



File Server-SMB2 Test Suite

Version 2.0

Tutorial

February 2020

[Abstract](#)

Training in File Server-SMB2 Test Suite Technology

Author

Greg Gille | Microsoft Senior Content Developer

Microsoft Protocols Test Team

Table of Contents

1.0 Introduction.....	3
1.1 Test Suite Training Audience.....	3
1.2 Goals of the Test Suite Training.....	3
1.3 Training Session Details	3
2.0 Glossary Definitions	4
3.0 Concepts	6
3.1 What You Will Learn.....	6
3.2 Protocol Communications.....	7
3.3 Test Environment Architecture.....	8
4.0 Configuring the Test Suite	10
4.1 Configure the Test Environment with the PTM Wizard.....	11
4.2 Configure the Test Environment by Loading a Profile	21
5.0 Running the Test Suite Test Cases.....	23
5.1 Saving a Profile	26
5.2 Executing Test Cases From the Command Line	27
6.0 Analyzing the Test Results Data	28
6.1 Test Results Output Status Indicators	28
6.2 Evaluating PTM Test Result Data.....	31
6.3 Common Failures	37
6.4 Analyzing Test Results with Message Analyzer.....	38
7.0 More Information	39
7.1 Resources.....	39

Table of Figures

Figure 1	SMB2 protocol messages example : Setting up an SMB2 share connection	7
Figure 2.	File Server-SMB2 Test Suite : Network test environment.....	9
Figure 3.	Test Environment : Test Cases communication path	10
Figure 4.	Protocol Test Manager : Select Test Suite tab Launching the Configure Wizard	12
Figure 5.	Protocol Test Manager : Reviewing the Test Suite protocol family.....	13
Figure 6.	Protocol Test Manager : Reviewing the Domain Test Environment.....	14
Figure 7.	Protocol Test Manager : Selecting a configuration method	15
Figure 8.	Protocol Test Manager : Verifying Auto-detect input information.....	16
Figure 9.	Protocol Test Manager : Reviewing detection results	17
Figure 10.	Protocol Test Manager : Filtering-Selecting test cases	18
Figure 11.	Protocol Test Manager : Verifying the Common property values.....	19
Figure 12.	Protocol Test Manager : Setting up for network capture of ETL data	20
Figure 13.	Protocol Test Manager : Configuring the SUT control adapter	21
Figure 14.	Protocol Test Manager : Loading an existing Profile	22
Figure 15.	Protocol Test Manager: Confirming the validity of Profile test cases	23
Figure 16.	Protocol Test Manager : Starting Test Case execution	25
Figure 17.	Protocol Test Manager : Test Case context menu commands	26
Figure 18.	Protocol Test Manager : Saving a test cases Profile.....	27
Figure 19.	Protocol Test Manager : SMB2 operation test categories Error and StandardOut results.....	31
Figure 20.	Protocol Test Manager : Test results grouped by Outcome	32
Figure 21.	Protocol Test Manager : Categorized display of Test Cases for SMB30 operations.....	33
Figure 22.	Protocol Test Manager : StandardOut results for a passed Test Case.....	34
Figure 23.	Protocol Test Manager : Applying Log Filtering to StandardOut test results data.....	35
Figure 24.	Protocol Test Manager : StandardOut results in separate HTML format.....	36
Figure 25.	Protocol Test Manager : ErrorStackTrace results.....	37
Figure 26.	Message Analyzer : Analysis Grid display of ETL Test Case data	39

1.0 Introduction

This Tutorial provides step-by-step instructions for connecting to and configuring the **File Server-SMB2 Test Suite (FSSTS)**, in preparation for executing predefined Test Cases that exercise various functions of the SMB2 protocol in file sharing scenarios. Following test execution, you will perform some basic analysis of test results. You will be provided with the test environment assets that facilitate the configuration and execution of tests.

The management of test environment configuration, **Test Case** execution, and test results analysis are all achieved with the use of a graphical user interface (GUI) tool known as the **Protocol Test Manager (PTM)**. This Tutorial will show you how to use the PTM to accomplish these tasks. This tool and the File Server-SMB2 Test Suite are already installed on a **Driver computer** where you will conduct the testing, and to which you will have access via the **Remote Desktop Protocol (RDP)**.

The audience for this Tutorial/Lab session is described just ahead, as are the goals of this training and the details of session organization.

1.1 Test Suite Training Audience

The primary audience for the FSSTS training is software developers who have little or no experience with running protocol test suites. Other audiences can include IT professionals and others who may be interested in learning about protocol test suites.

1.2 Goals of the Test Suite Training

The primary goals of this training session are as follows:

- Learn the basics concepts of simple protocol communications.
- Obtain a preliminary understanding of the FSSTS test environment.
- Learn how to use the **Protocol Test Manager** to manage the configuration, filtering, test execution, and test results analysis features of the **FSSTS**.

1.3 Training Session Details

The organization of this Lab session and the time allocated for the session tasks are indicated in the table follows:

Table 1. Training session organization

Session Task	Allocation (minutes)
Review glossary terms and conceptual material	20
Configure Test Suite and run Test Cases	25
Analyze test results and ask questions	45
Total time	90

2.0 Glossary Definitions

The following list defines the important terms that are frequently used in this Tutorial.

Authentication token — a security device that enables a user or application to gain authorized access to secured resources, such as a network service.

Driver computer — a computer on which the [File Server-SMB2 Test Suite \(FSSTS\)](#) is installed and configured. After installation, you can run the preconfigured test cases that are part of FSSTS. In the test environment, this computer typically runs a Windows client operating system.

File Server-SMB2 Test Suite (FSSTS) — a set of preconfigured, software-coded **Test Cases** that exercise features of the **SMB2 protocol** and other protocols that are associated with file services. Contains the framework for configuring the test environment, executing tests of File Server features, and facilities for analyzing test results.

Implementation — a coded representation of the functions of a protocol that contains the messages and formats required to establish communications between computers across a network, for the purpose of servicing application functions via sending client requests and replying with server responses.

Isolated network — for testing protocols with the FSSTS, this a network that is disconnected from the Internet, uses an isolated hub or switch, and is not part of a production network of any kind.

Message — a packet of data that sends instructions or other information in the form of a request or a response, from one computer to another.

Microsoft Message Analyzer — a network tracing and analysis tool that enables you to capture, display, and analyze protocol messaging traffic, and to trace and assess system events, and Windows component events. It also provides the capability to retrieve, aggregate, and analyze data from one or more saved traces.

Optionally used in this Tutorial to analyze event trace log (ETL) data, as generated during Test Case execution.

Profile — a file generated by the PTM that represents a configuration of Test Cases that you optionally create and store in a directory location following Test Case execution. A Profile acts as a template that enables the repetitive reapplication of an identical set of Test Cases against a common SUT environment.

Protocol — a set of rules or procedures that define how data is transmitted between computers. To achieve a successful interchange of information, a protocol establishes the structure of the information, the transmission method, and how the sending and receiving nodes process the information. The functions of a protocol are typically expressed as a set of message packets, which in turn reflect the protocol's rules.

Protocol Test Manager (PTM) — a graphic user interface that provides the facilities for performing all the tasks associated with using the Test Suite, which includes the following:

- Detecting the system under test (SUT) configuration and capabilities.
- Creating a default set of Test Cases, based on the assessed SUT environment.
- Optionally reconfiguring the Test Case selections.
- Running the Test Cases.
- Analyzing the test results.

See [Configuring the Test Suite](#) for further information.

Provisioning — a term that is used to describe the process of setting up computers and specific resources for a predetermined purpose. For example, a company might have need of an SMTP mail server or an IIS web server for their business. Companies such as Microsoft and others will set up, customize, and maintain those systems in the Azure hosting space for the customer in an agreed upon contract. Providing such resources is commonly known as provisioning.

Note that the **Driver computer** and **SUT computer** are virtual machines (VMs) that may be provisioned by Azure services or by other resources, for the purposes of this lab session.

Remote Desktop Protocol (RDP) — the **protocol** that facilitates the well-known Remote Desktop Connection application that enables users to connect from their local computer to a specified remote computer, so that users may access the resources of such a computer.

Server Message Block (SMB2) protocol — the SMB2 protocol is typically used by clients to request file and print services from a network file server. The protocol enables you to establish a server connection, an authentication context for the connection, and to thereafter request access to files, shares, and printers.

The SMB2 protocol is the mechanism that enables you to open, read, modify, and close files on a target server; to query and apply attributes to files or volumes on a target server; to moderate the shares and users; and to simultaneously open files. It also supports the creation of symbolic links, which for example enables you to create a link to a file or directory from another file in a different directory location.

SMB2 client — typically a computer that uses a particular SMB2 dialect or later to make SMB requests to an SMB file server to obtain access to file system resources for reading and/or writing data. The SMB2 client sends and receives data to/from the SMB2 server via message packets of the SMB protocol.

Note In the context of this training, the SMB2 client is the **Driver computer** and the SMB2 server is the **SUT computer**.

SMB2 server — a computer that receives and services requests from an SMB2 client via the message packets of the SMB2 protocol.

SMB2 dialect — the SMB protocol comes in several versions that are known as dialects, for example, SMB v2.0.2, 2.1, 3.0, 3.0.2, 3.1.1, and so on.

System under test (SUT) computer — a computer that hosts the system against which the pre-defined Test Cases are to be run by the **FSSTS** that is installed on the **Driver computer**. Typically, the

FSSTS tests an implementation of the **SMB2 protocol**, which can be either a proprietary, developed SMB2 implementation or the SMB2 service that runs on the SUT computer by default.

For purposes of this training and the preconfigured set up that is used, the Microsoft SMB2 service on the SUT computer serves as the underlying implementation being tested. In the test environment for this lab training, this computer typically runs a Microsoft server operating system.

Test Case — an executable application hosted by the Test Suite that is designed to test unique aspects of File Server features that use the **SMB2 protocol** within the context of an SMB2 client and SMB2 server communication session. **Note:** an **FSSTS** installation can contain thousands of Test Cases.

Virtual machine (VM) — typically an emulation of a computer system that has a computer architecture and provides the functionality of a physical computer, but its implementation is software based and has no physical component, other than a physical computer on which the VM is hosted.

3.0 Concepts

This section briefly describes the major concepts with which you will become familiar during the course of the **FSSTS** Lab Session Tutorial. The material begins with the basic concepts of protocol communication and descriptions of the test environment with which you will be working, as indicated in section [3.1 What You Will Learn](#), directly ahead. This section also points you to other sections of this Tutorial that show you how to use the Protocol Test Manager to configure the Test Suite, select and run the Test Cases, and analyze the test results.



Note

If you have not already done so, you should read the preceding glossary definitions to obtain a brief overview of pervasive concepts in this Tutorial through an understanding of terms.

3.1 What You Will Learn

This section provides an overview of the scope of this Tutorial, in terms of the specific things that you will be learning, as follows.

Protocol Communications — introduces the basics of protocol communications by giving a hypothetical example of the initial exchange of **SMB2 protocol** messages that are used to set up an SMB2 session.

Test Environment Architecture — shows a basic network diagram that is similar to the test environment in which you will be working, provides a description of its components, and shows a graphic representation of the Test Cases communication path between the **Driver** and **SUT** computers.

Configuring the Test Suite — shows you how to use the **Protocol Test Manager (PTM)**, the primary user interface that you will utilize to manage test environment configuration on the Driver computer.

Running the Test Suite Test Cases — shows you how to use the PTM to manage execution of **File Server-SMB2 Test Suite (FSSTS)** Test Cases on the **SUT computer**, initiated from the **Driver computer**.

Analyzing the Test Results Data — shows you how to use the PTM to manage analysis of the test results.



Note

When you have completed the procedures in the latter three sections above, you will be finished with this Tutorial.

3.2 Protocol Communications

This section provides an example of protocol communications between an **SMB2 client** and an **SMB2 server**. It shows how several types of messages of the **Server Message Block (SMB2) protocol** are used to set up a session between an SMB2 client and an SMB2 server, in order to provide the client with access to a share on the server. The message exchanges basically consist of initial client/server negotiations, the client **authentication** process, and a connection request for share access.

A visual representation of the communication process in Figure 1 is accompanied by explanatory steps. The sequence of messages shown in Figure 1 are high-level representations of the types of requests and responses that are typical of setting up an SMB2 session and connecting to a server share.

The goal of this section is to provide readers with a basic sense of how the **File Server-SMB2 Test Suite** on the Driver computer (client) communicates with the SUT computer (server). For example, in the Test Suite, a **Test Case** could invoke an SMB2 method and obtain confirmation as to whether encrypted access to a server share was successful or not; another test case might verify if a symbolic link within a share is working properly. These features are actually preconfigured on the SUT computer, against which some Test Cases are run from the Test Suite on the **Driver computer**.

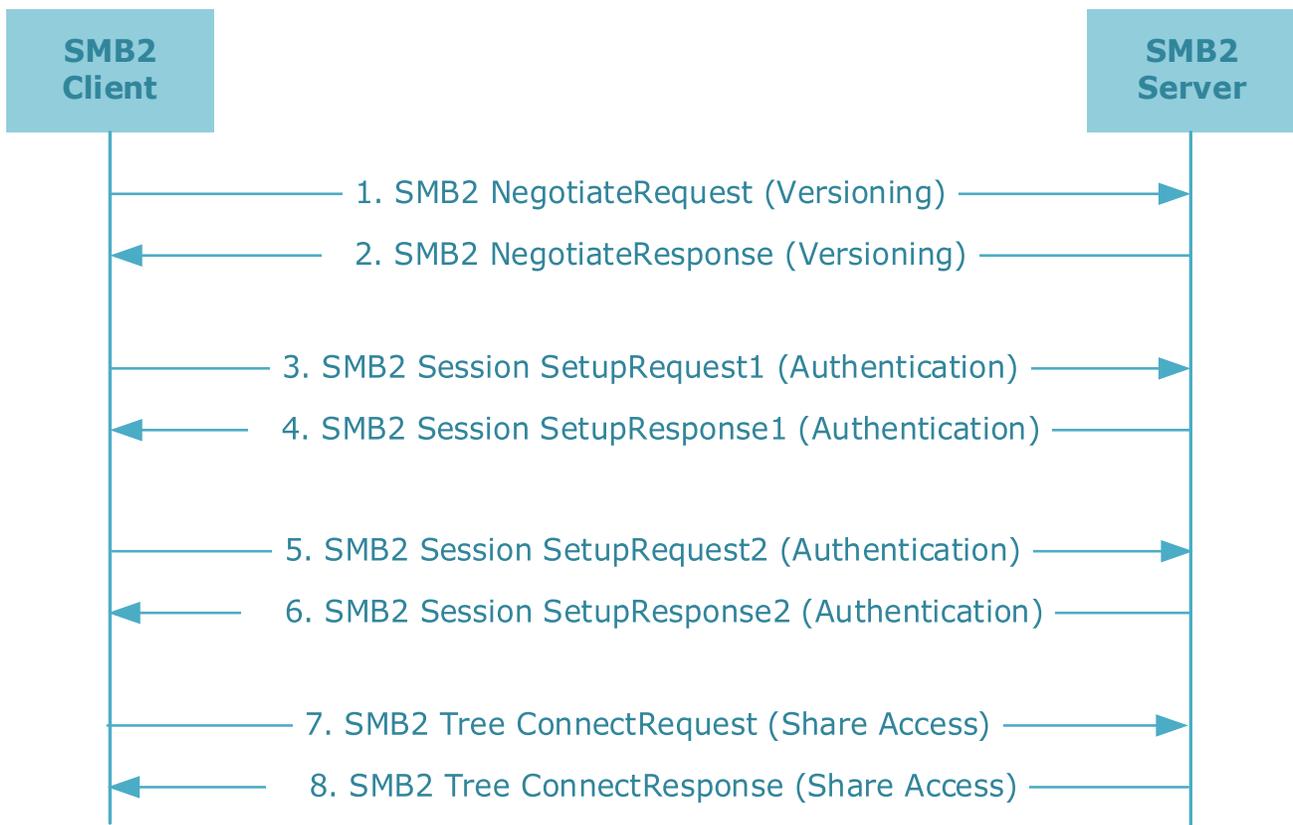


Figure 1 SMB2 protocol messages example : Setting up an SMB2 share connection

1. The SMB2 client sends a **NegotiateRequest** to the SMB server to establish the **SMB protocol** version or *dialect* in use by the client (SMB2, SMB3, and so on).
2. The SMB2 server acknowledges the client's **SMB dialect** and sends an SMB2 **NegotiateResponse** message back to the SMB2 client.
3. The SMB2 client queries a security service for an **Authentication token** and sends a **SetupRequest1** message to the SMB2 server.
4. The SMB2 server responds with a **SetupResponse1** message containing the authentication token and requests further processing from the SMB2 client.
5. The SMB2 client performs processing on the authentication token and passes it, along with a SessionId, in a **SetupRequest2** message to the SMB2 server.
6. The SMB2 server processes the authentication token and sends a **SetupResponse2** message back to the SMB2 client with a success status code.
7. The SMB2 client completes the authentication process and sends a **ConnectRequest** message to the SMB2 server containing a SessionId and the name of the share to connect to.
8. The SMB2 server enables the connection and sends a **ConnectResponse** message containing a success status code and other session information to the SMB2 client.

To offer a simplified correlation to the foregoing, when the **File Server-SMB2 Test Suite** performs tests, it begins by sending requests from the **Driver computer** and receiving responses from the **SUT computer**, in a manner that is similar to the message sequences shown in Figure 1. After a session is successfully set up, the Test Suite runs a host of preconfigured Test Cases that utilize important features and functions of the SMB2 protocol and others to assess the system under test.

The Test Environment in which this occurs is described in the sections that follow.

3.3 Test Environment Architecture

The Test Environment consists of an isolated network with a Driver computer (client) and a SUT computer (server) hosted as Azure **virtual machines** in a Domain environment. Users will access the Domain via the remote desktop protocol (RDP) with a predefined administrator name and password. The basic network configuration is shown in Figure 2, as follows:

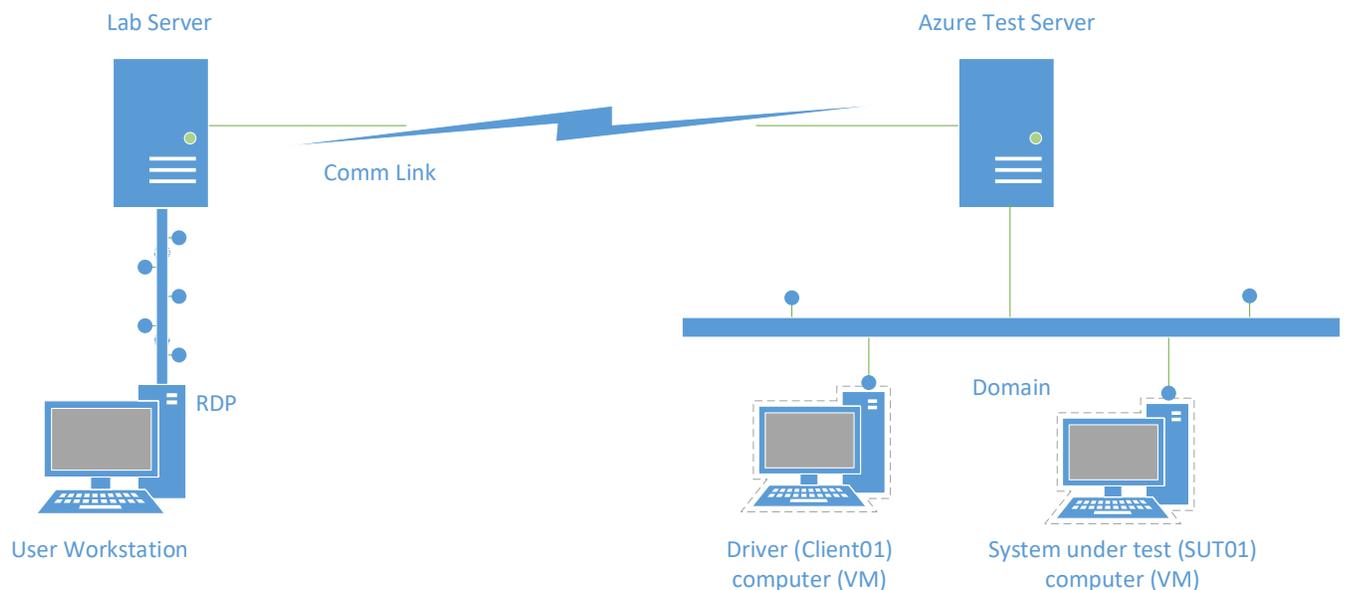


Figure 2. File Server-SMB2 Test Suite : Network test environment

The components of the File Server-SMB2 Test Suite network Test Environment are described in the list that follows:

- **User Workstation** — a user laptop, Surface, or other lab computer from which you will **RDP** into the cloud to specified virtual machines (VMs) in a Domain environment, where you will configure the File Server-SMB2 Test Suite as described in section [4.0 Configuring the Test Suite](#) for the system to be tested.
- **Remote Desktop Protocol (RDP)** — the common application you will use to connect with the Domain environment consisting of the Driver and SUT computers.
- **Lab Server** — an optional test lab server, depending on the specific lab environment in which you will connect to the internet to access the Driver and SUT computers.
- **Azure Test Server** — an optional server depending on the lab environment that will host the **VMs** configured in a Domain environment. It is setup by the Microsoft Test and/or Support teams, who will **provision** the VM computers and other resources you will be using in the Test Environment.
- **Driver computer** — a VM that hosts the **File Server-SMB2 Test Suite** that you will configure for running Test Cases against the SUT configuration.
- **SUT computer** — a VM computer that is pre-configured with the required file shares and other features to be tested by the **Test Cases** of the Test Suite.

Important

In this Test Environment, you will not be testing some proprietary SMB2 implementation on the SUT computer. Rather, the Microsoft SMB2 service that normally runs on that computer will respond to the test messages generated by the Test Cases of the **File Server-SMB2 Test Suite**, which reside on the Driver computer, as shown in the figure that follows:

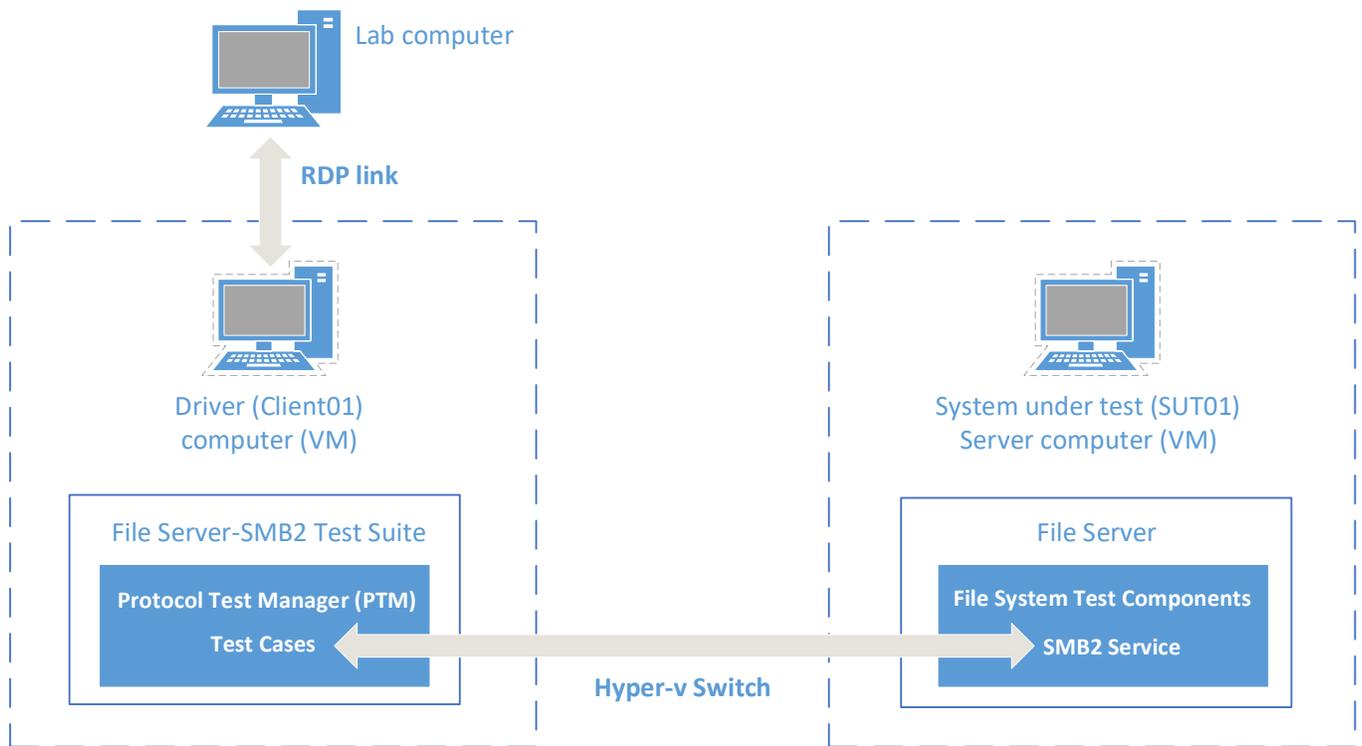


Figure 3. Test Environment : Test Cases communication path

4.0 Configuring the Test Suite

This section describes how to configure the Test Suite **Test Cases** with **Protocol Test Manager (PTM)** on the **Driver computer**. The list immediately below basically follows the steps of the PTM's **Configure Wizard**, which includes performing the following tasks after starting the PTM:

- Reviewing the execution environment (**Domain**).
- Selecting the **Auto-Detection** option for configuring the test environment.
- Reviewing the SUT detection results and validating user input information (**Target Share, User Name, Password**, etc.)
- PTM detection, inspection, and validation of the SUT environment for test readiness.
- Modifying the default output of the **Auto-Detection** process, by selecting/unselecting (filtering) Test Cases to create a unique test configuration. In this Tutorial, you will configure specific tests to create focus on a specific set of test results.
- Reviewing and optionally configuring Test Case properties.
- Selecting an SUT control adapter.

You will complete the items cited in the above list in [section 4.1](#) that follows. Thereafter, you will run the Test Cases and then perform simple analysis on test results, to obtain a basic understanding of the analysis features.

Optional Configuration Scenario

If you have already run through an execution of **PTM** Test Cases and saved a **Profile** per the procedure in section [5.1 Saving a Profile](#), you can proceed to section [4.2 Configure the Test Environment by Loading a Profile](#) to set the Test Case configuration with a **Profile** prior to execution, instead of performing the configuration in section 4.1 that follows. However, note that you will need to have the **PTM** open to the **Configure Method** tab to load an existing **Profile**.

4.1 Configure the Test Environment with the PTM Wizard

The **Protocol Test Manager (PTM)** is a user interface (UI)-based tool that helps you configure and run **Test Cases**.

► To access the **PTM** and begin configuration tasks, perform that steps that follow:

1. Using the **Remote Desktop Connection** application, connect with the Client01 computer by entering the client computer name in the text portion of the **Computer** drop-down and the Domain\Username in the **User name** text box.

The computer and user names for these inputs should be provided to you by the instructor of this Tutorial. You may need to click the **Show Options** drop-down arrow to access the data entry points.

2. When you are prompted for additional credentials, such as a password, enter the password that you received by the instructor of this Tutorial.
3. After you connect with the Driver VM computer, launch the **PTM** application from the blue icon desktop shortcut that should exist on this computer.

Important

At the discretion of the Lab instructor, optionally start the **PTM** while running as Administrator to enable the use of **Microsoft Message Analyzer** later on during the analysis phase. To do this, right-click the **PTM** shortcut on your desktop and select **Run as administrator** from the context menu that appears. This will allow Message Analyzer to open and analyze event trace log (ETL) data that is saved to file by executing Test Cases, that is, if tracing is enabled in **PTM** (see step 11 of this procedure). Message Analyzer provides an alternate method of analyzing test result details.

4. On the **Select Test Suite** tab of the **PTM**, click **Configure Wizard** in the upper-right sector of the UI to start the wizard, as shown in the figure that follows.

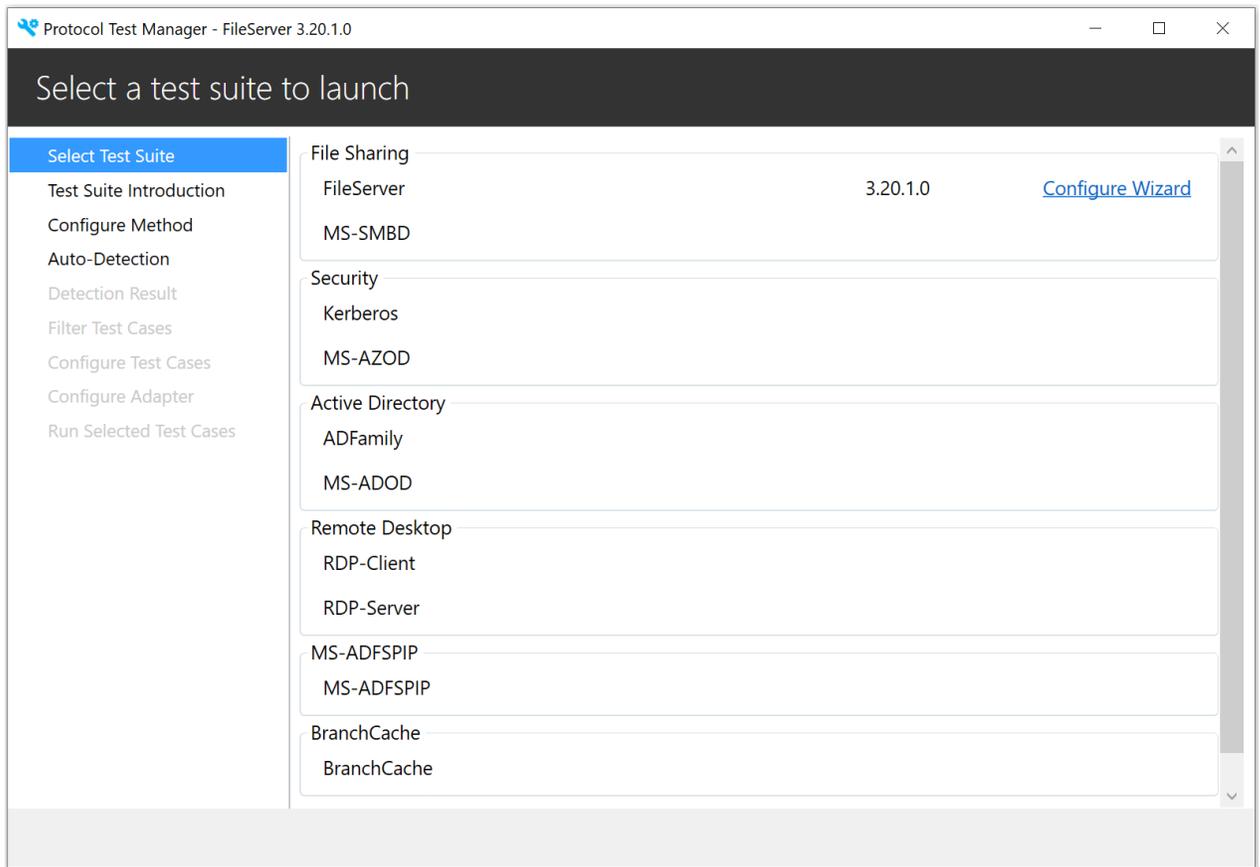


Figure 4. Protocol Test Manager : Select Test Suite tab | Launching the Configure Wizard

5. On the **Test Suite Introduction** tab of **PTM** that displays, review the general information about the File Server Protocol Family Server Test Suite and then click **Next** to display the **Configure Method** tab of the **PTM**.

Optionally, click **Deployment Guide** to quickly review a facsimile of the **Domain Environment** deployment in which you will be working. Click **Back** when complete; then click **Next**.



Note

The **Domain** environment shown ahead in Figure 6 of this Tutorial, does not require any configuration. You will simply be making use of an existing environment that is compatible with the [FileServer-SMB2 Test Suite](#) and **PTM**. However, outside the Lab session, you are free to use a **Workgroup** environment if you want, which provides a simpler network configuration for running the Test Suite. Also, given that the Lab session environment is primarily focused on users that are new to the Test Suites, it is limited for the sake of simplicity.

For this reason, several of the protocols that are included in the File Server family of protocols shown in the figure that follows, are not utilized in this Lab session, such as MS-FSRVP, MS-SWN, MS-RSVD, and MS-SQOS.

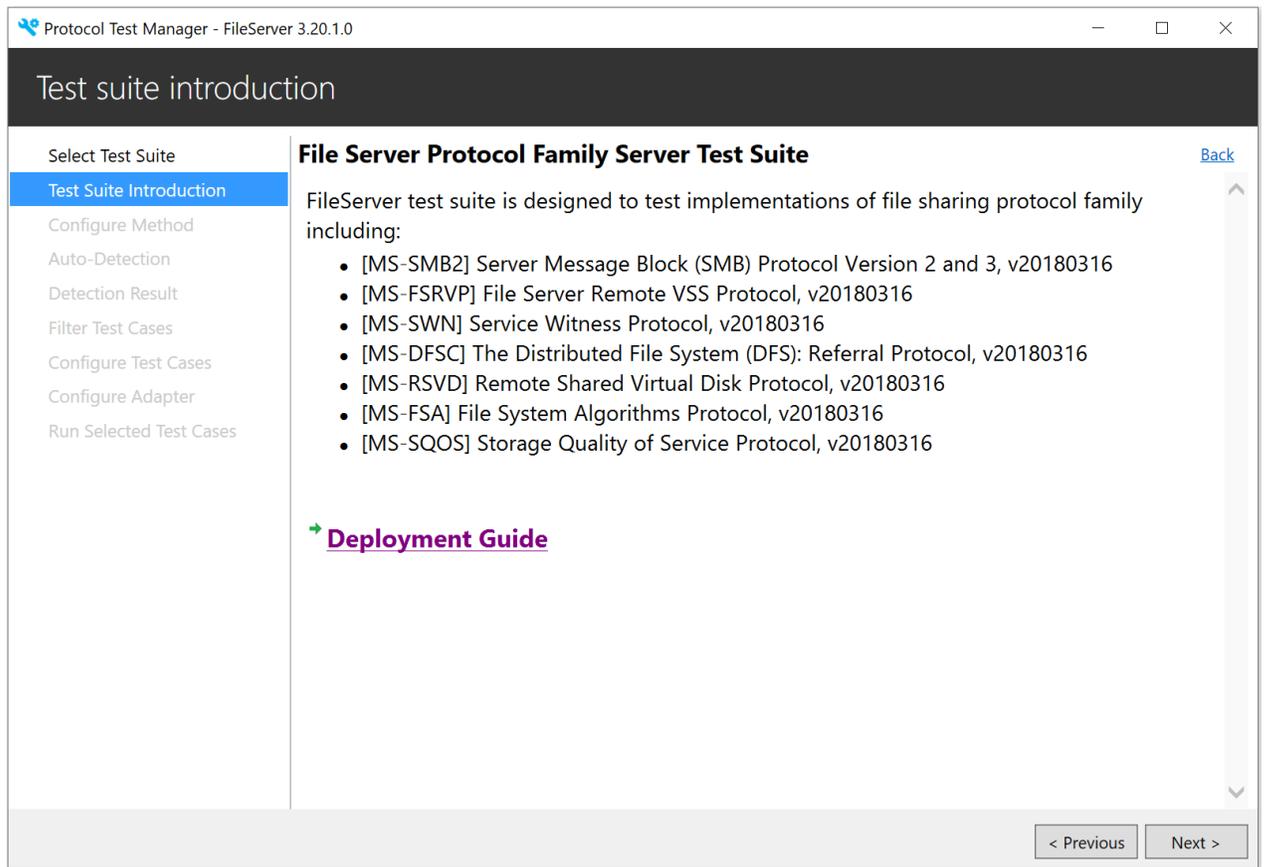


Figure 5. Protocol Test Manager : Reviewing the Test Suite protocol family

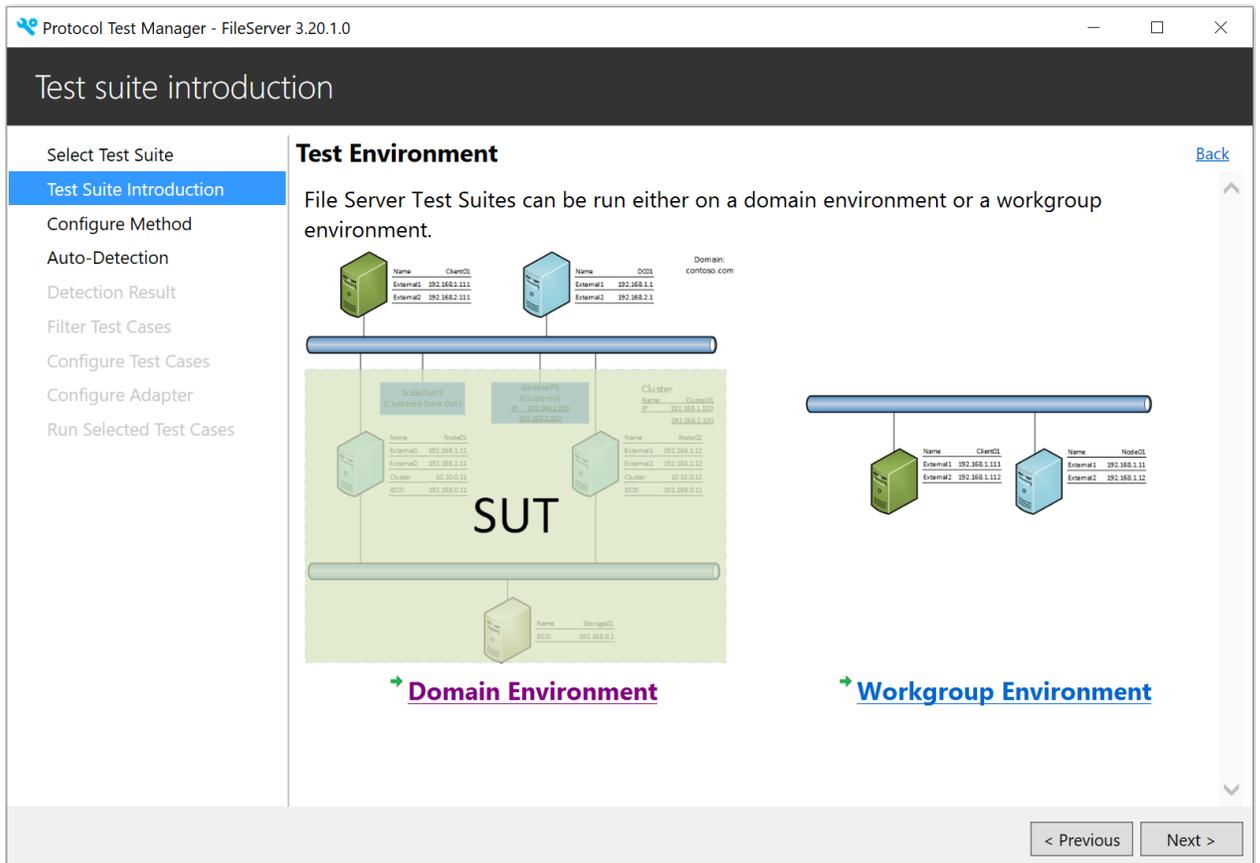


Figure 6. Protocol Test Manager : Reviewing the Domain Test Environment

 **Note**

For more information about the Domain environment setup, click the **Domain Environment** link on the **Test Environment** page of the **PTM**.

6. On the **Configure Method** tab of **Protocol Test Manager**, click the **Run Auto-Detection** option that is shown in the figure that follows. Click **Yes** when the **Configure test suite using auto-detection** dialog displays.

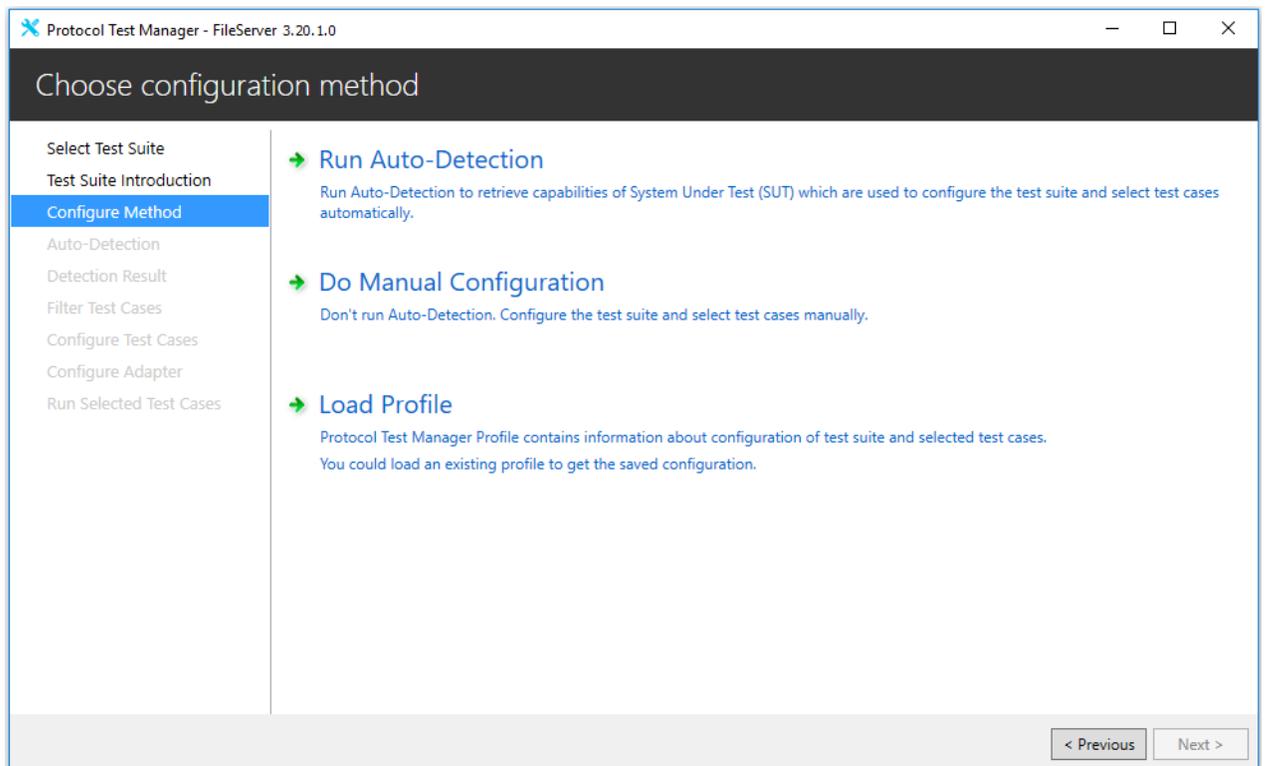


Figure 7. Protocol Test Manager : Selecting a configuration method

7. Follow the bullet points below to configure the **Test Suite** in accordance with **Auto-Detection** requirements:
 - On the **Auto-Detection** tab of **Protocol Test Manager**, ensure that the prerequisite information from auto-detection is correct based on the default values, which are loaded from .ptfconfig files, and make any necessary corrections. If the information to verify is not specified in the **Target Share**, **Domain Name**, **User Name**, **Password**, and **Authenticate** text boxes immediately below, your instructor should have the data. Compare these values to those on your **Auto-Detection** tab, as similar to the values shown in the Table that follows.

Table 2. Auto-Detection prerequisite info

Target Share:	\\<SUTName>\\$mbbasic
Domain Name:	<DomainName>.org
User Name:	lolab
Password:	Password01!
Authentication:	Negotiate

- If the specified information looks correct, click the **Detect** button on the **Auto-Detection** tab.
- Click the **Yes** button on the Warning dialog when it appears.

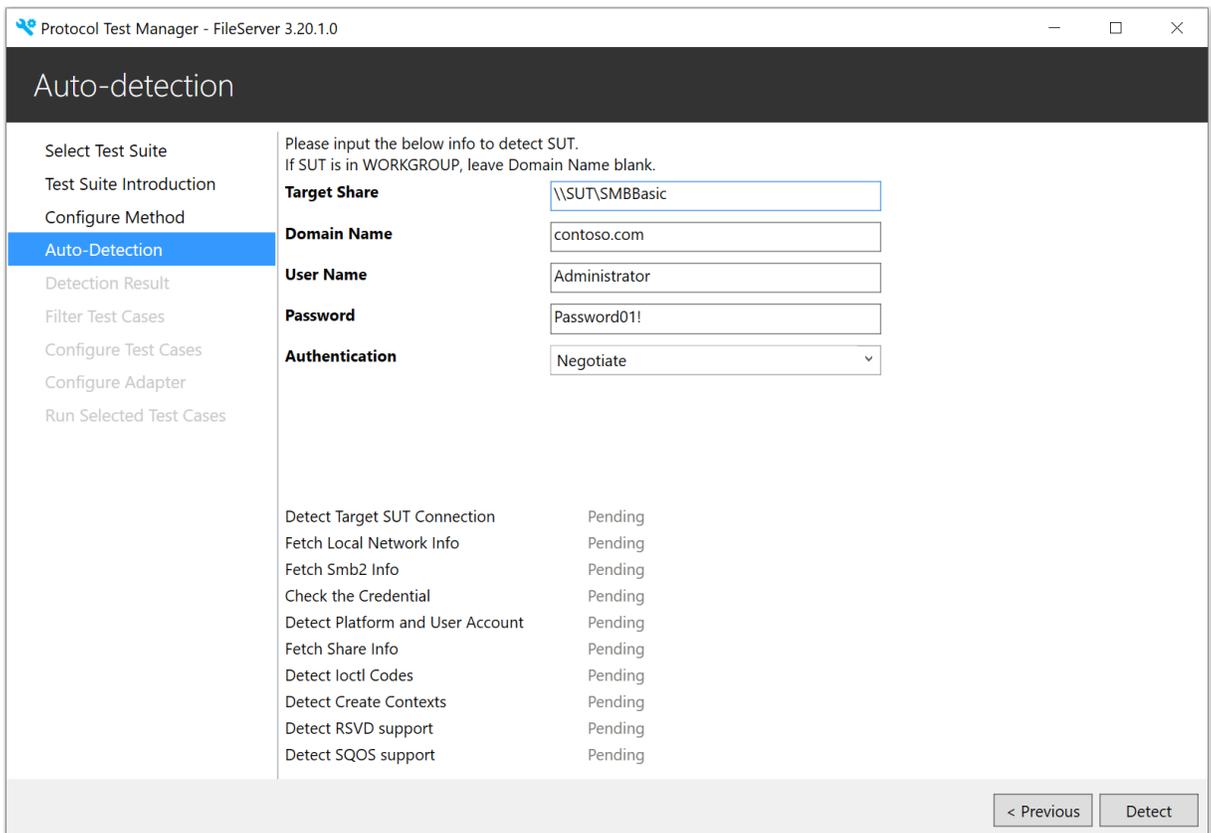


Figure 8. Protocol Test Manager : Verifying Auto-detect input information

 **Important**

Based on the Auto Detect process, the PTM will initially display the default Test Cases that support the current configuration of the SUT test environment on the **Filter Test Cases** tab, including all supported SMB2 dialects. However, note that you will be modifying the initial Test Case configuration for this Lab session environment.

In addition, features that are not supported by the provided SUT test environment, such as Remote Shared Virtual Disk (RSVD), will display the **Failure** indication  whenever an associated information node is expanded on the **Detection Result** tab of the PTM.

Features that are supported display the **Passed** indication .

- After detection has successfully completed, as indicated by the **Finished** flag next to each item in the lower Auto Detection list, click **Next** to check the **Detection Result**.
- On the **Detection Result** tab of **Protocol Test Manager**, as shown in the figure that follows, a summary is provided within information nodes that includes **Capabilities, IoCtl Codes, Create Contexts**, and so on, to indicate what is supported by the SUT configuration. Review this information to ensure that you have a similar result.
- Perform clicks on any node to toggle the results and expose or hide result details, respectively.

As you select different feature nodes, note that a detection method description appears in the lower sector of the PTM UI.

- When your review is complete, click **Next**.

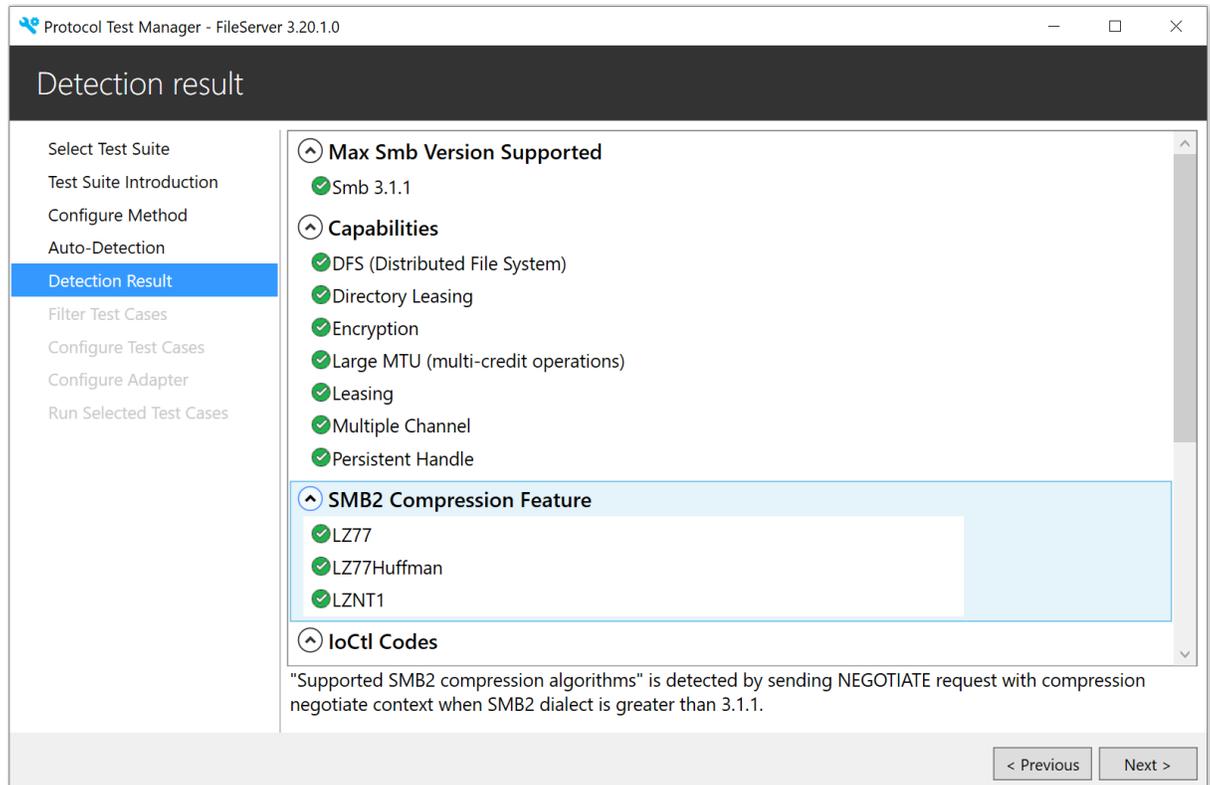


Figure 9. Protocol Test Manager : Reviewing detection results

8. In the **Selected test cases** pane on the **Filter Test Cases** tab of **Protocol Test Manager** shown in the figure that follows, observe the display of Test Cases. If the detection results show that your SUT is not supporting Test Cases for a particular feature, that feature name and/or the Test Cases will be rendered in italics.

More Information

If you want to know more about the functions of specific Test Cases, you can read descriptions in the [File Server Protocol Family Test Design Specification](https://github.com/Microsoft/WindowsProtocolTestSuites/blob/staging/TestSuites/FileServer/docs/FileServerTestDesignSpecification.md), <https://github.com/Microsoft/WindowsProtocolTestSuites/blob/staging/TestSuites/FileServer/docs/FileServerTestDesignSpecification.md>.

9. In the **Filter** and **Feature** panes on the **Filter Test Cases** tab, you will manually select or unselect Test Cases as guided by the Lab session instructor. You will be configuring the PTM to run the Test Cases associated with all SMB2 dialects, those in the BVT category, and a few others. You should end up with 91 tests when complete. This assures that you will be running the recommended Test Cases for this Tutorial.

Note

The 'BVT' test category name is used to infer the most important tests run by PTM.

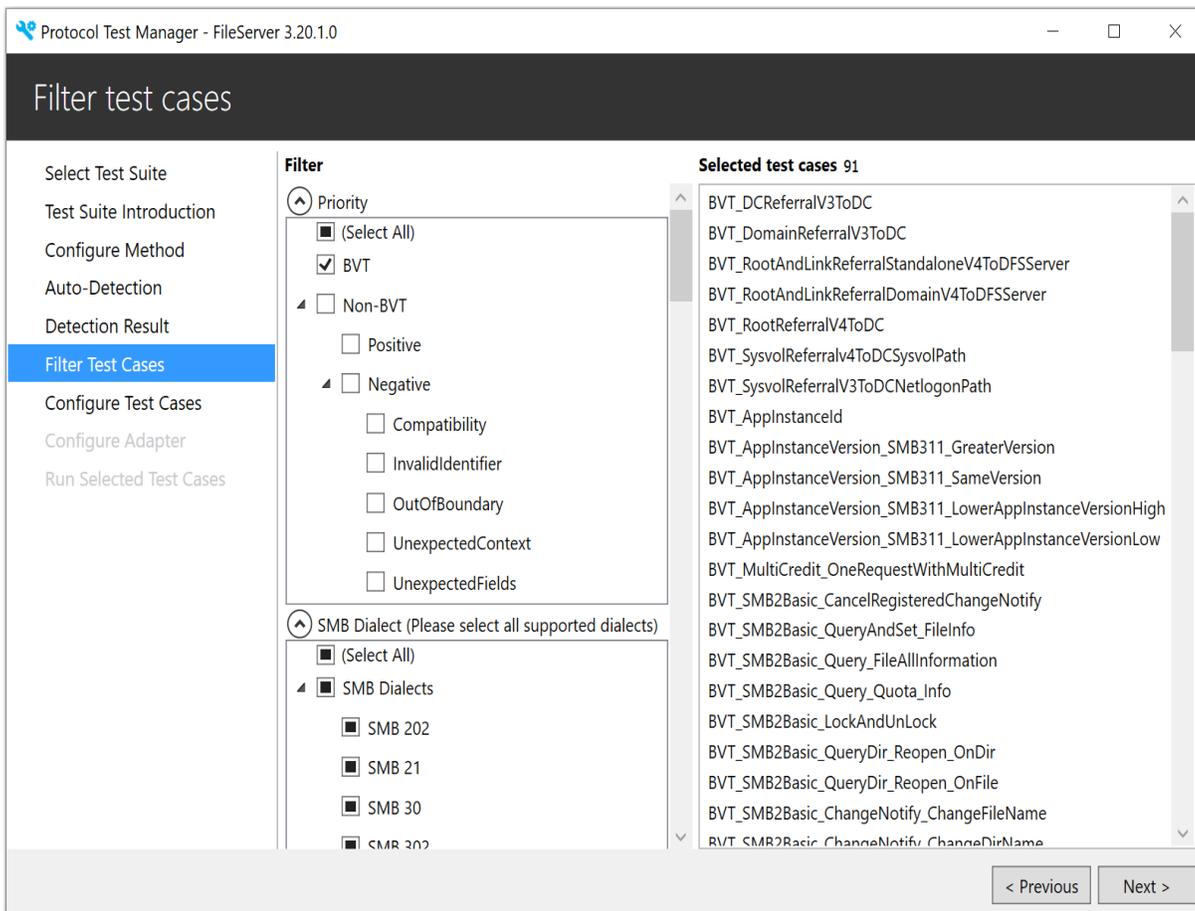


Figure 10. Protocol Test Manager : Filtering-Selecting test cases

10. On the **Configure Test Cases** tab of **Protocol Test Manager** shown in the figure that follows, verify the correctness of the default property values that were set for the specified **Groups**, with respect to your detection results. Note that you can edit the values if necessary.

 **Tip**

If you edit the configuration values for the **Common** group in the PTM and you need to recover them later, you can view the default detected values for your environment in the following configuration file on the Driver computer:

```
%systemdrive%\MicrosoftProtocolTests\FileServer\Server-Endpoint\3.20.1.0\bin\CommonTestSuite.Deployment.ptfconfig
```

Other configuration files for the **Test Suite** are located in the same directory:

```
%systemdrive%\MicrosoftProtocolTests\FileServer\Server-Endpoint\3.20.1.0\bin\
```

 **Tip**

To view the meaning of the items in the **Properties** column on the **Configure Test Cases** tab of **Protocol Test Manager**, mouse-hover over any property to display a pop-up containing the information, as shown in the figure that follows.

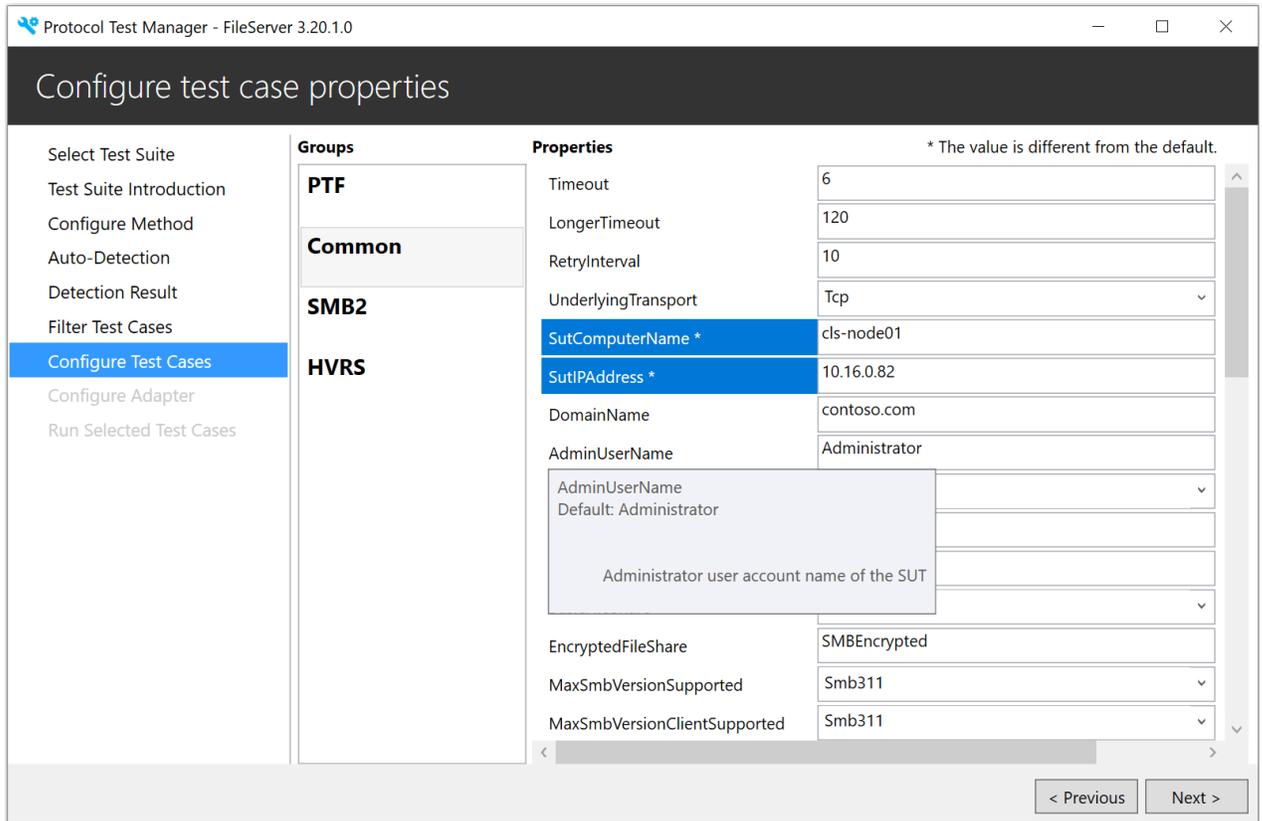


Figure 11. Protocol Test Manager : Verifying the Common property values

When complete, perform the next step *only* at the discretion of the Lab instructor. Otherwise, click **Next**.

11. Optionally, perform the tasks in the bullet points that follow to set up for assessing event trace log (ETL) data generated by the Test Cases, with the use of **Microsoft Message Analyzer** as an additional test results analysis tool:
 - Under the **Groups** pane on the **Configure Test Cases** tab of **Protocol Test Manager**, click **PTF** to open the **Properties** display for **NetworkCapture**, as shown in the figure that follows.
 - Click the **NetworkCapture** node to display the data entry fields.
 - Set the **StopRunningOnError** field to **false**, if it is not already set that way.
 - Ensure that the **CaptureFileFolder** path is set to “C:\FileServerCaptureFileDirectory” and also verify that this folder actually exists on the Driver computer.
If it does not, then please create it in the specified location.
 - Set the **Enabled** field to **true**.
 - When complete, click **Next**.

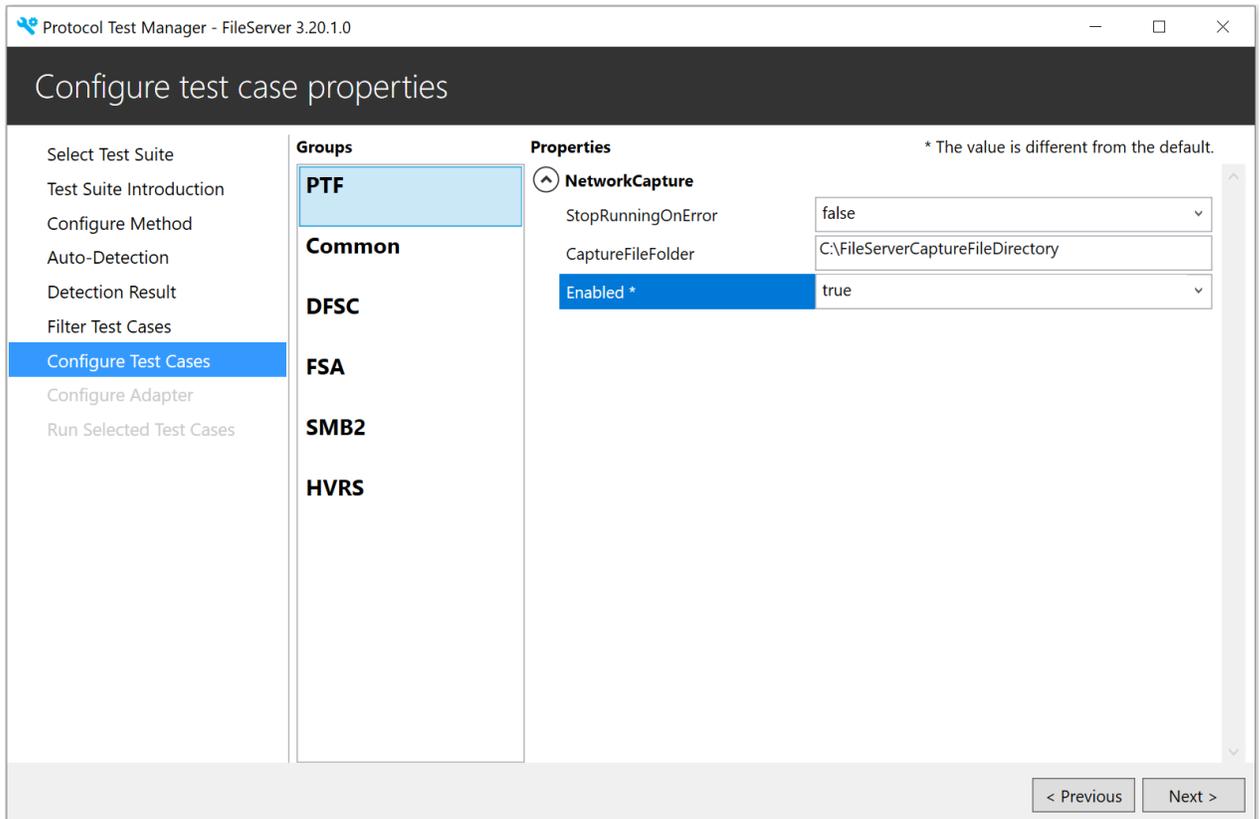


Figure 12. Protocol Test Manager : Setting up for network capture of ETL data

12. On the **Configure Adapter** tab of **Protocol Test Manager** and from the **Type** drop-down of the **Common SUT control adapter**, use the default setting of **Power Shell**, as shown in the figure that follows, then click **Next**.

Note that you can ignore the **Failover SUT control adapter** setting, given that failover tests are not run in this Lab session.

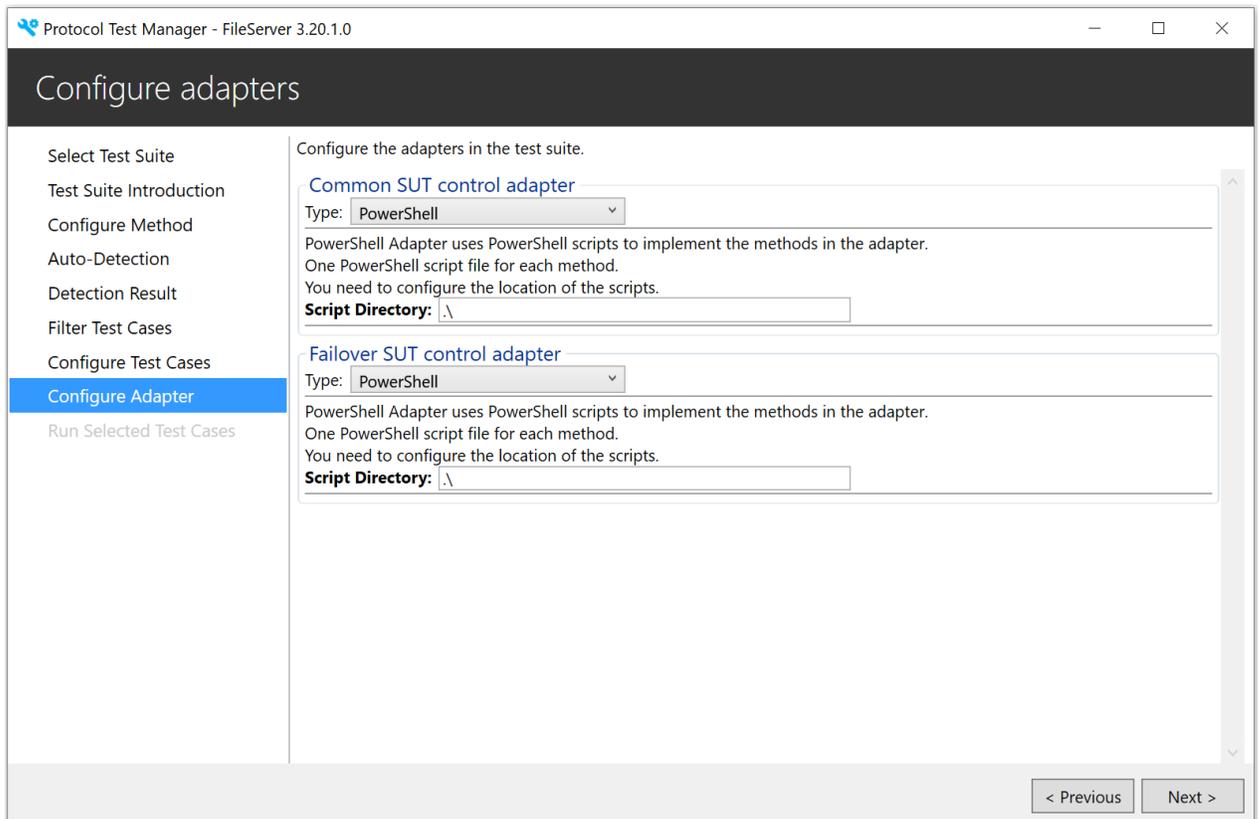


Figure 13. Protocol Test Manager : Configuring the SUT control adapter

13. Leave the **Protocol Test Manager** open and proceed to section [5.0 Running the Test Suite Test Cases](#) to run your tests.

4.2 Configure the Test Environment by Loading a Profile

If you have created one or more **Profiles**, follow the procedure in this section to load a **Profile** with the Protocol Test Manager **Load Profile** option.

► To load a PTM Profile

1. On the **Configuration Method** tab of the **PTM**, click **Load Profile**, select an existing profile in the **Open** dialog that displays, and then click the **Open** button, as shown in the figure below.

If the **Open** dialog does not open to the user **Documents** folder by default, navigate to it manually. If you stored one or more **Profiles** in a different directory location when performing the procedure in section [5.1 Saving a Profile](#), navigate to that location.

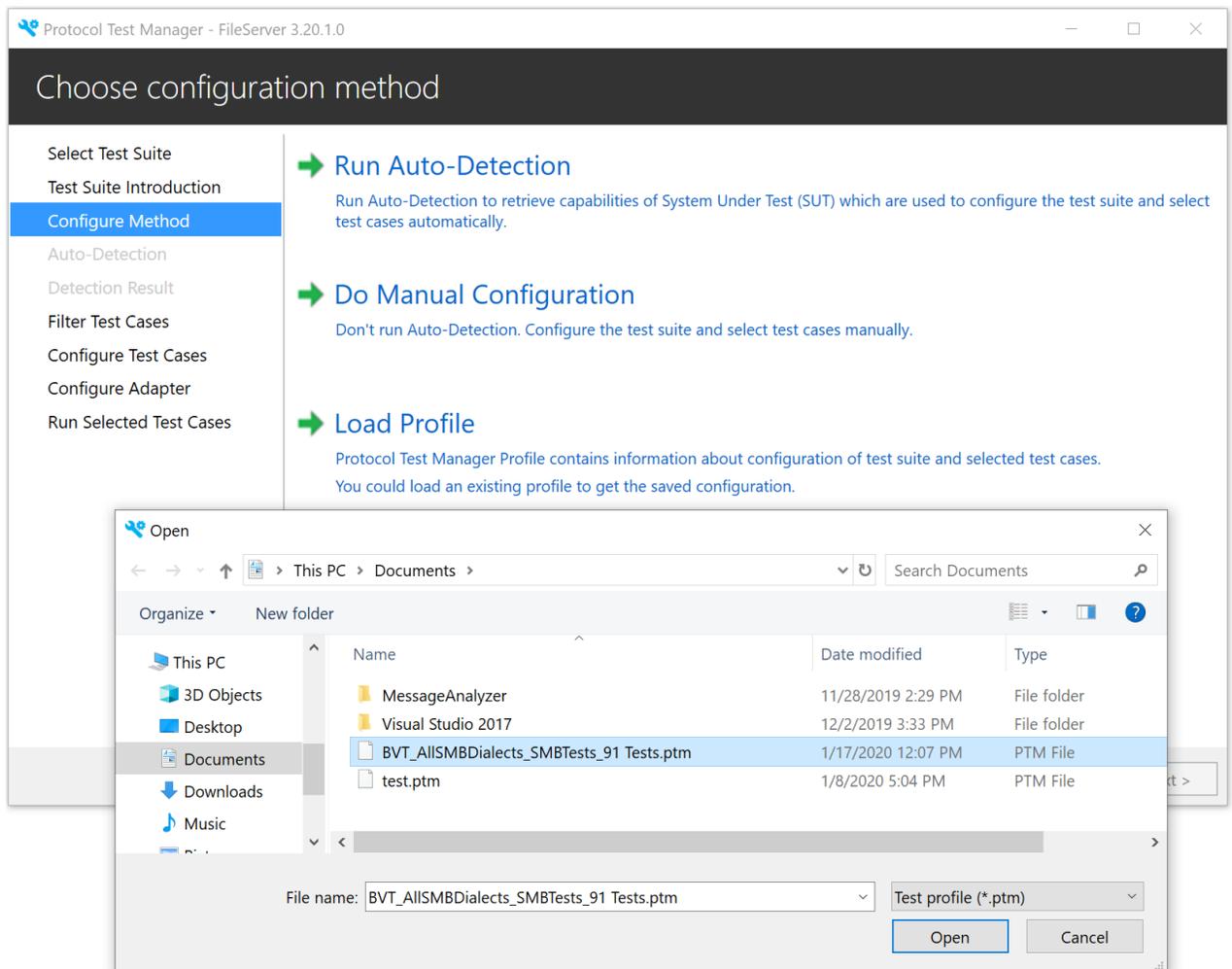


Figure 14. Protocol Test Manager : Loading an existing Profile

2. Observe that the PTM immediately opens to the **Run Selected Test Cases** tab with the **Not Run** checkbox selected along with the expected number of tests indicated.
3. Optionally navigate to the **Filter Test Cases** tab of **Protocol Test Manager**, as shown in the figure that follows, and verify that the checked Test Cases appear as expected from the **Profile** data that you imported.

Note that you can still modify your selections by selecting or unselecting Test Cases as necessary.



Important

If you want to preserve such changes, you will need to save them by overwriting the existing **Profile** or by creating a new one.

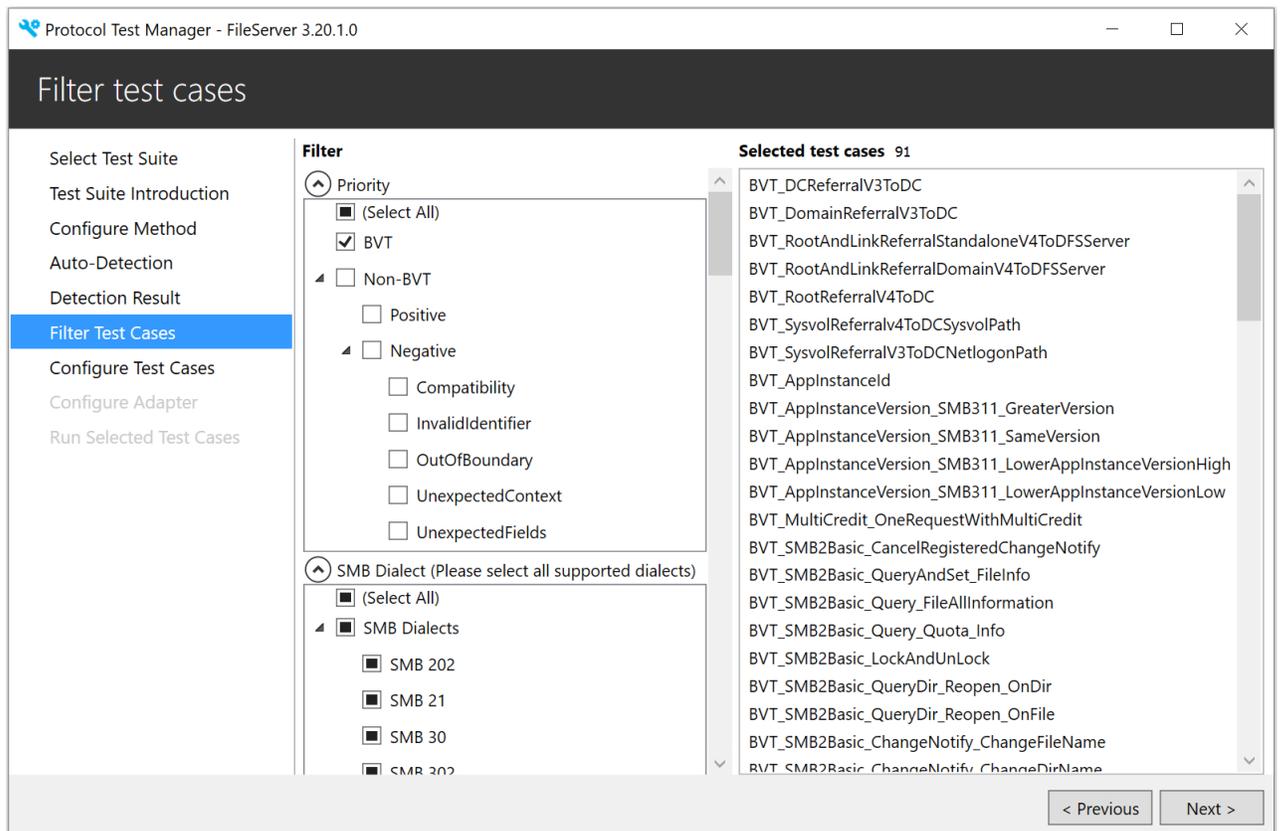


Figure 15. Protocol Test Manager: Confirming the validity of Profile test cases

5.0 Running the Test Suite Test Cases

The following options are available for running Test Cases from the **Run Selected Test Cases** tab of **Protocol Test Manager**. For this Tutorial, you will use the latter method of running selected test cases:

Important

Do not execute Test Cases now. Only execute from “To run the Test Cases” procedure ahead.

- **Run All** — click this link to run all Test Cases.

If you select this option, all Test Cases that exist under the **Not Run** checkbox on the **Run Selected Test Cases** tab of PTM will be executed, whether or not the Test Cases are actually selected. These Test Cases are the ones that you selected/filtered earlier on the **Filter Test Cases** tab of PTM and therefore appear on the **Run Selected Test Cases** tab.

This option *does not mean* you will be executing *all* the default Test Cases returned from auto-detection of the SUT configuration.

- **Run Selected Cases** — click this link to run the selected Test Cases.

If you select this option, only the Test Cases that are *selected* under the **Not Run** checkbox will be executed.

One or more command shells will display as the Test Cases execute. After execution is complete, you can view Test Case logs to the right of the Test Case list view, by selecting any Test Case in the list view.

As the tests are running you can view high level results in the following three categories. As test case execution progresses, you can observe these categories being incrementally updated:

- **Passed** — provides a dynamic indication of how many tests have passed, out of the total number selected for execution.
- **Failed** — provides a dynamic indication of how many tests have failed, out of the total number selected for execution.
- **Inconclusive** — indicates the tests that were inappropriate, unsupported, or the result of misconfiguration in the test environment.

For example, if a property set in the `CommonTestSuite.Deployment.ptfconfig` file is incorrectly configured, or a Test Case conflicts with an unexpected or invalid property value, that Test Case can finish as Inconclusive.

Note

You can drag the separator between the Test Case tree view and the log pane to adjust the width of the window for better viewing.

Important

If any of the core/preconfigured Test Cases in a test environment do not support certain features, errors will appear in the command console and PTM with respect to methods that failed when attempting to test those features. Notwithstanding errors from actual failures, you can avoid this in an outside test environment by only running Test Cases that support the features of *that* environment. However, note that the lab environment in which you are running Test Cases in this session avoids this by simply running only the recommended Test Cases for this test environment.

More Information

To learn more about Test Cases and the tests they perform, review their descriptions in the [File Server Protocol Family Test Design Specification](#).

► **To run the Test Cases:**

1. On the **Run Selected Test Cases** tab of the **PTM**, select the **Not Run** check box if it is not already selected and then click the **Run Selected Tests** link below the Test Cases list view, as shown in the figure that follows.

Given that the **Not Run** checkbox is selected, all the Test Cases under the checkbox are also selected and will run when you click the **Run Selected Cases** link, as described at the beginning of this section.

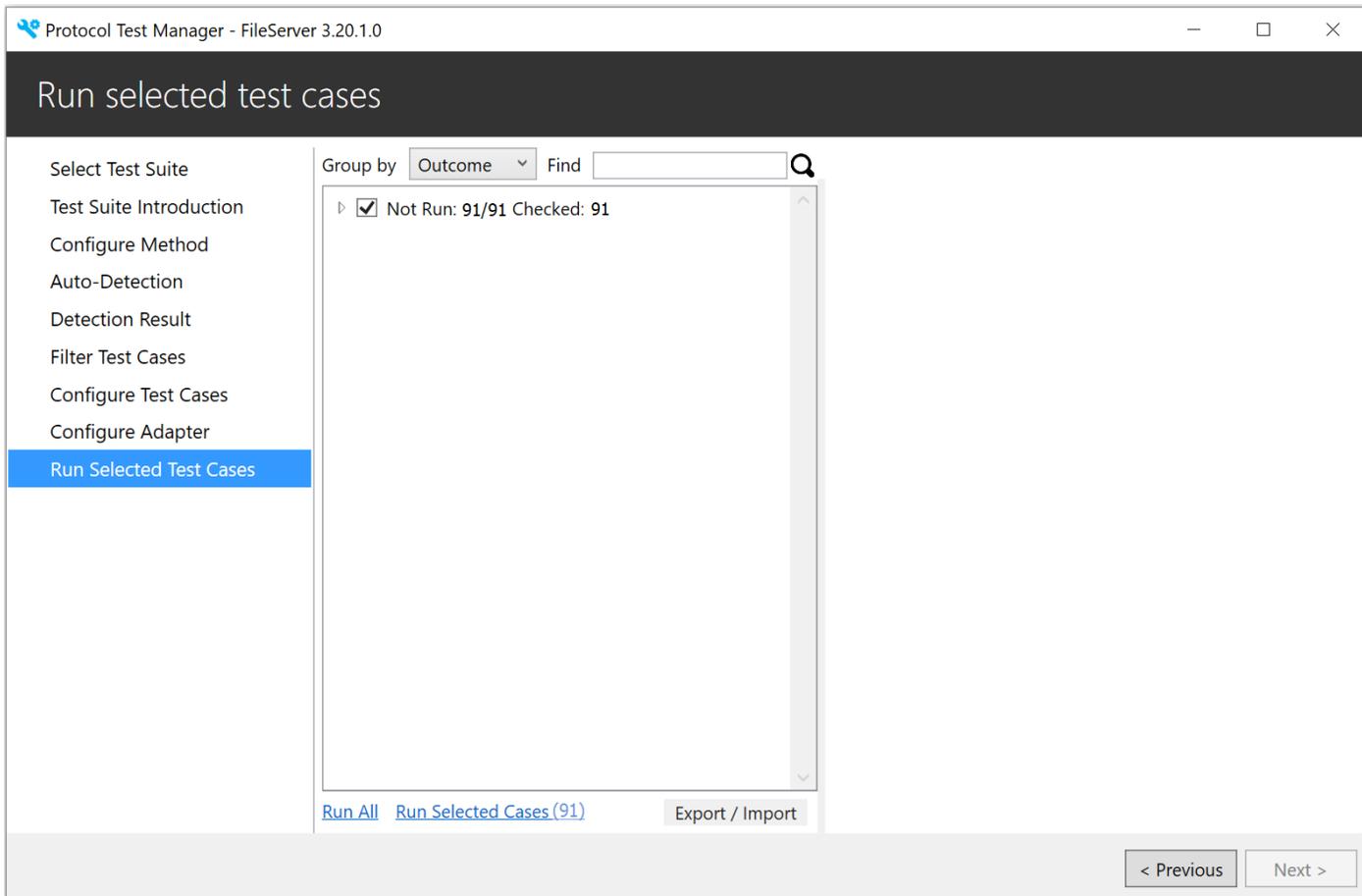


Figure 16. Protocol Test Manager : Starting Test Case execution

 **Tip**

You can also run Test Cases from the context menu that appears when you right-click in the Test Case list view itself. The context menu provides commands such as **Run Selected Cases** and **Run All Cases**, as shown in the following figure. In addition, you can select the **Uncheck All** command to uncheck all selected Test Cases.

Whether you select the **Run Selected Cases** or the **Run All Cases** context menu command, the effect will be the same as described earlier in this section. In addition, if you select the **Uncheck All** context menu command, then no Test Cases will run afterwards when you click the **Run Selected Cases** context menu command. However, if you select the **Run All** context menu command afterwards, all Test Cases listed under the **Not Run** checkbox will execute.

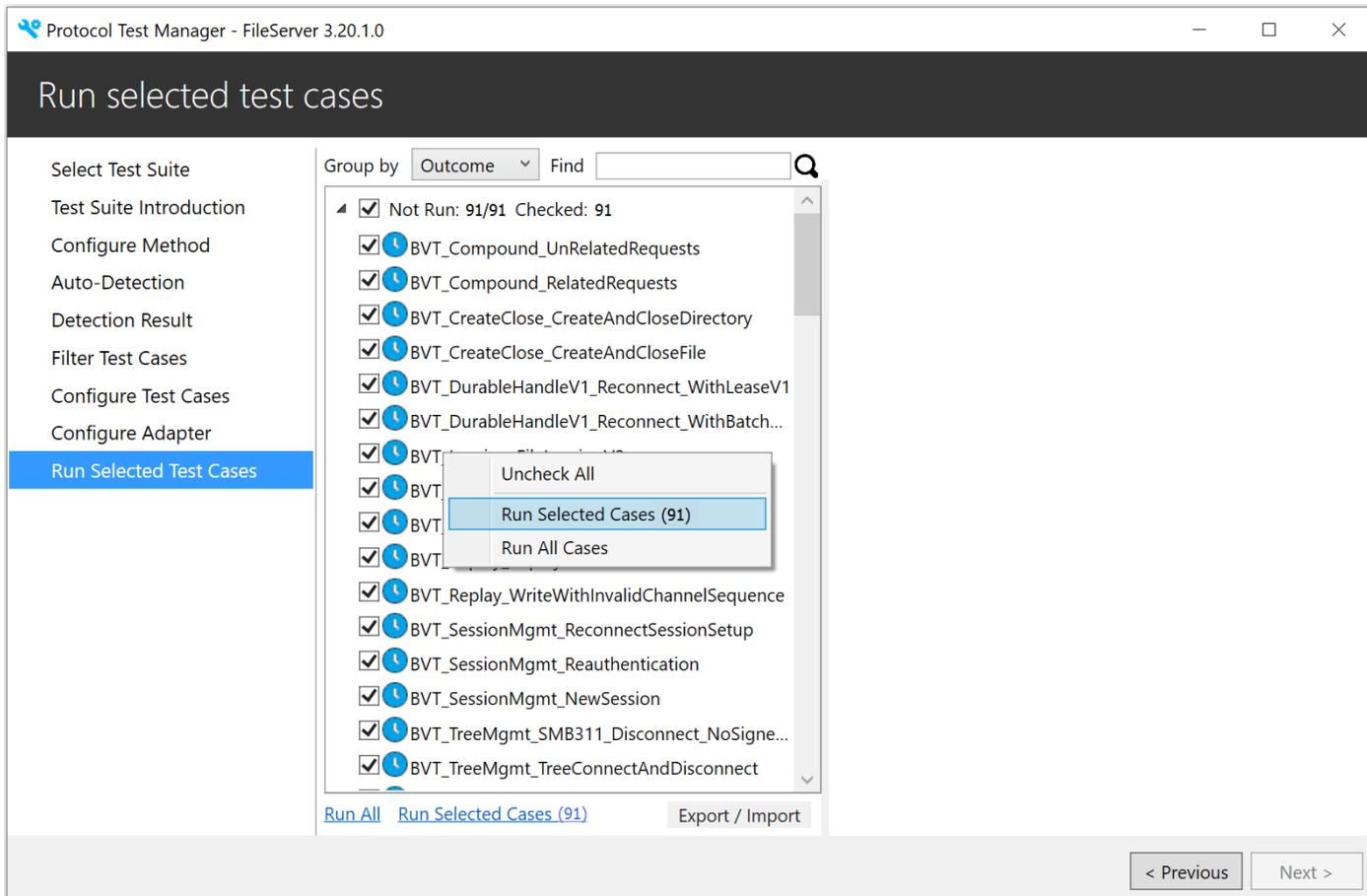


Figure 17. Protocol Test Manager : Test Case context menu commands

2. To display the context menu indicated in the figure immediately above, click the expansion node to the left of the **Not Run** check box in the Test Cases list view, then right-click the list of tests to display the menu. If Test Case execution is currently **not** in progress, click the **Run Selected Cases** command in the context menu to begin.
3. While the Test Cases are executing, observe the indications that appear in the **Passed**, **Failed**, and **Inconclusive** check box labels.



Note

You can also view the results of Test Case execution in the command consoles that host the test execution. However, note that the PTM makes the results more accessible and understandable through categorization, summaries, and the status indicators described just ahead.

Proceed to section [6.0 Analyzing the Test Results](#) to learn more about analyzing the results of Test Case execution, which includes descriptions of what was tested by any Test Case that you select.

5.1 Saving a Profile

After you complete a test run in section [5.0 Running the Test Suite Test Cases](#), based on a particular Test Case configuration, you have the option to use the PTM to save the configuration as a **Profile** that you can re-run on demand simply by locating the **Profile** in a specified directory, loading it into

the PTM, and executing it. Thereafter, you can analyze the data in the same way you would during an analysis session that you started based on the **Auto-Detect** mode described in [4.1 Configure the Test Environment Configuration with PTM Wizard](#).

In order to utilize a **Profile** in the stated manner, it is recommended that you run the Test Suite at least once and save a **Profile** that extracts the selected test cases and related configuration information, which will be the case for this Lab session.

However, in real world scenarios, note that you can optionally save a **Profile** before running your Test Case configuration, but in this case, you will not have the advantage of knowing how well the Test Case configuration performed and whether you really want to save it — for example, as a test results baseline for a certain set of features you plan to re-test for comparison.

After you save a **Profile**, you can then use it in subsequent re-runs of the profiled test environment where you specify use of the **Load Profile** option in the Protocol Test Manager. Save a **Profile** as follows.

► **To save a Profile that encapsulates the current Test Case configuration**

1. In the lower-right sector below the Test Cases list view of the **PTM** shown in the figure that follows, click the **Export/Import** link and select the **Save Profile...** drop-down menu item to save the selected Test Cases of the current test run and all related configurations.

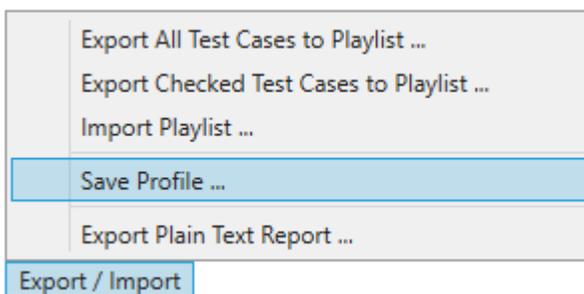


Figure 18. Protocol Test Manager : Saving a test cases Profile

This action should open a **Save** dialog. By default, the **Profile** is saved in the user **Documents** folder, but you can change it by navigating to another directory location as necessary.

2. Specify an appropriate name for the **Profile** and then close the open **Save** dialog.

5.2 Executing Test Cases From the Command Line

The Protocol Test Manager enables you execute the Test Case configuration of a previously saved **Profile** (see section [5.1](#)) with the use of a simple command string. This means you can initiate execution of a set of Test Cases from outside the PTM UI test environment and potentially from a remote location.

The application that enables you to do this is known as `PtmCli.exe`. This file resides in the following directory path on the **Driver** computer:

`C:\Program Files\Protocol Test Manager\bin\PtmCli.exe`

To execute the Test Cases of a **Profile** by using a command string, perform the procedure below:

Important

Your *should not* run the PtmCli.exe application while the PTM is running or an error will occur.

► To load a Profile from the command line

1. From the **Start** menu, type “Cmd” and then double-click the **Command Prompt** icon.
2. From the command line, navigate to the following directory location on the **Driver** computer:

```
C:\Program Files\Protocol Test Manager\bin\
```

3. At the command line, type the following command string:

```
PtmCli.exe -p <profilepath> -s
```

Note: The -p switch requires the directory path to a saved Profile and the -s switch in this command enforces execution of selected Test Cases only.

4. Confirm that the Test Case execution results begin to appear in the command console.



Note

For additional information about functionality that is available with the PtmCli tool, use the -help switch to display it:

```
PtmCli.exe -help
```

6.0 Analyzing the Test Results Data

After Test Case execution is complete, you can view the details of the execution results. The results that are likely to be of the most interest are the Test Cases that failed. The PTM provides a number of tools that can quickly point you to what the causes of a failure might be.

The PTM enables you to utilize various status indicators, stack messages, error messages, icons, and highlighted indications that can pinpoint the cause of failures very rapidly. In addition, you can launch an HTML-based display of output results that independently reproduce all the output data that displays in the PTM.



Note

Outside the space of this Tutorial and Lab environment, developers who are using a Microsoft Test Suite to test a real-world protocol implementation similarly utilize the PTM analysis features to verify the outcome of custom Test Case execution as **Passed**, **Failed**, or **Inconclusive**. Each of these indications can be of equal importance when it comes to resolving issues that are critical to a successfully functioning protocol.

6.1 Test Results Output Status Indicators

Some of the status indicators that you will encounter as you review your test results are described in the table that follows:

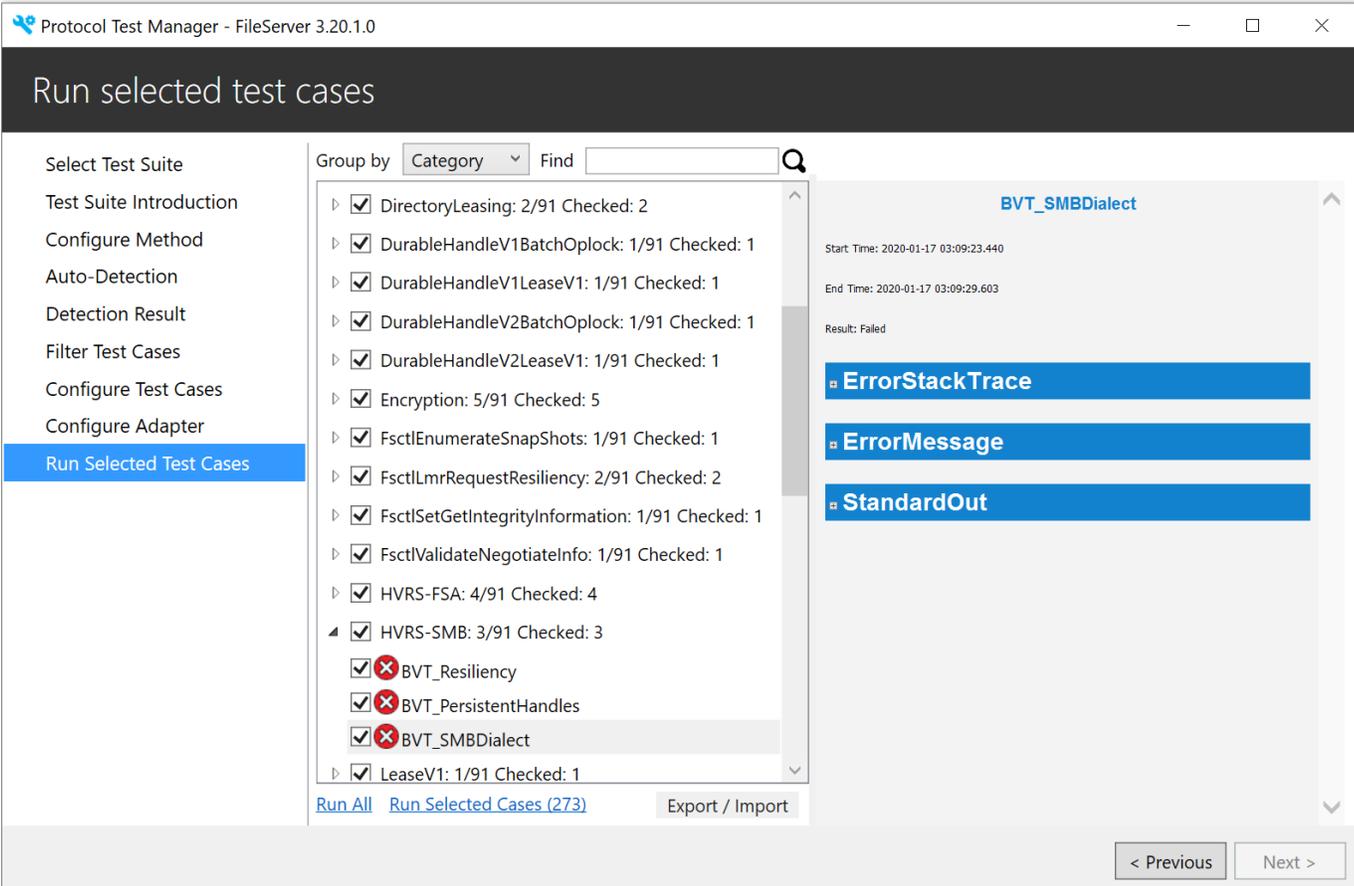
Table 3. Test results status indicators

Status Indicator	Description	UI Location
Initial test results status summary	<p>Results display in either of these configurations:</p> <ol style="list-style-type: none"> 1. The number of Test Cases that passed, failed, or were inconclusive. 2. The number of Test Cases that executed for specific SMB2 operations, where the number of tests that were run are identified. 	<ol style="list-style-type: none"> 1. The first results display configuration is shown as expandable Passed, Failed, and Inconclusive results category* check boxes in the left-hand test cases pane of the PTM. Appears when selecting the Outcome item in the Group by drop-down list. 2. The second results display configuration is broken down into the various categories of SMB2 operations being tested, for example Credit or Negotiate, where the number of Test Cases associated with the operation is indicated, while the number of tests having issues is specified by a Checked indicator. Appears when selecting the Category item in the Group by drop-down list.
Start Time and End Time	Exposes the overall duration of Test Case execution.	Appears in the upper left sector of the right-hand test results pane for any selected Test Case in any results category.
Result	The results category for a particular Test Case, for example; Passed , Failed , or Inconclusive .	Appears in the upper left sector of the right-hand test results pane for any selected Test Case in any results category.
Test Case functionality	The general purpose of a Test Case.	Typically described in a [Comment] tag that appears in the StandardOut category of results.
Debug output data	Informative data that is displayed in tags in the StandardOut category of results.	<p>Includes data that displays in information tags such as the following:</p> <ul style="list-style-type: none"> ▪ [TestInProgress] ▪ [Comment] ▪ [Debug] ▪ [CheckPoint] ▪ [CheckSucceeded] ▪ [CheckFailed] ▪ [TestStep] ▪ [TestPassed] ▪ [TestFailed]
Error output data	Holds data that you can analyze to identify the	Appears in the ErrorStackTrace and ErrorMessage category of results.

	source of Test Case failures.	
Test case status	<ul style="list-style-type: none"> ▪  — indicates the executed Test Case status as Passed. ▪  — indicates the executed Test Case status as Inconclusive. ▪  — indicates the Test Case status is Not Run. ▪  — indicates the executed Test Case status as Failed. 	These indicators appear immediately to the right of the Test Case check box after test execution, to indicate the test result status, with exception of the blue Not Run status indicator which appears prior to test execution.

* This same information is repeated as the text of a hyperlink that appears in the upper-right-hand sector of the PTM. This hyperlink opens an HTML-based display of test results.

The figure that follows is an example of the **Category** grouping for Test Cases (grouped by SMB2 operations that contain the Test Cases), and also shows the three results output categories for test analysis in the right-hand pane of the PTM.



The screenshot shows the Protocol Test Manager (PTM) interface. The title bar reads "Protocol Test Manager - FileServer 3.20.1.0". The main window title is "Run selected test cases". On the left, a navigation pane lists various options, with "Run Selected Test Cases" highlighted. The main area is divided into two panes. The left pane shows a list of test cases grouped by "Category". The list includes items like "DirectoryLeasing: 2/91 Checked: 2", "DurableHandleV1BatchOplock: 1/91 Checked: 1", and "BVT_SMBDialect" which is marked with a red X icon. The right pane displays the results for the selected test case, "BVT_SMBDialect". It shows the start and end times, and the result "Failed". Below this, three expandable sections are visible: "ErrorStackTrace", "ErrorMessage", and "StandardOut". At the bottom of the interface, there are buttons for "Run All", "Run Selected Cases (273)", and "Export / Import".

Figure 19. Protocol Test Manager : SMB2 operation test categories | Error and StandardOut results

StandardOut Category Status Indicators

Some of the low-level indicators of the **StandardOut** category that are directly associated with the incremental step-by-step record of how the tests were conducted, along with interim results, are described as follows:

- **[TestStep]** — highlighted in **Blue**. Describes the details of a particular step in a Test Case.
- **[Debug]** — no highlighting, plain text. Describes actions that were taken during a portion of a Test Case, such as connecting to a server over TCP, as part of the **[TestStep]** in which it exists.
- **[Checkpoint]** — no highlighting, plain text. Provides values at key points during a test that can provide insights into the causes of an imminent failure. Can also include pointers to the protocol specification sections that define acceptable value types and ranges, to assist in troubleshooting.
- **[CheckSucceeded]** — highlighted in **Green**. Indicates that the actions taken at a particular check point of a **[TestStep]** were successful.
- **[CheckFailed]** — highlighted in **Red**. Indicates that the actions taken at a particular check point were unsuccessful.
- **[TestPassed]** — highlighted in **Green**. Indicates that the Test Case passed.
- **[TestFailed]** — highlighted in **Red**. Indicates that the Test Case failed.
- **[Comments]** — no highlighting, plain text. Provides other information such as brief descriptions of Test Case actions, states, values, and so on.

ErrorStackTrace Category

Provides a visual indication of the call stack where a failure occurred. The Error message itself displays at the end of the **ErrorStackTrace** information, which extends into the **ErrorMessage** category.

ErrorMessage Category

Provides details and comments that can identify the source of the error event and the conditions that existed when the error event occurred. As an aid to further inspection of the error event, the PTM can provide comments that point you to the related sections of the Microsoft SMB2 specification documentation, which may shed light on what the underlying failure may be related to, such as expected states, values, parameters, event data, and so on.

When you evaluate the PTM test results data in the following section for your selected Test Case runs, you will have a chance to review some of these features in actual Test Case execution results.

6.2 Evaluating PTM Test Result Data

To begin evaluation of the test results data presented by the PTM, perform the following procedure:

- ▶ **To analyze the results of Test Case execution, perform the steps that follow:**

1. After test execution is complete, click the expansion nodes next to the **Passed**, **Failed**, and **Inconclusive** check boxes, as shown in the figure that follows, to obtain a general sense of the test results, in terms of what passed, what failed, and tests that did not complete.

Note that by default the tests are grouped by **Outcome** in the **Group by** drop-down and that you will change this up ahead for analysis purposes.

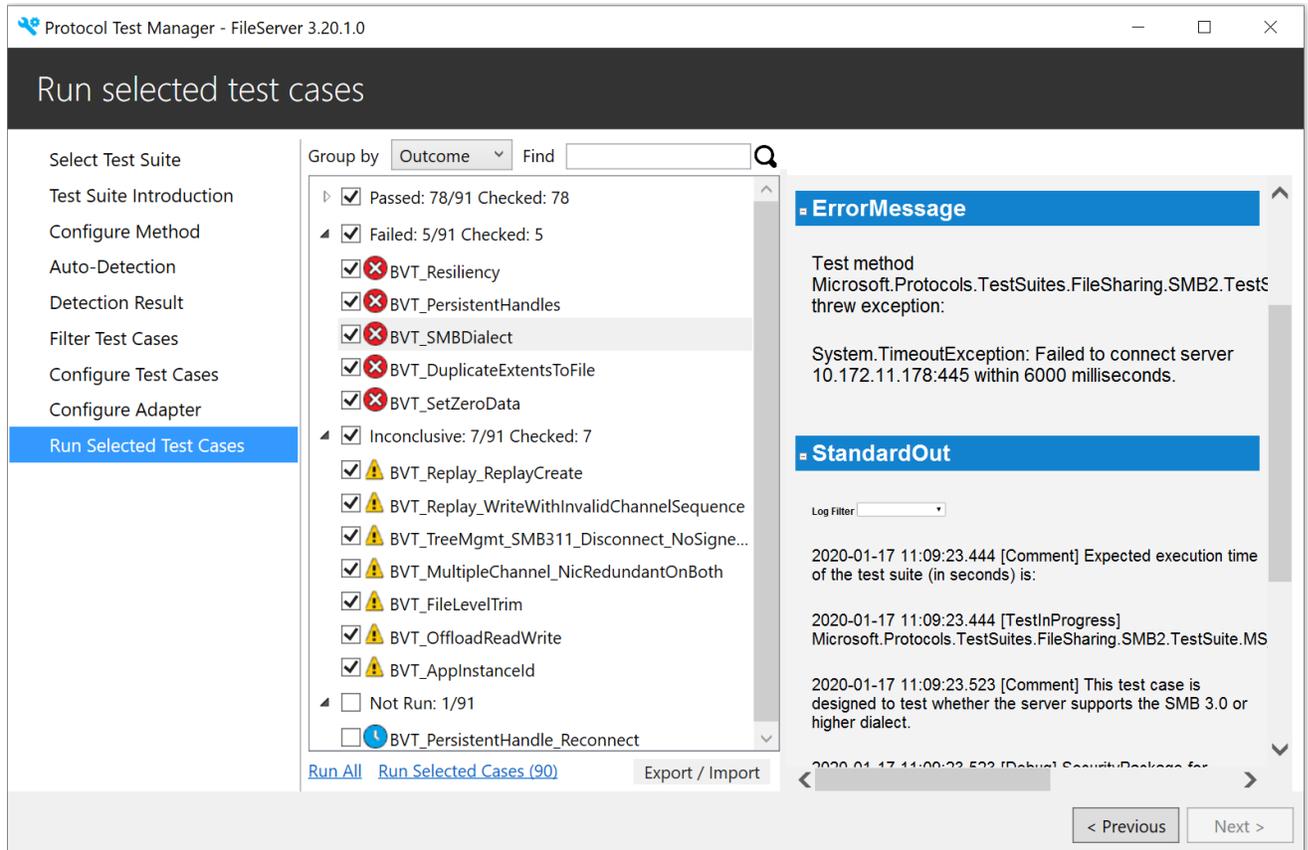


Figure 20. Protocol Test Manager : Test results grouped by Outcome

2. To drive down into the test results more definitively, select some of the test results under each of the **Passed**, **Failed**, and **Inconclusive** check boxes and observe the output indications to the right of the executed Test Cases.

But first, select the **Category** item in the **Group by** drop-down above the Test Cases list view to obtain a display such as the one shown in the figure that follows. This enables you to select Test Cases as sub-check boxes under a top-level operation, such as Smb30, that associate specific Test Case results to that particular top-level SMB30 operation.

In the figure that follows, the **SMB30** operation top-level node is expanded to show 20 tests that are related to this operation; along with a **Passed**, **Failed**, **Inconclusive**, or a **Not Run** status as appropriate to each Test Case. See Table 2 in section [6.1 Test Results Output Status Indicators](#) for relevant information on Test Case status icons.

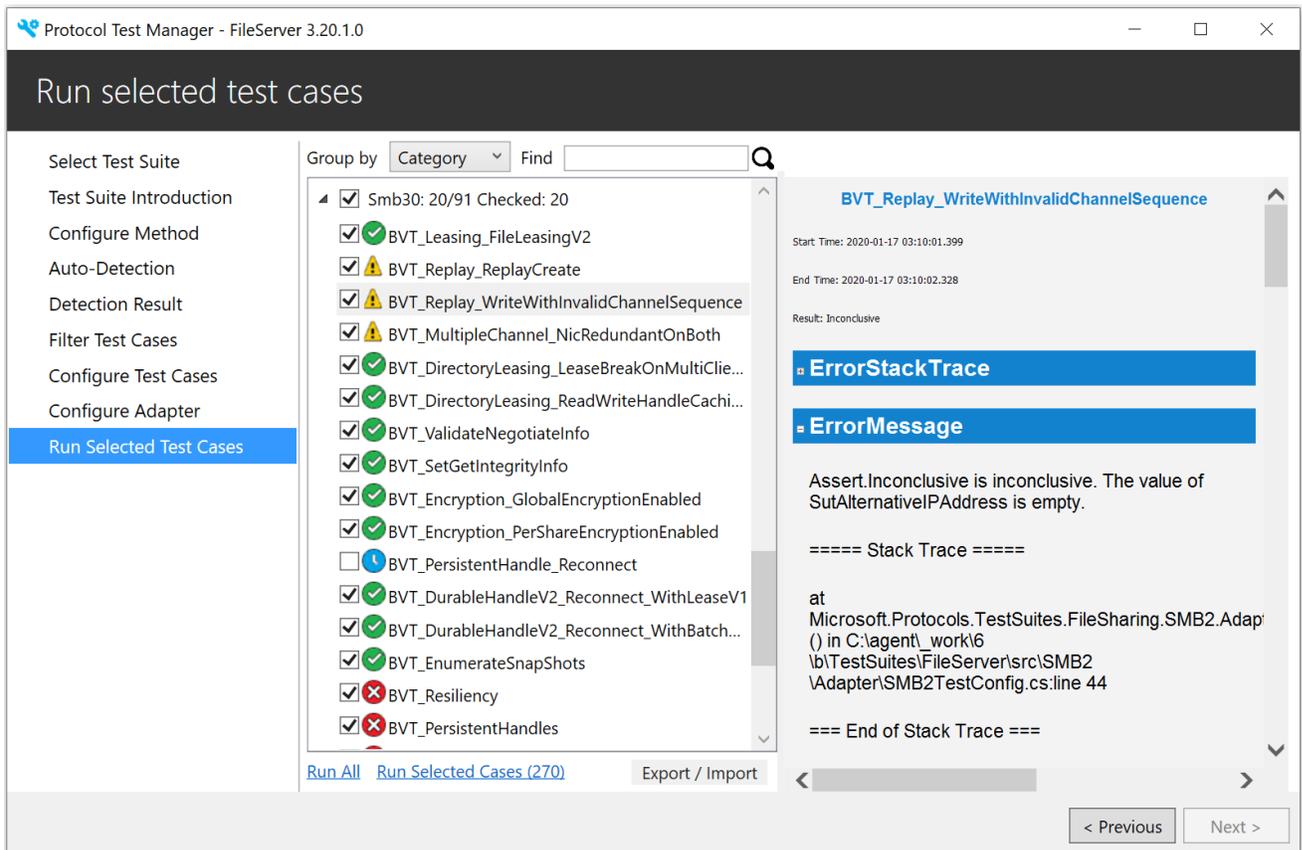


Figure 21. Protocol Test Manager : Categorized display of Test Cases for SMB30 operations

3. Select the Inconclusive Test Case **BVT_Replay_WriteWithInvalidChannelSequence** for the **Smb30** operation, as shown in the previous figure, and note the reason why the test was inconclusive in the **ErrorMessage** category of results in the right-hand pane of the PTM.

Also, if you scroll down towards the end of the **StandardOut** results (which is presently hidden in the figure), you may see that a **[TestStep]** for sending a NEGOTIATE request to start a client from an alternate channel is what resulted in the **[CheckInconclusive]** response, as below:

2020-01-17 11:10:01.862 **[TestStep]** Start a client from alternative channel by sending NEGOTIATE request.

2020-01-17 11:10:01.863 **[CheckInconclusive]** Assert.Inconclusive is inconclusive. The value of SutAlternativeIPAddress is empty.

4. Select the Passed Test Case **BVT_ValidateNegotiateInfo** for the Smb30 operation, as shown in the figure that follows, and observe that only a **StandardOut** display appears in the right-hand pane of the PTM.

The display may show various information tags such as **[TestInProgress]** **[Comments]**, **[Debug]**, and so on, in the right-hand pane of the PTM.

Note the **[CheckSucceeded]** tag highlighted in green in the figure that follows, which provides some information about IP addresses. Please go ahead and scroll through the **StandardOut** results and review the data exposed by other status indicators to familiarize yourself with the type of information they provide. You should pick a few other Test Cases at random and do the same thing.

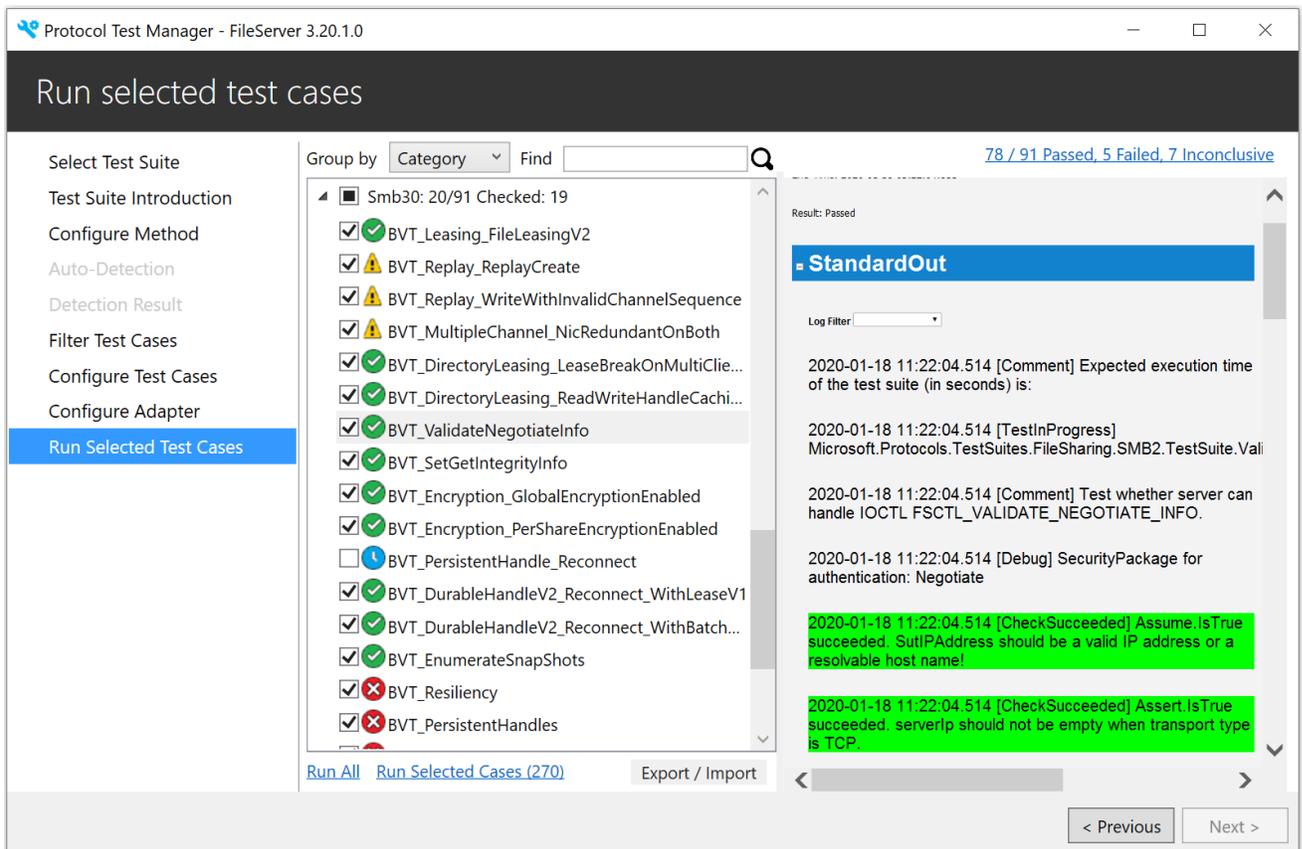


Figure 22. Protocol Test Manager : StandardOut results for a passed Test Case

- In the PTM, you can focus on the data associated with specific status indicators in the **StandardOut** display by clicking the **Log Filter** drop-down just below the **StandardOut** category title and select one or more of the filters that display, as illustrated in the figure that follows:

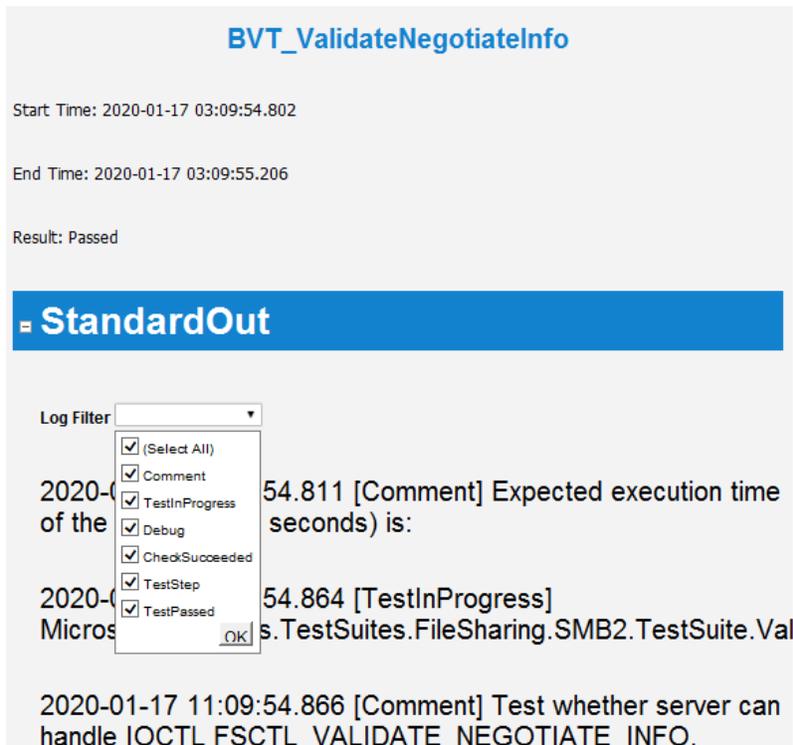


Figure 23. Protocol Test Manager : Applying Log Filtering to StandardOut test results data

- Click **OK** in the **Log Filter** drop-down and observe that the **StandardOut** results show data for only the status indicators that you selected.

This feature enables you to focus on specific types of **StandardOut** results in order to isolate the data that is important to your analysis.

Tip

You can also view the **StandardOut** results data in HTML format by clicking the test results hyperlink in the upper-right sector of the PTM user interface, as shown earlier in Figure 22.

The HTML-based test results display is shown in the figure that follows and functions similarly to the PTM, regarding the selection of Test Cases in the left-hand pane and the appearance of appropriate test results in the right-hand pane of the HTML display.

In addition, you have the option of grouping the **Case List** by **Test Result**, **Category**, or **Class**, in order to achieve varying analysis perspectives.

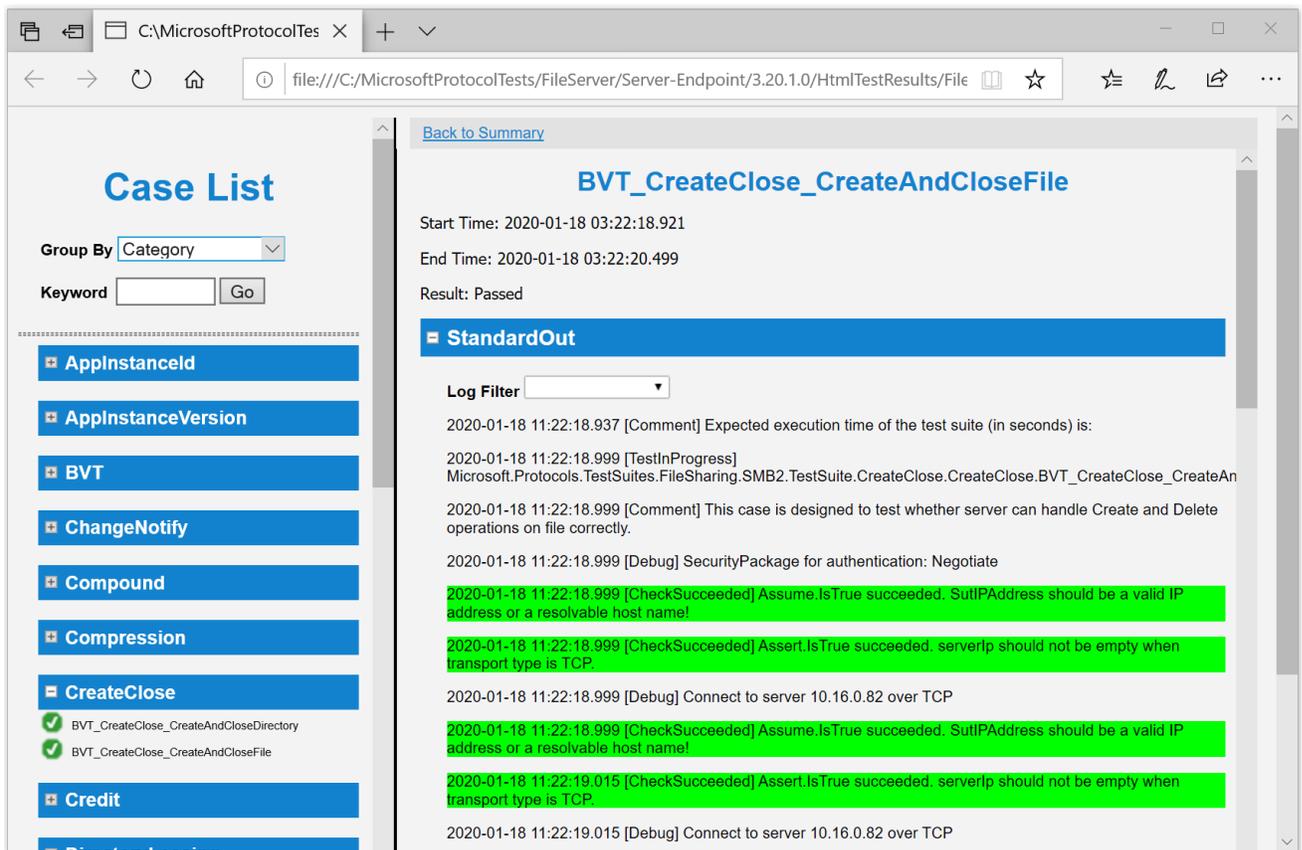


Figure 24. Protocol Test Manager : StandardOut results in separate HTML format

- In the figure that follows, the Failed Test Case **BVT_PersistentHandles** in the **Smb30** operation node is selected, as identified with a circular red icon. In the figure, note that the **ErrorMessage** and **StandardOut** results are collapsed in order to expose the **ErrorStackTrace** data more clearly.

The data contained in the **ErrorStackTrace** display is similar to what you might see in other software development troubleshooting environments, where a call was traced as it traversed through the message stack to the point where an error occurred.

In the case of the following figure, what you are seeing is the trace of a call through the FileServer-SMB2 Test Suite components to the point in code where an error occurred for an SUT component being tested. Whereas in a real world case, you might be looking at the message stack tracing of a protocol implementation where an error occurred while its functions were being exercised.

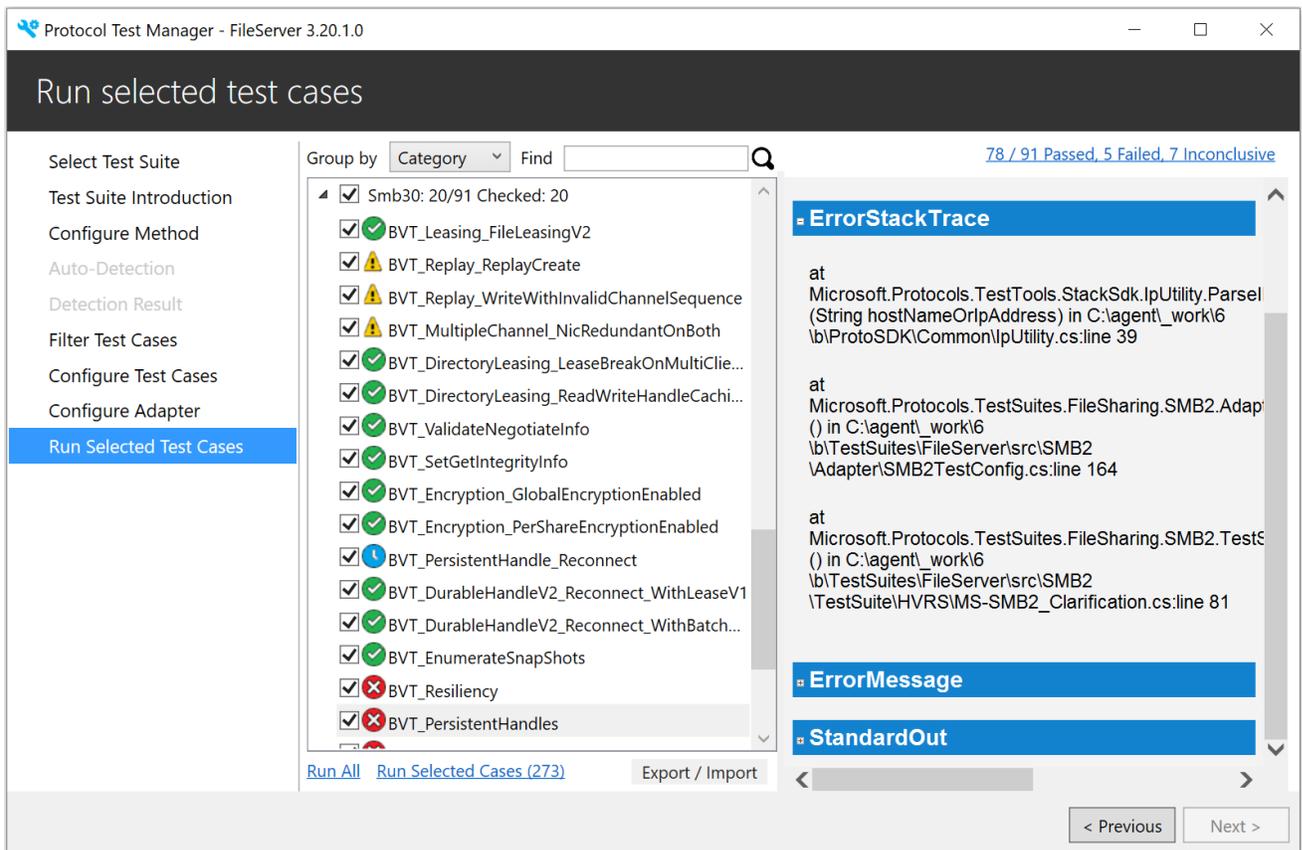


Figure 25. Protocol Test Manager : ErrorStackTrace results



Tip

Observe that as you scroll through **StandardOut** results, [Comments] information tags can provide some related background [MS-SMB2](#) documentation references that can be checked for further clarification of issues. These can appear primarily for Test Cases that have an execution status of **Inconclusive** or **Failed** and can help identify the causes of failures.



Note

To learn about analyzing Test Case results with **Message Analyzer**, see section [6.4 Analyzing Test Results with Message Analyzer](#).

6.3 Common Failures

The table in this section describes some common failures that you may encounter when running Test Cases. A section is also provided for you to enter information about any unique or unexpected issues that occurred as the result of running Test Cases.

Table 3. Common Test Case failures

Failure	Description	Potential Cause
Pervasive errors	Command line reports massive failure of tests.	Incorrect configuration involving the selection of a Windows platform while a non-

		Windows platform is actually in use.
Selective errors	Features appear on the Filter Test Cases tab in italics. Errors appear in the Selected Test Cases pane in the Group by Outcome configuration.	Features displayed in italics indicate PTM determined that the feature is not supported on the SUT. If the feature is tested anyway, failures will occur. Features were not supported by one or more test cases that ran.
Test case errors	Test case failures are reported in the right-hand sector of the Run Selected Test Cases tab of PTM.	Descriptions are provided in the ErrorMessage category of the test results.
Note: Use the sections below to note unique or unusual errors you may have detected in this Lab session		

6.4 Analyzing Test Results with Message Analyzer

If the Lab instructor elected earlier to have you enable [Microsoft Message Analyzer](#), go to the event trace log (ETL) capture file for a specific Test Case (file location is:

“C:\FileServerCaptureFileDirectory”) and perform the procedure that follows.

► **To analyze Test Case ETL data with Message Analyzer, perform the following steps:**

1. Right-click the file and then select **Open in Message Analyzer** from the context menu that appears.

The captured trace data should open in Message Analyzer.

2. Follow the directions from your instructor regarding how to discover data of interest for a selected Test Case, including test [Comments] and other status indicators described earlier.

For example, if you are looking at the results from a particular test case in PTM, you might see a display that is similar to the one that follows. In the example, the **Analysis Grid** is shown, which is the main analysis surface of Message Analyzer.

In the Analysis Grid, you will see two highlighted messages that show communication back and forth between the Driver computer (shown as IP address 191.168.1.1) launching a test case and the system under test (shown as GG74-SUT) responding. If you scroll the display downward, you will see more of this type of information under the **Source** and **Destination** columns.

However, of more interest is the data contained in the **Summary** column in the right-hand sector of the Message Analyzer display. Here you will see some of the Status indicators described in

section [6.1 Test Results Output Status Indicators](#), such as [CheckSucceeded] and [Debug]. To read more of the **Summary** data, you might have to use the horizontal scroll bar to expose it. Message Analyzer contains many other different viewers and features that could be useful in analyzing your Test Cases. If you are interested in learning more about Message Analyzer, see [Microsoft Message Analyzer Operating Guide](#) on docs.microsoft.com.

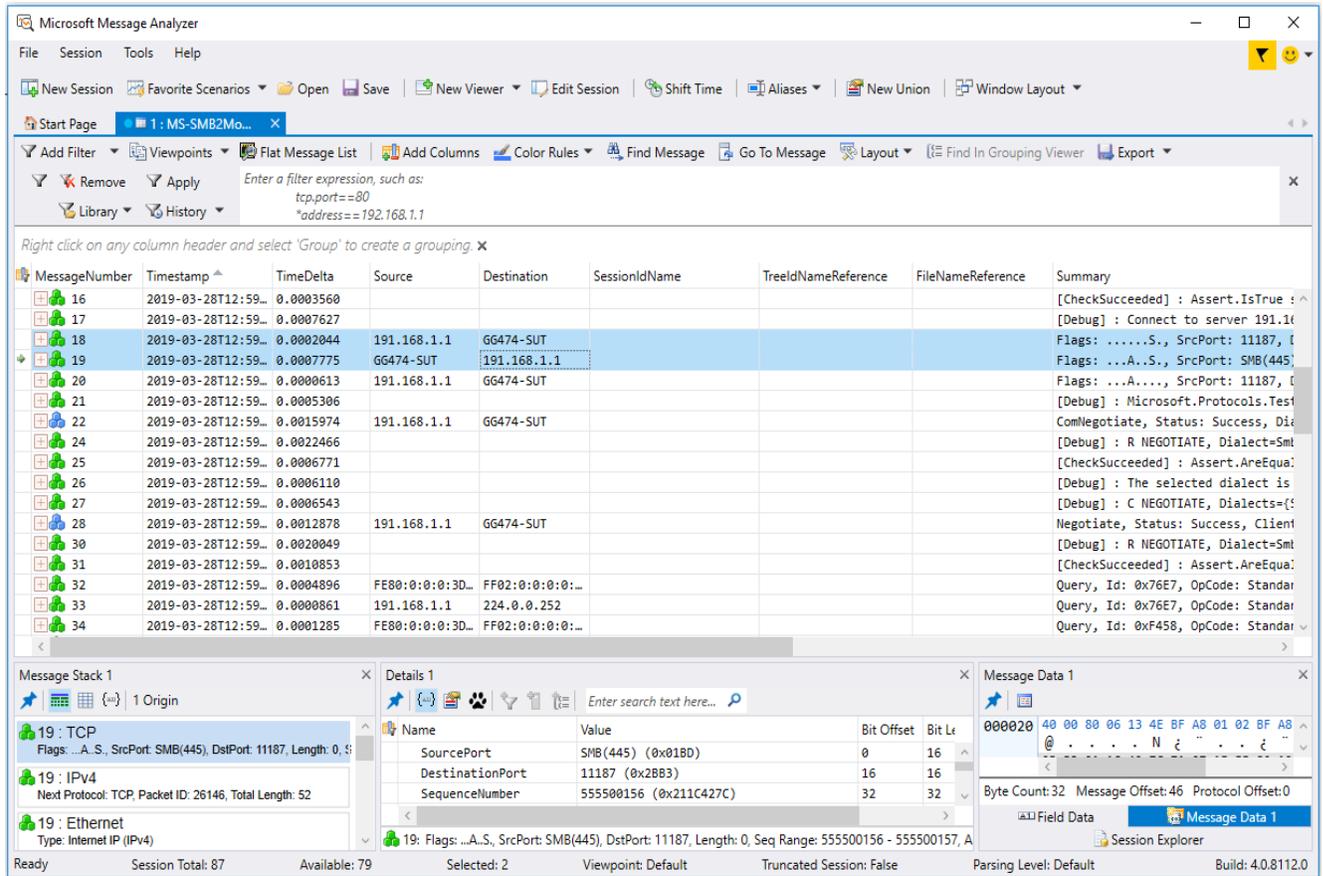


Figure 26. Message Analyzer : Analysis Grid display of ETL Test Case data

7.0 More Information

This section contains additional information about Resources that may be helpful if you wish to dive deeper into the subject matter to which you have been introduced in this Tutorial.

7.1 Resources

The following resources contain advanced information that is related to this Tutorial. Consult this information only if you are prepared to engage with advanced and complex technologies:

- [File Server Protocol Family Test Suite User Guide](#) — a complete guide to setting up the File Server Protocol Test Suite, including software installation and instructions for configuring the test network, Workgroup or Domain test environment, Driver and SUT computers, and the Protocol Test Manager. URL:

<https://github.com/Microsoft/WindowsProtocolTestSuites/blob/staging/TestSuites/FileServer/docs/FileServerUserGuide.md#3.4>

- **File Server Protocol Family Test Suite protocols** — the full File Server Test Suite is designed to test implementations of the File Server protocol family, which includes the following protocols that are documented on the Microsoft [Technical Documents](#) site:
 - [\[MS-SMB2\]](#) — Server Message Block (SMB) Protocol Version 2 and 3
<https://msdn.microsoft.com/en-us/library/cc246482.aspx>
 - [\[MS-FSRVP\]](#) — File Server Remote VSS Protocol
<https://msdn.microsoft.com/en-us/library/hh554852.aspx>
 - [\[MS-SWN\]](#) — Service Witness Protocol
<https://msdn.microsoft.com/en-us/library/hh536748.aspx>
 - [\[MS-DFSC\]](#) — Distributed File System (DFS): Namespace Referral Protocol
<https://msdn.microsoft.com/en-us/library/cc226982.aspx>
 - [\[MS-SQOS\]](#) — Storage Quality of Service Protocol
<https://msdn.microsoft.com/en-us/library/mt226249.aspx>
 - [\[MS-RSVD\]](#) — Remote Shared Virtual Disk Protocol
<https://msdn.microsoft.com/en-us/library/dn393384.aspx>
 - [\[MS-FSA\]](#) — File System Algorithms
<https://msdn.microsoft.com/en-us/library/ff469524.aspx>
- **Test Suite design specifications** — to learn more about **Test Suite** design, see the following documentation:
 - [MS-SMB_ServerTestDesignSpecification](#)
<https://github.com/Microsoft/WindowsProtocolTestSuites/tree/staging/TestSuites/MS-SMB/docs/>
 - [File Server Protocol Family Test Design Specification](#)
<https://github.com/Microsoft/WindowsProtocolTestSuites/blob/staging/TestSuites/FileServer/docs/FileServerTestDesignSpecification.md>
 - [Authentication Protocol Server Test Design Specification](#)
https://github.com/Microsoft/WindowsProtocolTestSuites/blob/staging/TestSuites/FileServer/docs/Auth_ServerTestDesignSpecification.md
 - [MS-FSA Protocol Server Test Design Specification](#)
https://github.com/Microsoft/WindowsProtocolTestSuites/blob/staging/TestSuites/FileServer/docs/MS-FSA_ServerTestDesignSpecification.md
 - [MS-SQOS Protocol Server Test Design Specification](#)
https://github.com/Microsoft/WindowsProtocolTestSuites/blob/staging/TestSuites/FileServer/docs/MS-SQOS_ServerTestDesignSpecification.md