

Table of Contents

1	Overview	2
2	Use Cases	2
2.1	Use Case: Create a Requirements Repository	2
2.2	Use Case: Authenticate with the Server.....	4
2.3	Use Case: Import Requirements	9
2.4	Use Case: Link Requirements to Test Cases.....	11
2.5	Use Case: Export Requirements.....	14
2.6	Use Case: Review Updated Requirements	21
3	Additional Resources	24
4	Trademarks	24
5	Contacts.....	24

1 Overview

For VectorCAST/C++, the VectorCAST Requirements Gateway (RGW) provides traceability between software requirements and test cases and allows the import and mapping of requirements to test cases.

Requirements Gateway provides a way to:

- Import testing requirements from your Requirements Management Tool (RMT) to a VectorCAST
- Assign specific requirements to testcases in a unit test environment
- Export the resulting pass or fail status back to the RMT

In RGW, the requirements database is implemented using JSON, allowing the user to commit the repository to Source Control Management (SCM).

In addition to CSV Files, VectorCAST RGW supports the following Requirements Management Systems:

- codeBeamer
- IBM CLM
- IBM Rational Doors
- Jama
- Polarion®
- Visure

This Application Note will show you how to use the RGW 3.0 IBM Rational DOORS Integration.

2 Use Cases

The following use cases will be shown:

- Create a Requirements Repository
- Authenticate with the Server
- Import Requirements
- Link Requirements to Test Cases
- Export Requirements
- Review Updated Requirements

2.1 Use Case: Create a Requirements Repository

To use the Requirements Gateway, a location must be specified for the Repository. If no location is specified, a prompt opens allowing the user to select the location. A standard Choose a Directory dialog opens, prompting the user to enter the path to the directory.

Set the repository location by selecting **Tools => Requirements Gateway => Options...** from the Menu Bar shown in Figure 1.

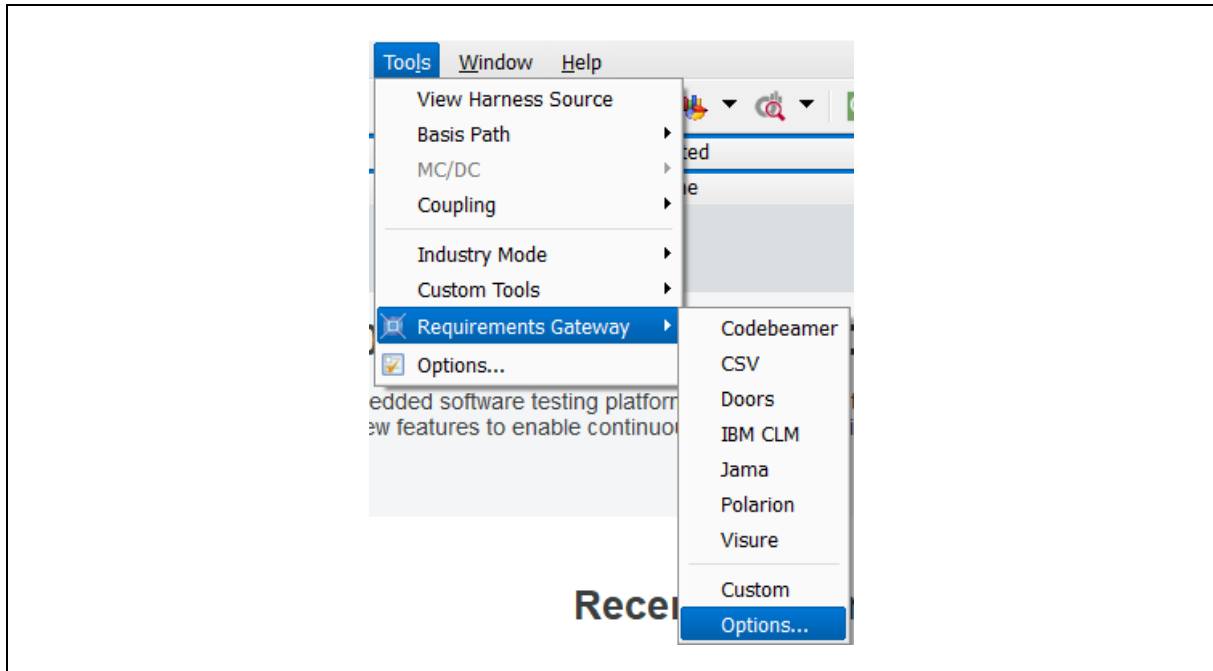


Figure 1-Select Requirements Gateway Options

The **Requirements Gateway Options** dialog will open as shown in Figure 2.

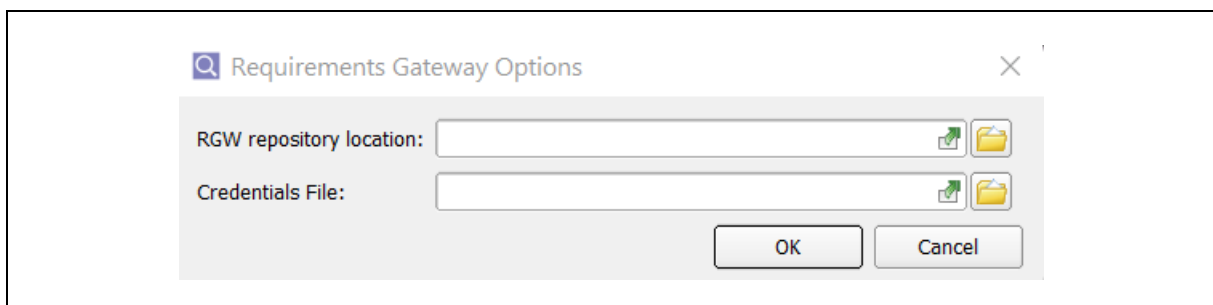


Figure 2-The Requirements Gateway Options Dialog

In the **RGW Repository Location** field, enter the path to the repository and select the **OK** button as shown in Figure 3.

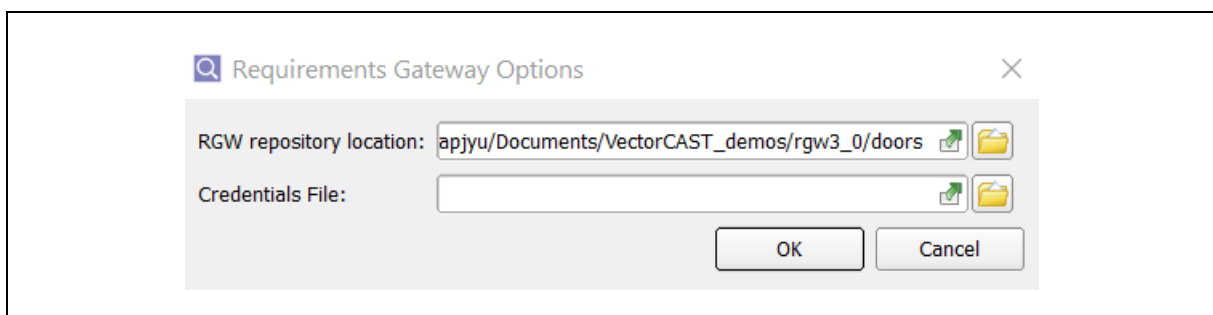


Figure 3-Set RGW Repository Location

The RGW repository location is now set, and the initial Requirements Repository is created in this location. VectorCAST creates a directory under the set location called `requirements_gateway`, that will hold the setting and log files used by RGW.

A new option starting in VectorCAST 2021SP5 is the ability to store server credentials in a Credentials File. The previous option of adding the credentials during the authentication step is still available. The Credentials File can be set in the Requirements Gateway Options GUI, shown in Figure 4, or set the environment variable `VCAST_CREDENTIALS_FILE` to the location of the file.

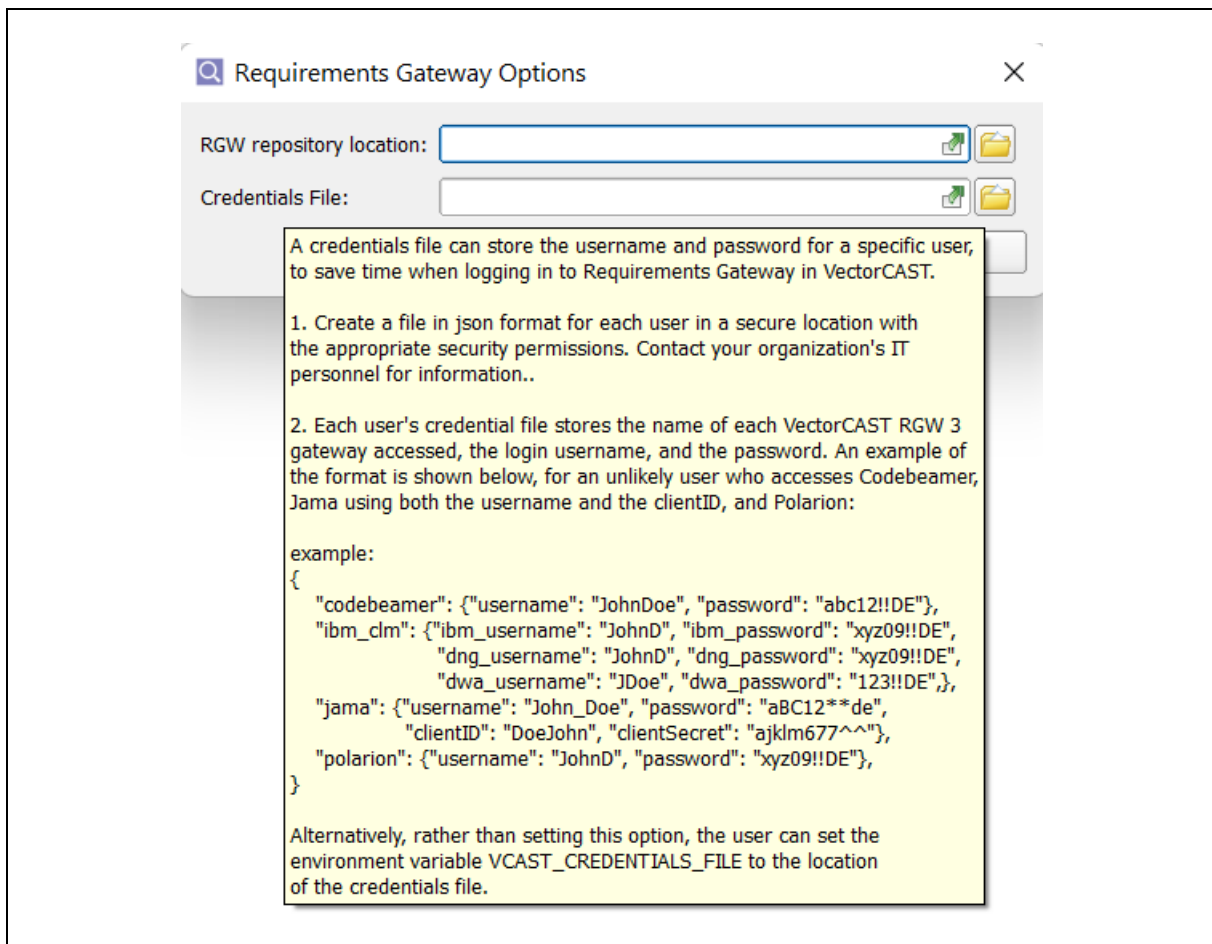


Figure 4-Credentials File

2.2 Use Case: Authenticate with the Server

To authenticate with the IBM rational Doors server, first Select **Tools => Requirements Gateway => Doors** from the Menu Bar as shown in Figure 5.

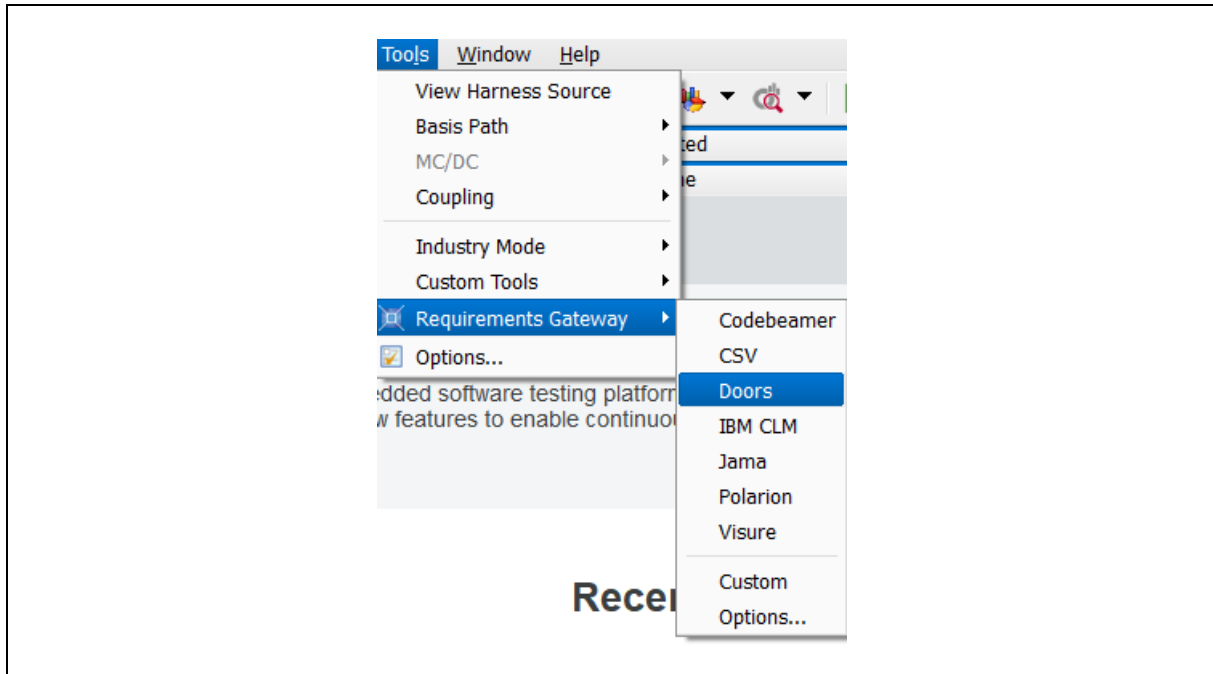


Figure 5-Select the Gateway

The **Requirements Gateway** dialog will open with **Doors** as the selected **Gateway Profile** and the **Authentication** tab selected at the bottom as shown in Figure 6.

The screenshot shows the 'Requirements Gateway' dialog box with the 'Authentication' tab selected. The dialog is titled 'Specify options for interfacing with the requirements gateway.' and has a 'Gateway Profile' dropdown set to 'Doors'. The 'Server Addresses visible' checkbox is checked. The 'Use Proxy' checkbox is unchecked, and the 'Proxy Server' field is empty. The 'DOORS Client & Database Server Settings' section contains fields for 'DOORS Client Executable' (with a folder icon), 'Database Server', and 'Database Port' (set to 36677). The 'DWA Server Settings - Import Only' section is expanded, showing 'Protocol' set to 'https', 'DWA Server', 'DWA Port' (set to 8443), and 'DWA Context Root' (set to dwa). The 'Use SSL unverified context' checkbox is checked. The 'User' section has fields for 'Username' and 'Password'. The 'Output' section is a large empty text area. An 'Authenticate' button is located at the bottom right of the dialog. At the bottom of the window, there are tabs for 'Authentication', 'Import', 'Export', 'Requirements', and 'Script', and a 'Close' button.

Figure 6-Authentication Tab

Once the required information is entered, select the **Authenticate** button as shown in Figure 7. The integration will now attempt connect to the Doors Database Server. If **DWA Server Settings** is checked, enter the server information for import. DWA stands for DOORS Web Access. With DWA, DOORS can be accessed via a web browser since it is based on the http protocol. DWA is an alternative way of importing requirements in a situation where a DOORS client is not installed or cannot be installed.

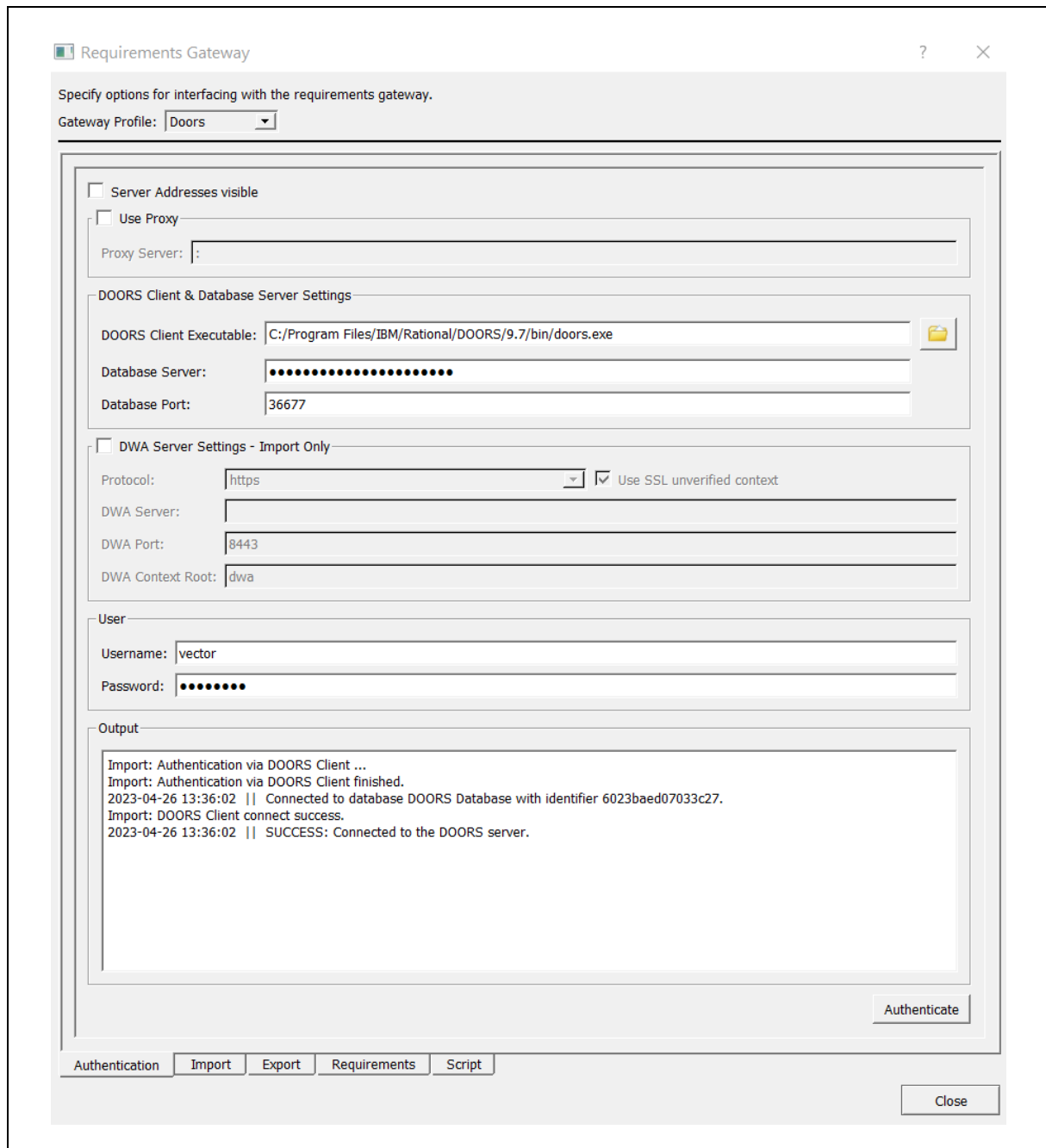


Figure 7-Authenticate with the Server

You will see a message indicating the status of the authentication. If the authentication fails, the doors.exe client will hang, as seen in Figure 8.

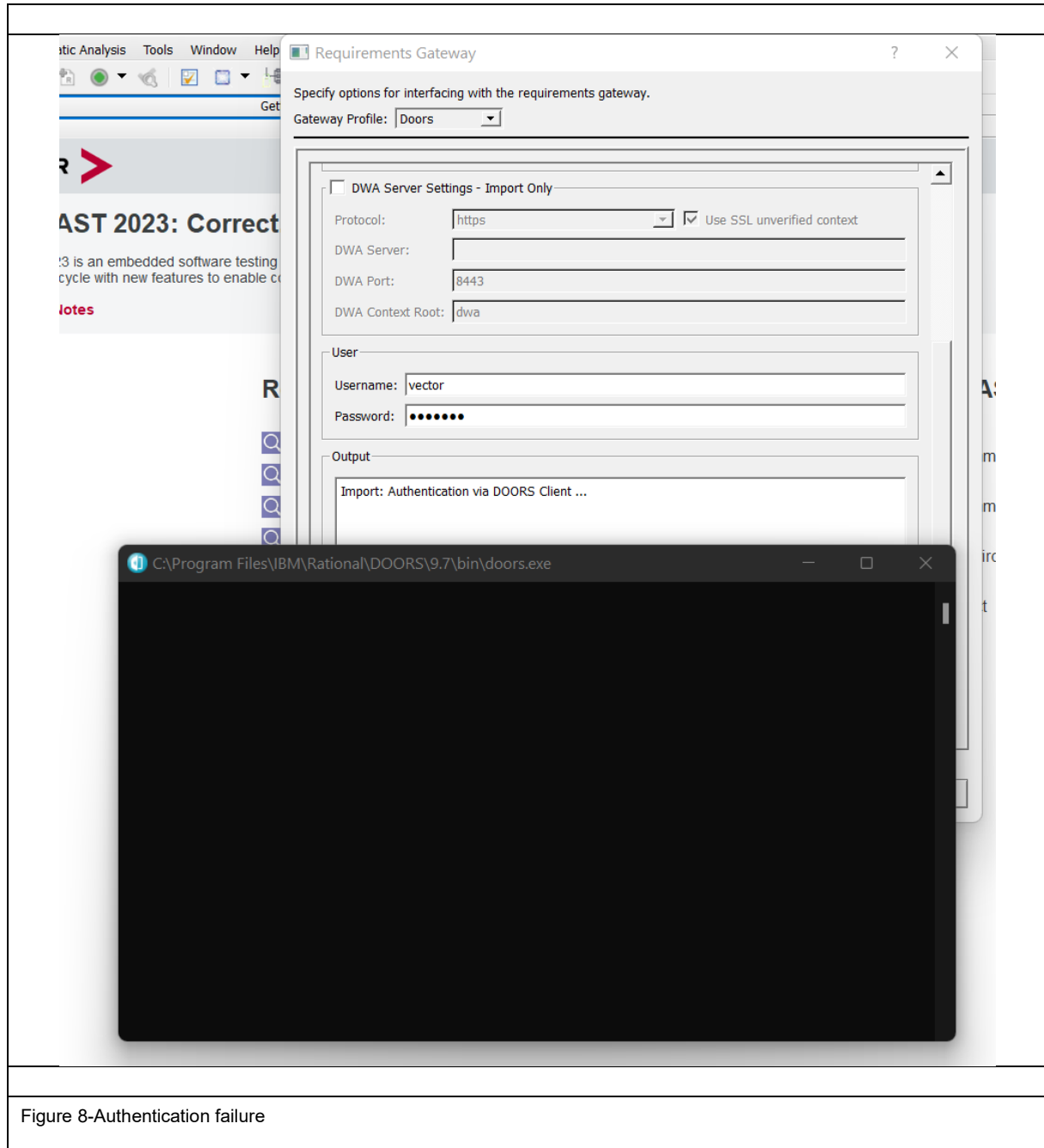


Figure 8-Authentication failure

Close the doors.exe client and a failure message will appear in the message output.

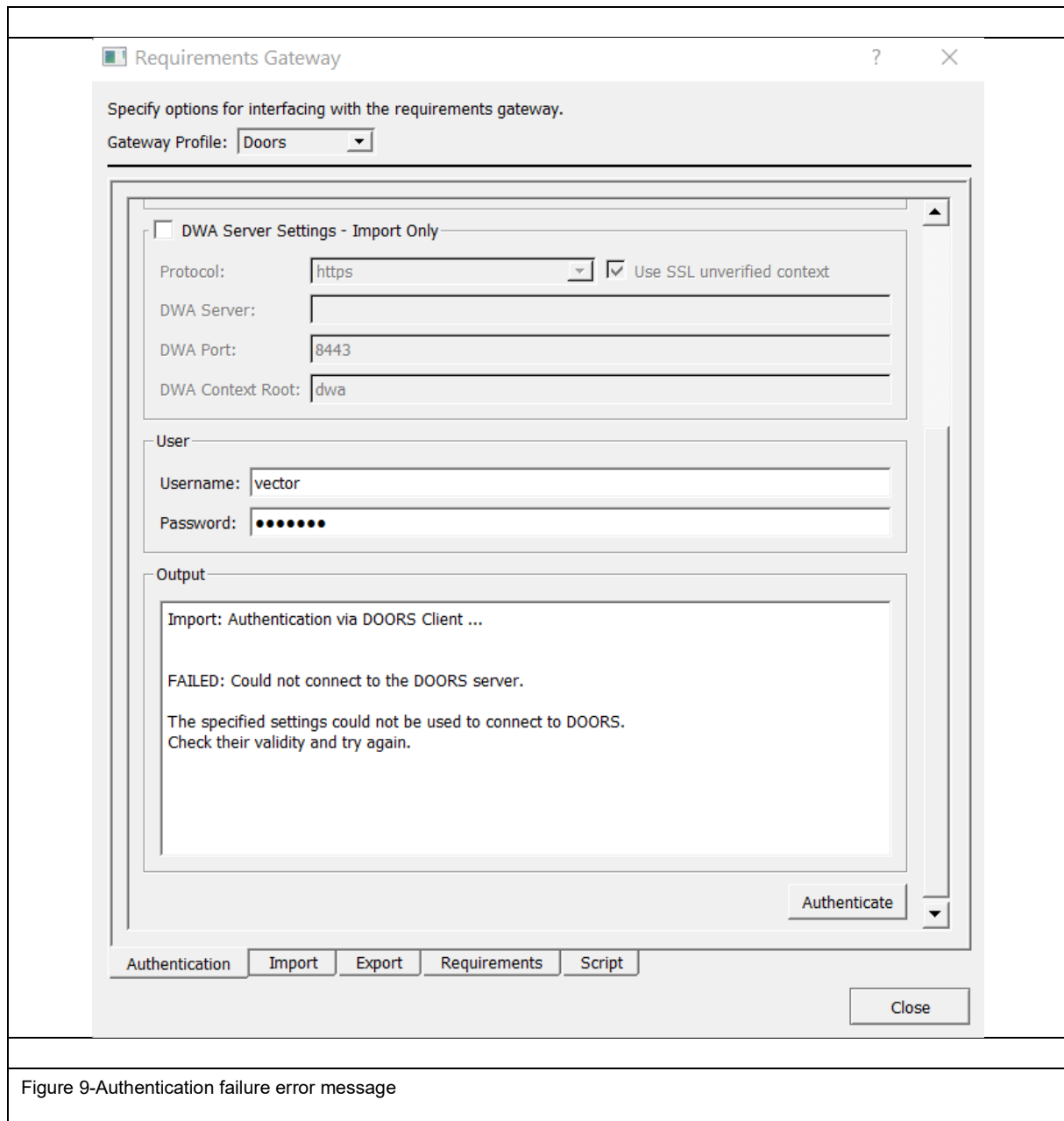


Figure 9-Authentication failure error message

2.3 Use Case: Import Requirements

To import requirements, you will need to open the Requirements Gateway dialog as shown in Figure 5. Once the dialog is open, select the import tab at the bottom. The Module / Service Provider Selection fields will automatically be filled with available modules if already authenticated. Select the valid module and then select either Requirements or Test Case as the content type of the module. On export the content type chosen now will determine if Test Cases are exported or only the Test Results. Fill in the Field Selection with the desired attributes, at least one of the following attributes must be set in order to import: ID, Title and/or Description. Once all the fields are filled as seen in Figure 9 then click the import button.

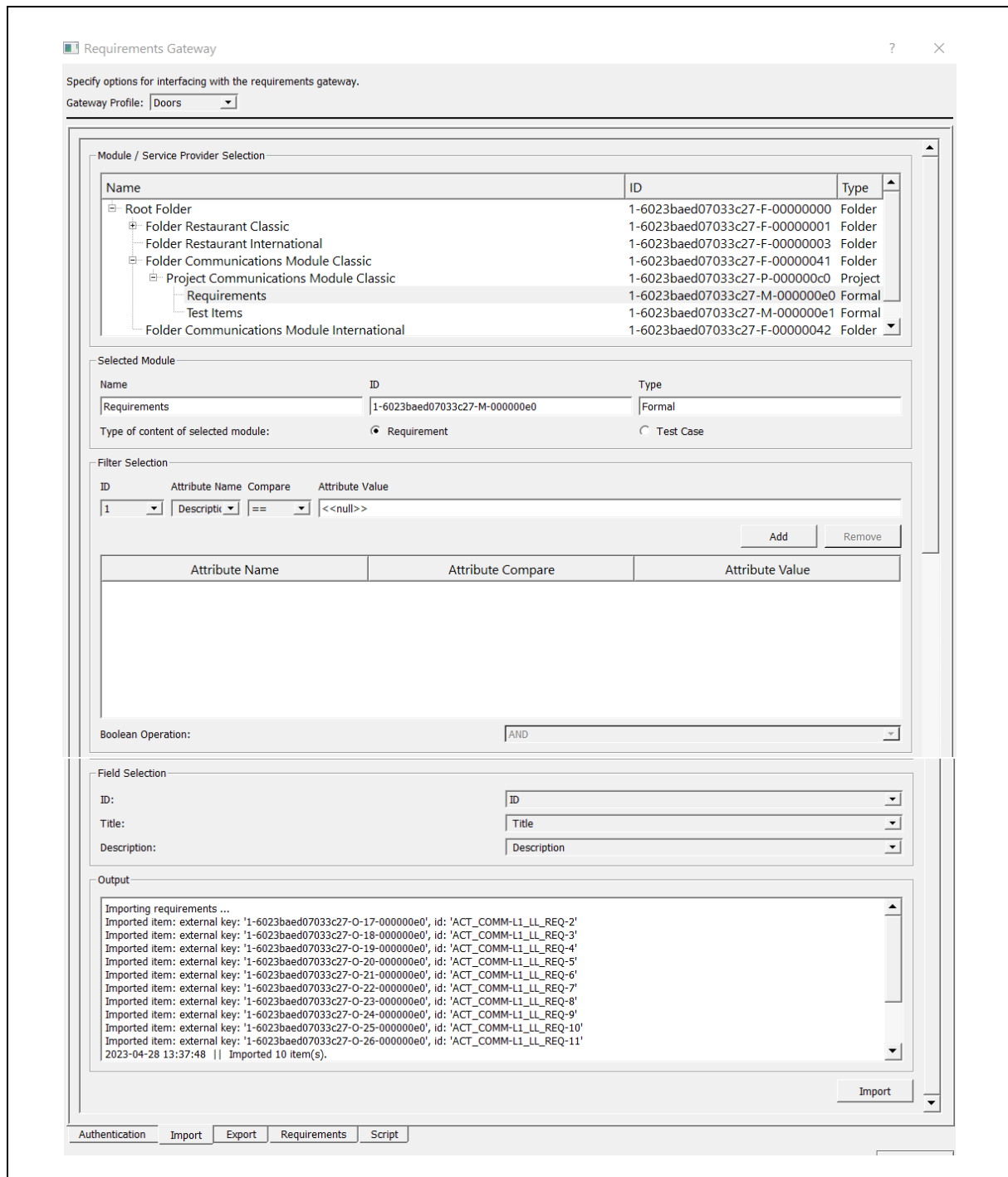


Figure 9-Select Requirements to be import from the server.

VectorCAST will now fetch the requirements selected from the server and import them into VectorCAST. A listing of the requirements will be shown after clicking on the **Requirements** tab as seen in Figure 10.

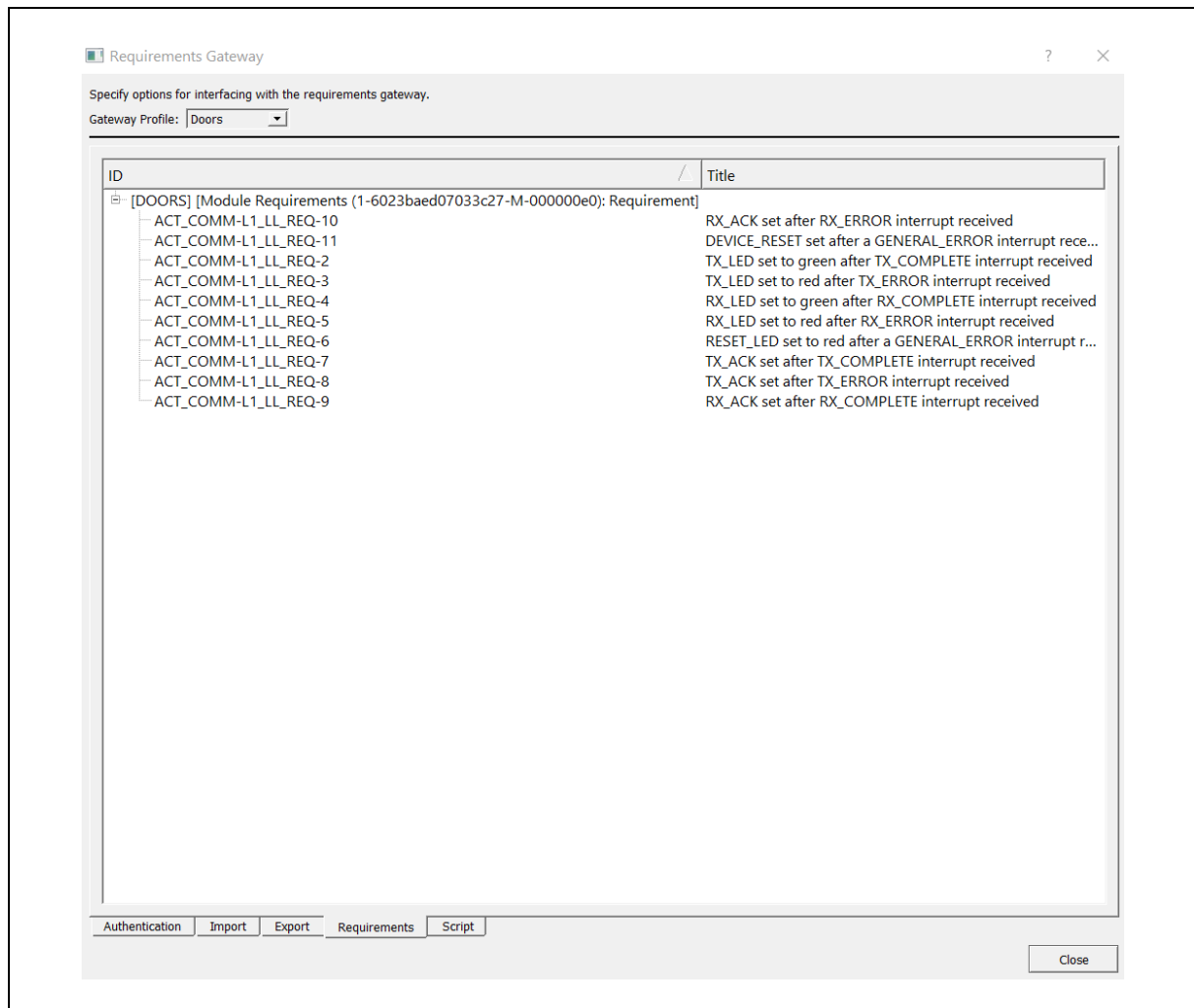


Figure 10-Import Requirements from the server

2.4 Use Case: Link Requirements to Test Cases

To link requirements to a test case, first open a unit test environment and select a test case. Select the **Requirements** button as shown in Figure 11.

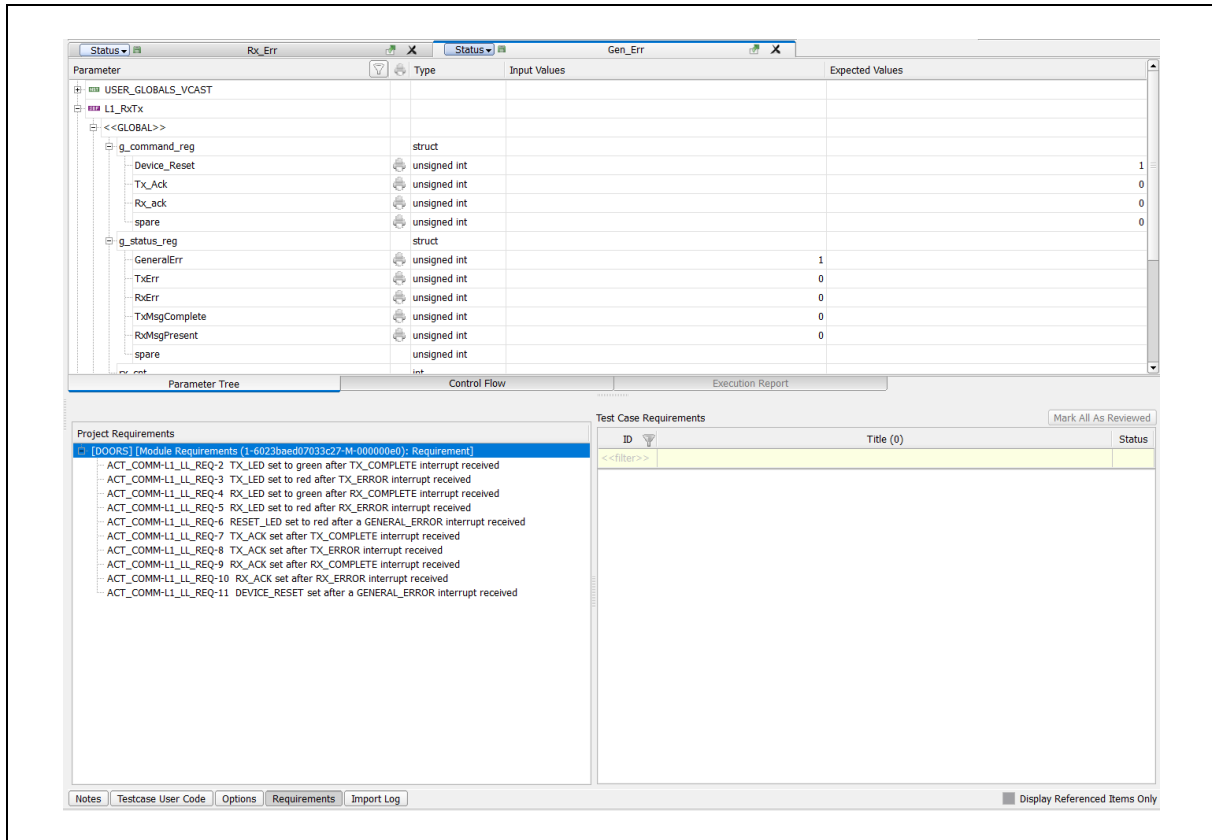


Figure 11-Select Requirements in the Unit Test Environment

In the Project Requirements window, you will see the list of imported requirements from DOORS. To link a requirement to the test case, double click on the requirement. This will add the requirement to the **Test Case Requirements** as seen in Figure 12.

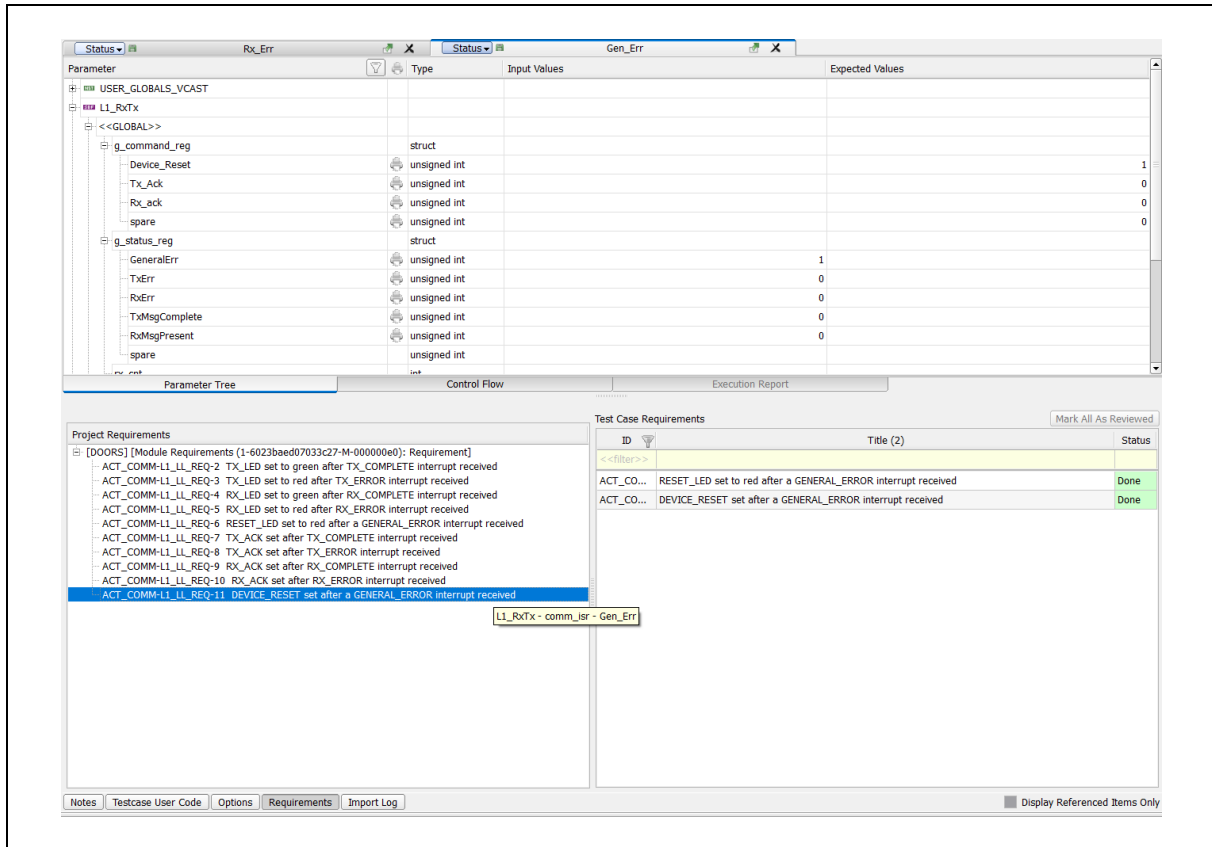


Figure 12-Link a Requirement to a Unit Test Case

At this point, execute the test case to prepare for exporting the requirements as shown in Figure 13.

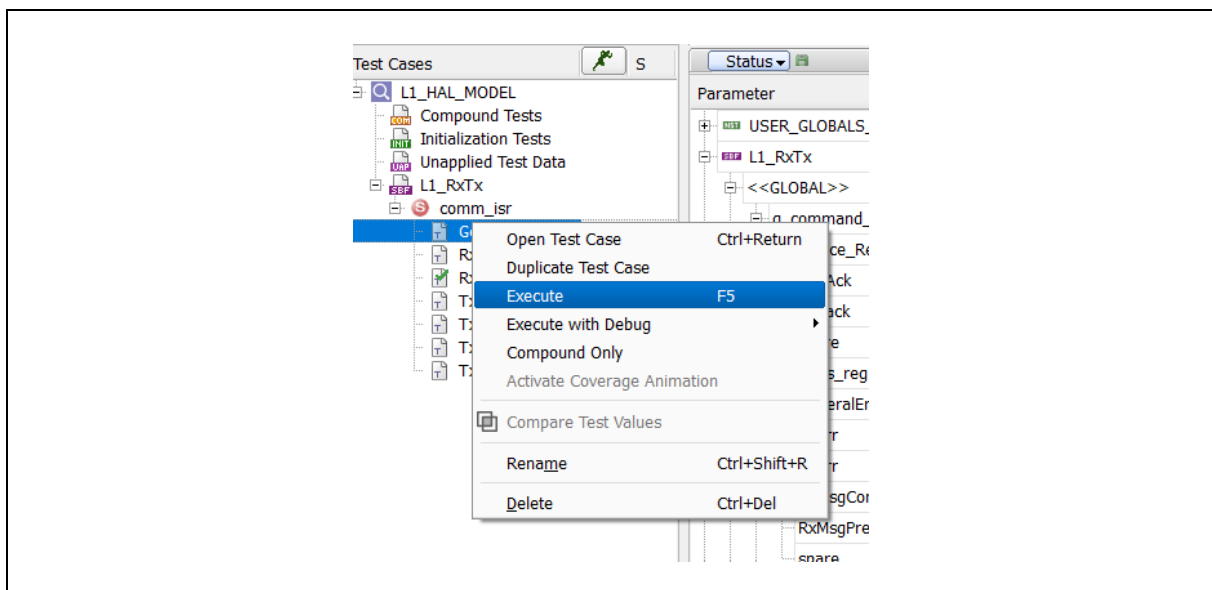


Figure 13-Execute the Unit Test Case

Once the test case is executed (see Figure 14), you are ready to perform the export.

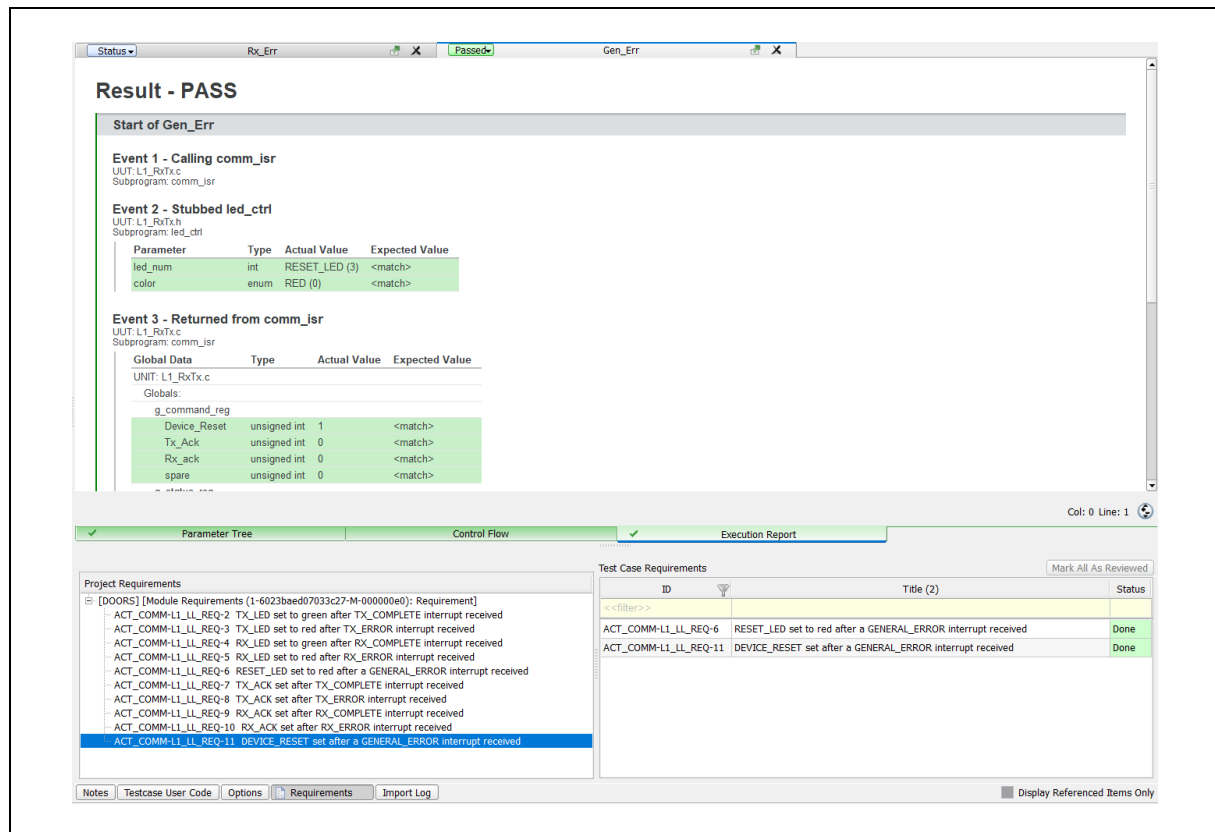


Figure 14-Unit Test Case After Execution

2.5 Use Case: Export Requirements

To export the requirements, first Select **Tools => Requirements Gateway => Doors** from the Menu Bar as shown in Figure 5. The **Requirements Gateway** dialog will open with **Doors** as the selected **Gateway Profile**. Select the **Export** tab at the bottom as shown in Figure 15. Once the Export tab loads, it will automatically connect and use the settings previously selected during **Authentication** phase.

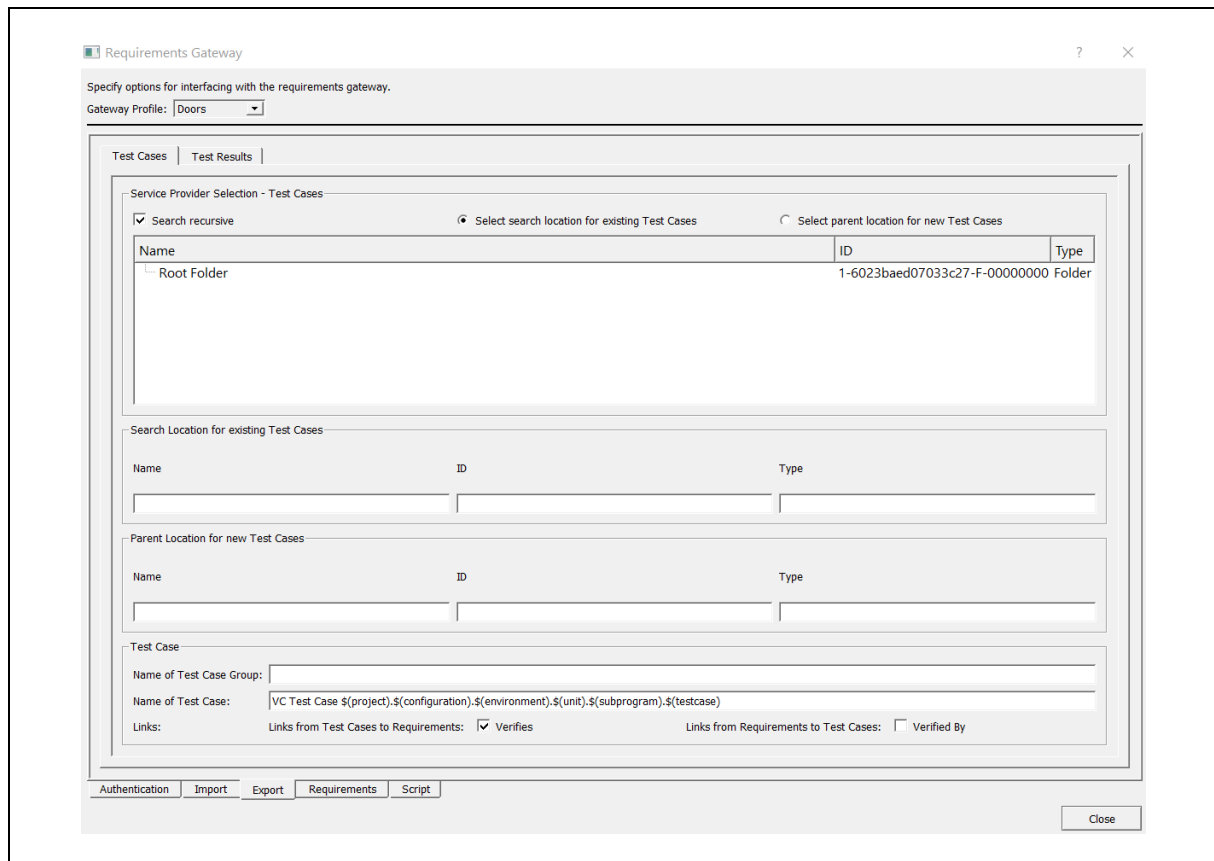


Figure 15-Select the Export Tab

There are two pages for the **Export** tab; **Test Cases** and **Test Results**.

The Test Cases tab is opened by default. The search for existing test cases is used for checking if test cases are duplicates, this is only necessary the first-time test cases are exported. Select either a Module, Folder or Project. Selecting a Folder or Modules starts a search for Test Cases on the direct children of the selection, unless “Search recursive” is checked then it will recursively search all child modules.

Select a Module to have Test Cases created in that Module. Select a Folder or Project to use the naming policy defined in Name of Test Case Group (\$environment), etc.) for adding Test Cases to one or more Modules.

To export test cases, click on **Select the search location for existing Test Cases** button and choose the desired location displayed in the window. Then click on the **Select parent location for new Test Cases** button and choose the location accordingly.

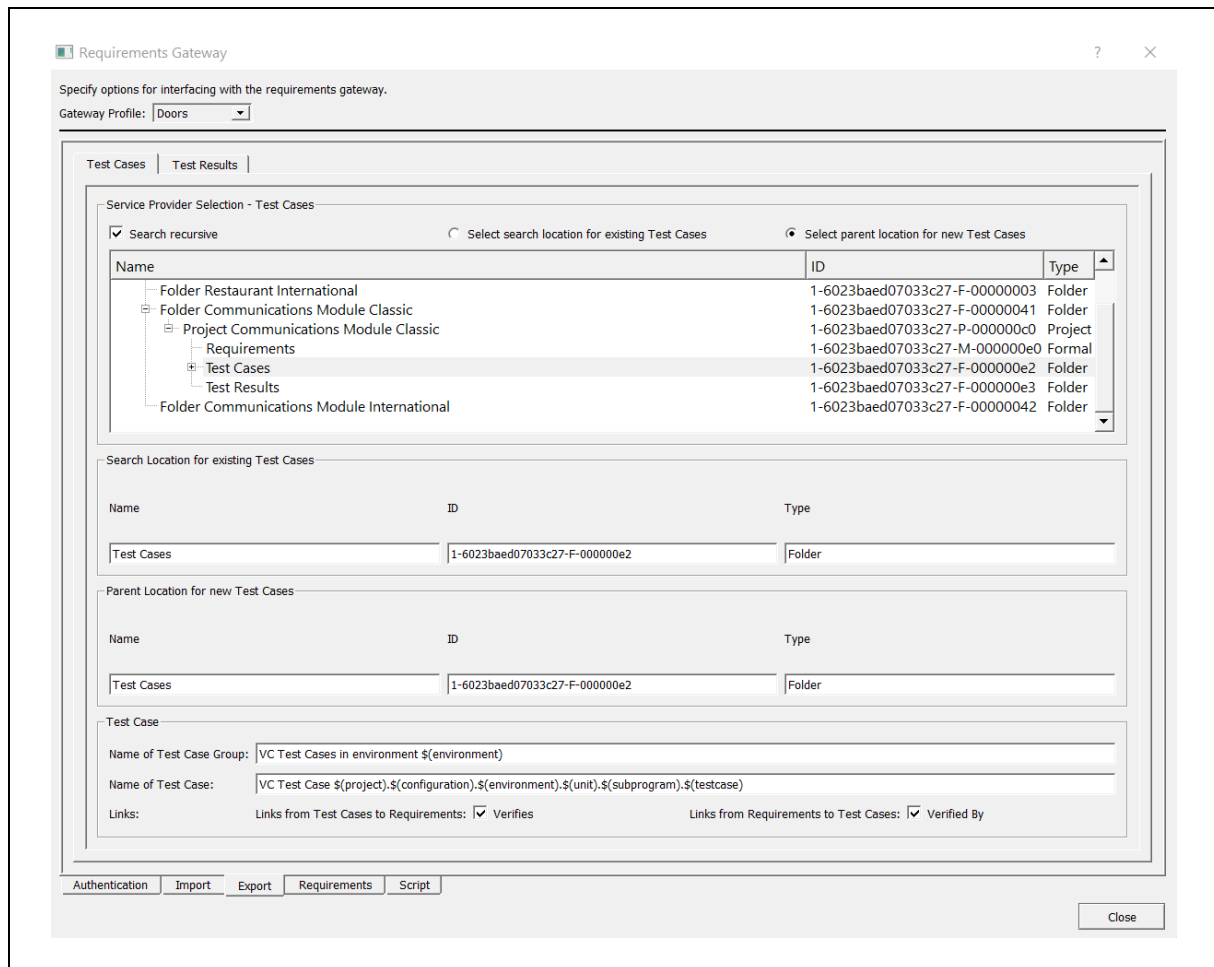


Figure 16-Export Test Cases

Links are directional and added to the Link Module in the same Folder as the source of a relationship (from source to target). Users must select one directional relationship between Test Cases and Requirements or both on export.

To customize the following variable names, edit the file **userGlobals.py** located at `%VECTORCAST_DIR%\python\vector\lib\rgw\plugins\doors\common` at the start of a project.

```
g_str_export_tcase_req_rel_name = "Verifies"
```

```
g_str_export_req_tcase_rel_name = "Verified By"
```

To export the test results, click on the **Test Results** tab as seen in Figure 17.

Requirements Gateway

Specify options for interfacing with the requirements gateway.

Gateway Profile: Doors

Test Cases | Test Results

Parent Location for new Test Results

Name	ID	Type
------	----	------

☒ Test Result

Name of Test Result Group:

Name of Test Result:

Release Identifier:

Build Identifier:

Links: Links from Test Results to Test Cases: ☒ Runs Links from Test Cases to Test Results: ☐ Run By

☐ Keep only the latest Test Results: no history of Test Results.

Output

Figure 17-Export Test Results Tab

The search for existing test runs is used for checking if test runs are duplicates. Select either a Module, Folder or Project. Selecting a Folder or Modules starts a search for Test Runs on the direct children of the selection, unless “Search recursive” is checked then it will recursively search all child modules.

Select a Module to have Test Runs created in that Module. Select a Folder or Project to use the naming policy defined in Name of Test Run Group (\$(environment), etc.) for adding Test Runs to one or more Modules.

To export test results, click on **Select the search location for existing Test Runs** button and choose the desired location displayed in the window. Then click on the **Select parent location for new Test Runs** button and choose the location accordingly.

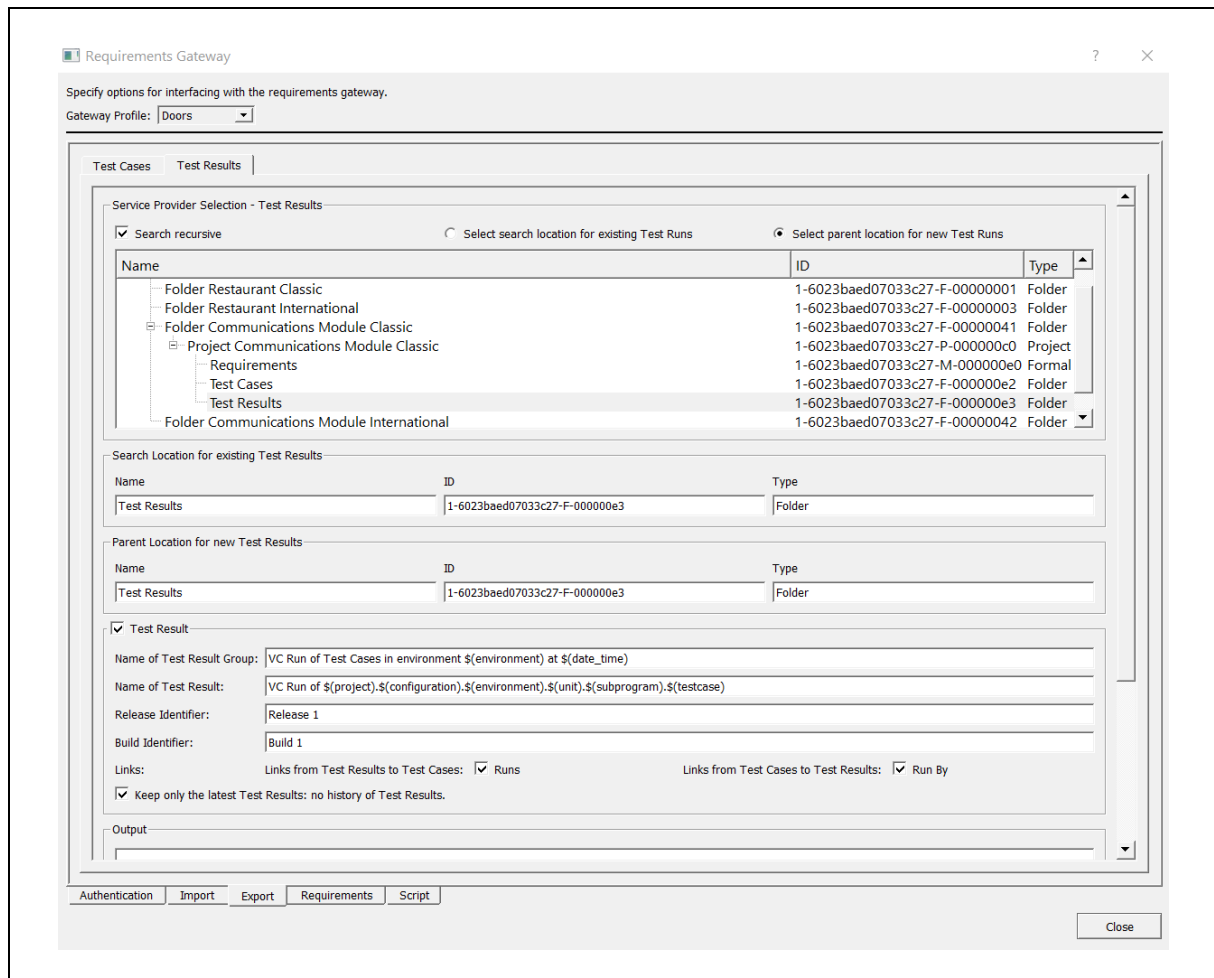


Figure 18-Export Test Results

Export only the Test Cases but not Test Results by unchecking Test Result.

The checkbox for Keep only latest Test Result: no history of Test Results, determines if an existing Test Case's Result gets overridden or always created to preserve the Test History.

NOTE: If initially deselected and then later selected means that the Test History to a Test Case will be deleted!

For the explanation about links refer to the corresponding section given for Test Cases.

To customize the following variable names, edit the file **userGlobals.py** located at `%VECTORCAST_DIR%\python\vector\lib\rgw\plugins\doors\common` at the start of a project.

```
g_str_export_tresult_tcase_rel_name = "Runs"
```

```
g_str_export_tcase_tresult_rel_name = "Run By"
```

Once all the information is entered, click Preview to see what is being exported.

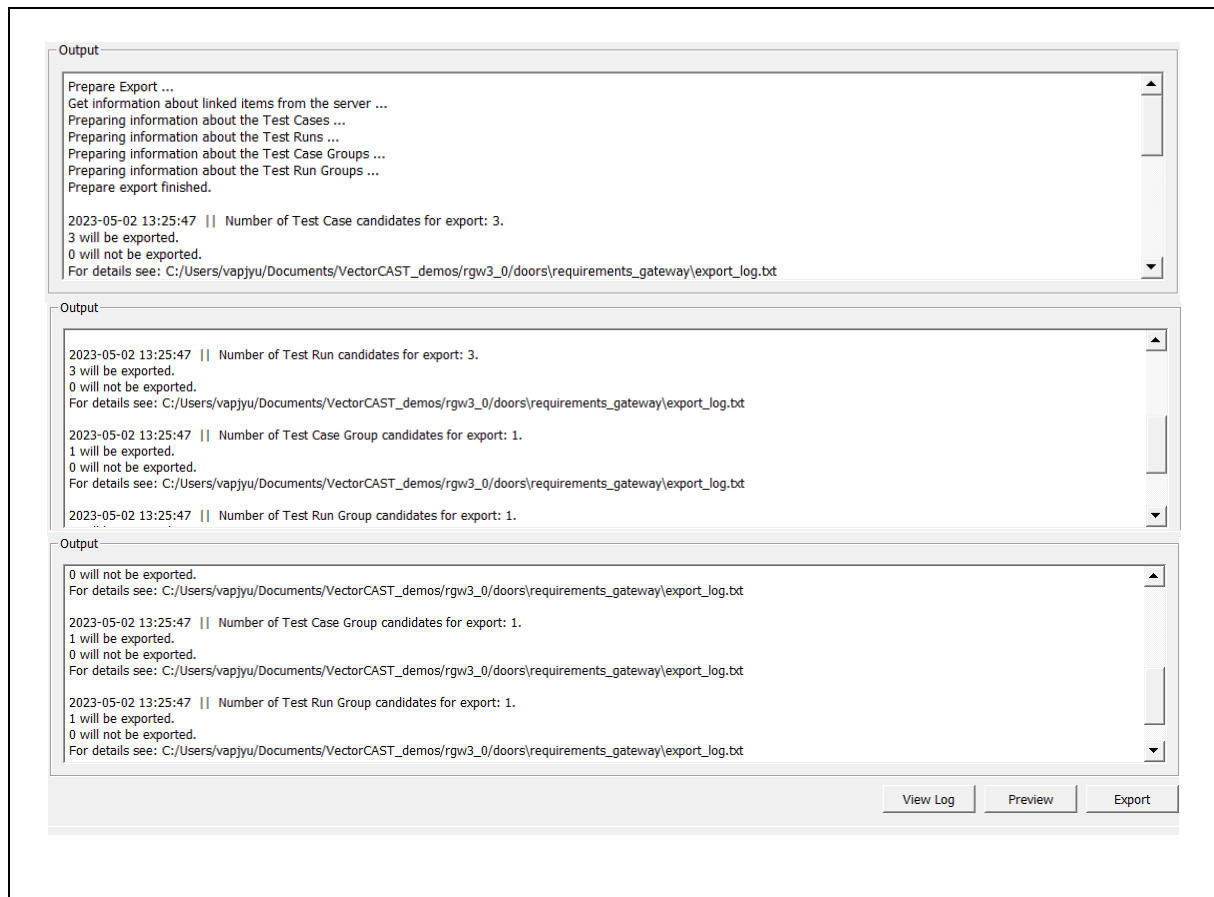


Figure 19-Preview Export

Once satisfied with the preview, click the **Export** button, see Figure 20.

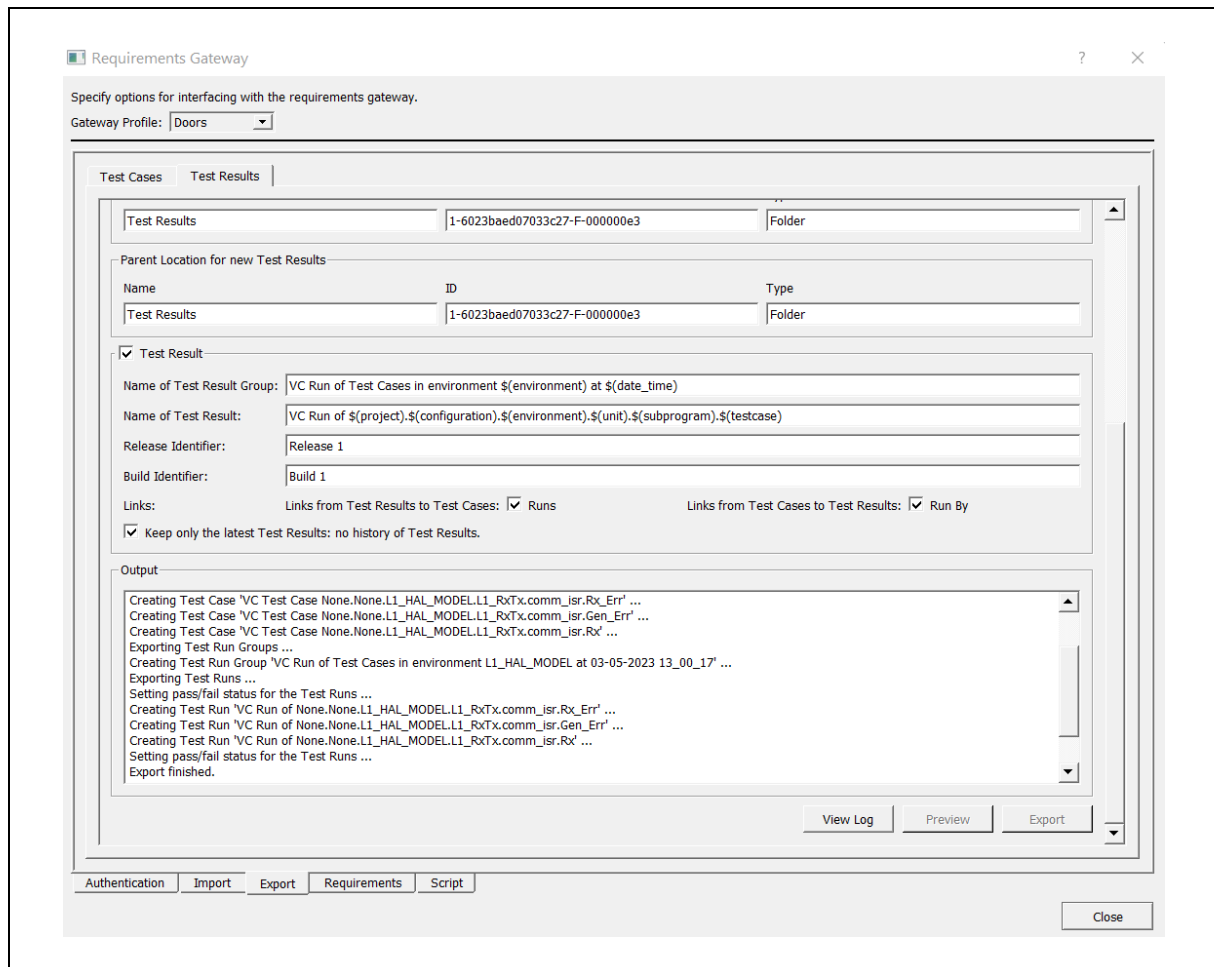
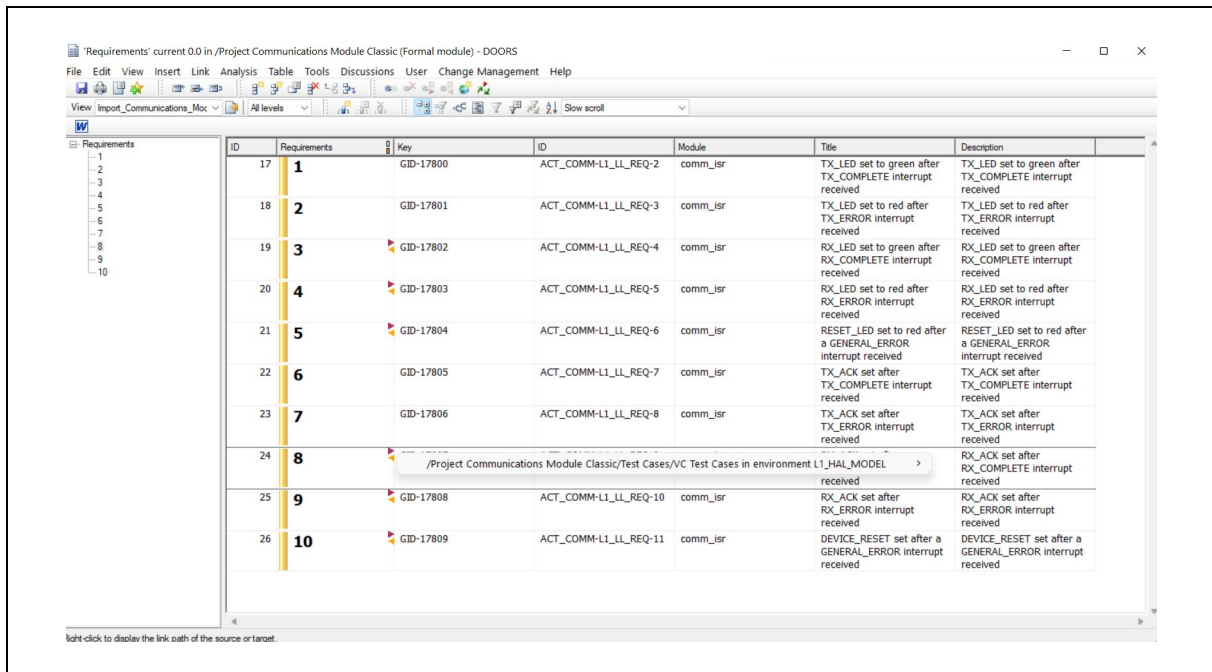


Figure 20-Export Results

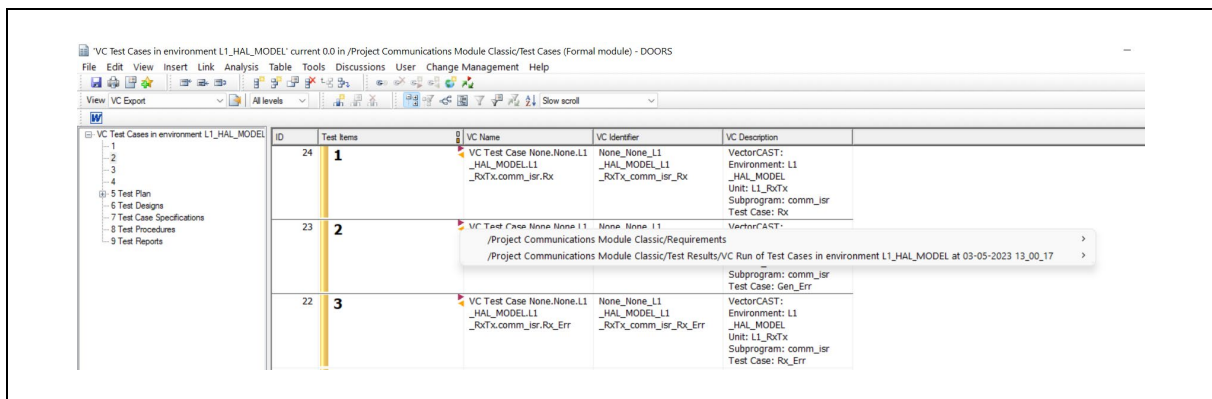
2.6 Use Case: Review Updated Requirements



The screenshot shows the DOORS application interface with a table of requirements. The table has columns for ID, Requirements, Key, ID, Module, Title, and Description. The requirements are numbered 1 through 10. Requirement 8 is highlighted, and a tooltip shows the link path: /Project Communications Module Classic/Test Cases/VC Test Cases in environment L1_HAL_MODEL.

ID	Requirements	Key	ID	Module	Title	Description
17	1	GID-17800	ACT_COMM-L1_LL_REQ-2	comm_isr	TX_LED set to green after TX_COMPLETE interrupt received	TX_LED set to green after TX_COMPLETE interrupt received
18	2	GID-17801	ACT_COMM-L1_LL_REQ-3	comm_isr	TX_LED set to red after TX_ERROR interrupt received	TX_LED set to red after TX_ERROR interrupt received
19	3	GID-17802	ACT_COMM-L1_LL_REQ-4	comm_isr	RX_LED set to green after RX_COMPLETE interrupt received	RX_LED set to green after RX_COMPLETE interrupt received
20	4	GID-17803	ACT_COMM-L1_LL_REQ-5	comm_isr	RX_LED set to red after RX_ERROR interrupt received	RX_LED set to red after RX_ERROR interrupt received
21	5	GID-17804	ACT_COMM-L1_LL_REQ-6	comm_isr	RESET_LED set to red after a GENERAL_ERROR interrupt received	RESET_LED set to red after a GENERAL_ERROR interrupt received
22	6	GID-17805	ACT_COMM-L1_LL_REQ-7	comm_isr	TX_ACK set after TX_COMPLETE interrupt received	TX_ACK set after TX_COMPLETE interrupt received
23	7	GID-17806	ACT_COMM-L1_LL_REQ-8	comm_isr	TX_ACK set after TX_ERROR interrupt received	TX_ACK set after TX_ERROR interrupt received
24	8					RX_ACK set after RX_COMPLETE interrupt received
25	9	GID-17808	ACT_COMM-L1_LL_REQ-10	comm_isr	RX_ACK set after RX_ERROR interrupt received	RX_ACK set after RX_ERROR interrupt received
26	10	GID-17809	ACT_COMM-L1_LL_REQ-11	comm_isr	DEVICE_RESET set after a GENERAL_ERROR interrupt received	DEVICE_RESET set after a GENERAL_ERROR interrupt received

Figure 21-Requirements with Links to Test Cases



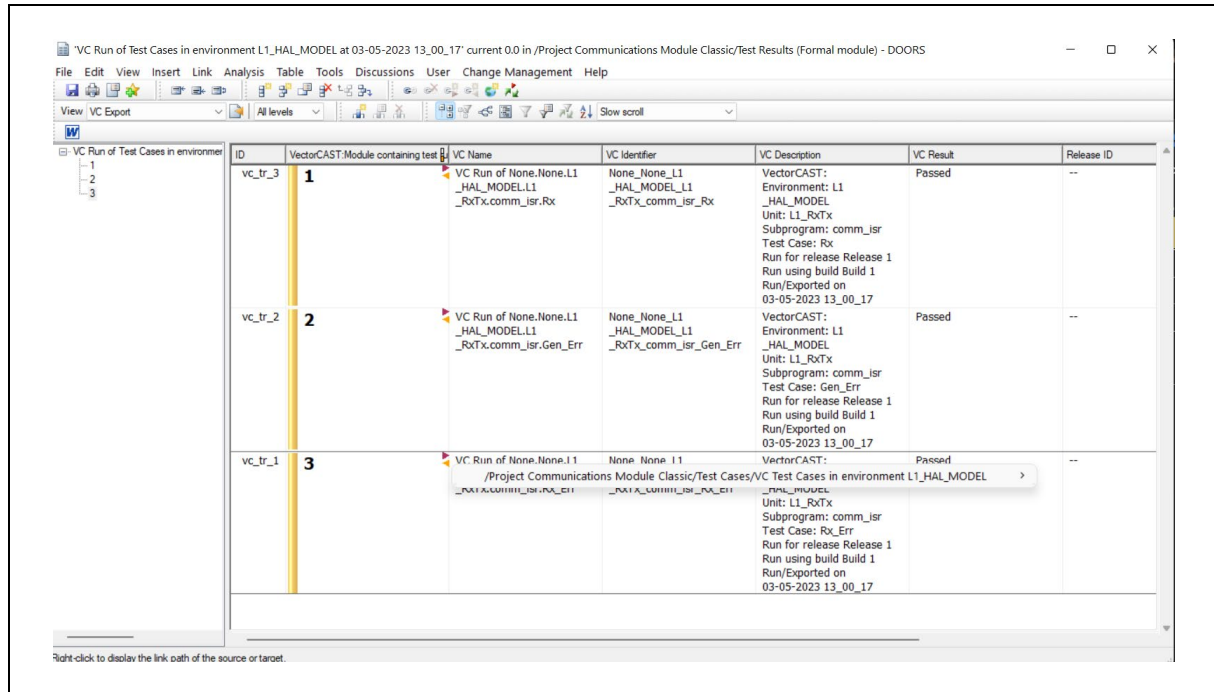
The screenshot shows the DOORS application interface with a table of test cases. The table has columns for ID, Test Item, VC Name, VC Identifier, and VC Description. The test cases are numbered 1 through 3. Test case 2 is highlighted, and a tooltip shows the link path: /Project Communications Module Classic/Test Results/VC Run of Test Cases in environment L1_HAL_MODEL at 03-05-2023 13_00_17.

ID	Test Item	VC Name	VC Identifier	VC Description
24	1	VC Test Case None.None.L1_HAL_MODEL.L1_RxTx.comm_isr.Rx	None_None.L1_HAL_MODEL.L1_RxTx.comm_isr.Rx	VectorCAST: Environment: L1_HAL_MODEL Unit: L1_RxTx Subprogram: comm_isr Test Case: Rx
23	2	VC Test Case None.None.L1_HAL_MODEL.L1_RxTx.comm_isr.Rx_Err	None_None.L1_HAL_MODEL.L1_RxTx.comm_isr.Rx_Err	VectorCAST: Subprogram: comm_isr Test Case: Gen_Err
22	3	VC Test Case None.None.L1_HAL_MODEL.L1_RxTx.comm_isr.Rx_Err	None_None.L1_HAL_MODEL.L1_RxTx.comm_isr.Rx_Err	VectorCAST: Environment: L1_HAL_MODEL Unit: L1_RxTx Subprogram: comm_isr Test Case: Rx_Err

Figure 22-Test Case in DOORS with Links to Test Runs

Note: The columns are VectorCAST attributes which can be added via **Insert -> Columns** in the tool bar.

- VC Name
- VC Identifier
- VC Description



ID	VectorCAST Module containing test	VC Name	VC Identifier	VC Description	VC Result	Release ID
vc_tr_3	1	VC Run of None.None.L1 _HAL_MODEL.L1 _RxTx.comm_isr_Rx	None_None.L1 _HAL_MODEL.L1 _RxTx.comm_isr_Rx	VectorCAST: Environment: L1 _HAL_MODEL Unit: L1_RxTx Subprogram: comm_isr Test Case: Rx Run for release Release 1 Run using build Build 1 Run/Exported on 03-05-2023 13_00_17	Passed	--
vc_tr_2	2	VC Run of None.None.L1 _HAL_MODEL.L1 _RxTx.comm_isr_Gen_Err	None_None.L1 _HAL_MODEL.L1 _RxTx.comm_isr_Gen_Err	VectorCAST: Environment: L1 _HAL_MODEL Unit: L1_RxTx Subprogram: comm_isr Test Case: Gen_Err Run for release Release 1 Run using build Build 1 Run/Exported on 03-05-2023 13_00_17	Passed	--
vc_tr_1	3	VC Run of None.None.L1 _HAL_MODEL.L1 _RxTx.comm_isr_Rx	None_None.L1 _HAL_MODEL.L1 _RxTx.comm_isr_Rx	VectorCAST: Environment: L1 _HAL_MODEL Unit: L1_RxTx Subprogram: comm_isr Test Case: Rx_Err Run for release Release 1 Run using build Build 1 Run/Exported on 03-05-2023 13_00_17	Passed	--

Figure 23-Test Runs in DOORS with Links to Test Cases.

Note: The columns are VectorCAST attributes which can be added via **Insert -> Columns** in the tool bar.

- VC Name
- VC Identifier
- VC Description
- VC Result
- Release ID (Optional)
- Build ID (Optional)

To customize the VectorCAST attributes, edit the file **userGlobals.py** located at **%VECTORCAST_DIR%\python\vector\lib\rgw\plugins\doors\common** at the start of a project.

These values can be modified at the start of a project.

```
ATTR_VC_ID = "VC Identifier"
```

```
ATTR_VC_NAME = "VC Name"
```

```
ATTR_VC_DESCRIPTION = "VC Description"
```

```
ATTR_VC_RESULT = "VC Result"
```

```
ATTR_RELEASE_ID = "Release ID"
```

```
ATTR_BUILD_ID = "Build ID"
```

```
USE_RELEASE_ID = True
```

```
USE_BUILD_ID = True
```

3 Additional Resources

[IBM Engineering Requirements Management DOORS](#)

4 Trademarks

All mentioned names are either registered or unregistered trademarks of their respective owners.

5 Contacts

For a full list with all Vector locations and addresses worldwide, please visit <http://vector.com/contact/>.