RDP Client Endpoint Test Suite

Version 1.0

Lab Tutorial

January 2021

Abstract

Training in RDP Test Suite Technology

Author

Greg Gille | Microsoft Senior Content Developer

Microsoft Protocols Test Team

WDG GRC Protocols Content Team

Table of Contents

Table of Figures

Introduction

This Tutorial provides step-by-step instructions for connecting to and configuring the **RDP Client Endpoint Test Suite (RDPCETS)**, in preparation for executing predefined **Remote Desktop Protocol (RDP)** **Test Cases** that exercise various functions of specific RDP protocols in Remote Desktop scenarios. Following test execution, you will perform some basic analysis of test results. You will also be provided with the test environment assets that facilitate the configuration and execution of tests in a virtual session.

Although you can manually configure the test environment, execute Test Cases with command scripts (.ps1 and .sh), and review test results analysis from the command console, this Tutorial focuses rather on using a graphical user interface (GUI) tool known as the **Protocol Test Manager (PTM)** to perform these tasks and summarize results while making use of automation to simplify configurations and improve task efficiencies.

The **PTM** and the **RDPCETS** are already installed on a Driver computer **VM** where you will conduct the testing, and to which you will have access via a **Remote Desktop Services** connection.

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

Note

The **RDPCETS** is also referred to in this Tutorial as the **RDP Client Test Suite** or just the **Test Suite**.

Test Suite Training Audience

The primary audience for the **RDP Client Test Suite** training is software developers who have little or no experience with running protocol Test Suites, RDP protocol support personnel, and even those who already have significant experience with RDP technologies. Each can benefit from taking this course.

Other audiences can include IT professionals and others who may be interested in learning about protocol Test Suites.

Training Environment

This Tutorial is primarily intended for use in an RDP IO Lab session during a Microsoft-hosted event with support and instructional personnel on hand for assistance. However, this RDP IO Lab session Tutorial is also available to anyone who wishes to use it for testing RDP implementations. It will also continue to be available to event attendees who want to further sharpen their skills with RDP testing technologies following their initial training session at a related RDP event.

It is with this in mind that the document was written to be a stand alone volume that you can consult at any time, with the content of such quality that it should enable you to walk through any testing scenario you might be required to perform on your own.

This document and the RDP technologies it describes for testing RDP implementations is available on GitHub here.

Goals of the Test Suite Training

The primary goals of this training session are as follows:

- Learn some basics concepts of RDP protocol communications with an RDP connection sequence.
- Obtain a preliminary understanding of the **RDP Client Test Suite** test environment.
- Learn how to use the **Protocol Test Manager** to manage the configuration, filtering, test execution, and test results analysis features that are available in the **RDP Client Test Suite**.

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	10
Configure Test Suite and run Test Cases	30
Analyze test results | Q & A	20
Total time	**60**

Glossary Definitions

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

Driver computer — a computer on which the **RDP Client Endpoint Test Suite (RDPCETS)** is installed and configured. After installation of the Test Suite is complete, you can run the preconfigured test cases that are part of the **RDP Test Suite** from this computer. In the test environment, this computer typically runs a Windows Server 2019 operating system.

Generic Conference Control (GCC) — a high-level protocol for passing control information during a session between remote computers, which includes setting up and managing the session. In addition, the GCC protocol is used by applications to coordinate independent use of the **MCS** channels.

Image Quality Assessment (IQA) — an algorithm that takes an arbitrary image as input and produces a quality score as output. A common type of IQA algorithm uses a reference image free of distortion to compare with a distorted image to measure the quality and produce a difference score.

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, typically for the purpose of servicing application functions via sending client requests and replying with server responses, or vice versa.

In the context of running the RDP Client Test Suite, the implementation typically refers to a proprietary RDP protocol implementation that is being tested on the **SUT computer** with Test Cases that are specifically designed to put such a protocol through its paces.

Isolated network — for testing protocols with the **RDPCETS**; 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.

Multipoint Communication Service (MCS) — defines a multipoint data delivery service for use in audiographics and audiovisual services. Enables multipoint-aware applications to send data to a group of recipients or to a subset of such a group. Can enforce uniformly sequenced reception of data to all recipients.

Profile — a file generated by the **Protocol Test Manager (PTM)** component of the Test Suite that represents a configuration of Test Cases that you optionally create and store as a ***.ptm file in a specified 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 or potentially changing SUT environment.

Protocol — a set of rules or procedures that define how data is transmitted between computers and processed. 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.

See Protocol Communications for a simple example.

Protocol data unit (PDU) — information delivered as a package among peer entities of a network that may contain control information, address information, or data.

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

- Detecting and assessing 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 and properties.
- Running the Test Cases.
- Analyzing the test results.

See Configuring the RDP Test Suite for further information.

PTMCli — a command line executable that enables you to execute the Test Cases of a saved Profile from the command console.

See Using a Command Line Tool to Execute Test Cases for further information.

RDP Client — the **SUT computer** that hosts the RDP protocol implementation/s to be tested.

RDP Client Endpoint Test Suite (RDPCETS) — a set of preconfigured, software-coded Test Cases that exercise features of a subset of the RDP family of protocols that are associated with RDP services. Contains the framework for configuring the test environment, executing tests of RDP features, and facilities for analyzing test results.

RDP connection sequence — a series of messages in request/response format that are used to set up an RDP connection between an RDP Client and RDP Server computer.

See Protocol Communications for further information.

RDP Server — also known as the RDP Session Host, the **Driver computer** contains the **RDP Client Test Suite** from where you perform test configuration, execution, and analysis tasks.

Remote Desktop Services (RDS) — sometimes referred to as simply the Remote Desktop Protocol, **RDS** is a Microsoft Windows component that enables users to control a remote computer or virtual machine over a network connection. Such a connection is maintained between an RDP Client computer and a server known as a Remote Desktop Session Host.

The **RDS** architecture allows Windows applications, resources, and the desktop of the computer running **RDS** to be accessible to any remote client computer that supports the Remote Desktop Protocol. The RDP protocol enables the negotiation of client and server settings for use during the duration of an RDP connection, so that input, graphics, and other data can be processed through the exchange between client and server.

RDS consists of a family of RDP protocols that include a core protocol known as MS-RDPBCGR, along with many other RDP protocols that are referred to as extensions, such as MS-RDPEUSB, MS-RDPEVOR, MS-RDPRFX, MS-RDPEI, and so on.

SUT computer — the system under test (SUT) is the computer that hosts the system against which the pre-defined Test Cases are to be run by the **RDPCETS** that is installed on the **Driver computer**. Typically, the **RDPCETS** tests an implementation of one or more RDP protocols, which can be either proprietary, developed RDP implementations; or the Remote Desktop Service protocols that run on the **SUT computer** by default.

For purposes of this training and the preconfigured set up that is used, the Microsoft Remote Desktop Services on the **SUT computer** serve as the underlying implementation being tested. In the test environment for this lab training, this computer typically runs a Microsoft Windows 10 client operating system.

Test Case — an executable application hosted by the Test Suite that is designed to test unique aspects of RDP features that use the RDP protocols within the context of an RDP client and RDP server communication session.

 **Note**

An **RDPCETS** installation can contain hundreds 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.

Concepts

This section briefly describes the major concepts with which you will become familiar during the course of the RDP IO Lab Session Tutorial. The material begins with the basic concepts of **protocol** communication and descriptions of the test environment that you will be working with, as indicated in 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.

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 an example of RDP **protocol** communications by providing some of the **messages** that comprise the initial **RDP connection sequence** that is used to set up an RDP 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 computer** and **SUT computer**.

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

Running the Test Cases : Options — shows you how to use the PTM to manage execution of **RDPCETS** Test Cases on the **SUT computer**, as initiated from the **Driver computer**. Also includes working with **Profiles** and optionally executing Test Cases from the command line.

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

More Information — provides additional resources that support the concepts and references described in this Tutorial.

Protocol Communications

This section provides an example of **protocol** communications between an **RDP client** and an **RDP server** (session host). It shows how several types of protocol **messages** are used in the initial stages of setting up an RDP session, which is characterized by the following phases of the connection sequence:

- Initiating the RDP connection sequence

- Exchanging basic settings
- Connecting with various channels
- Commencing client/server security negotiations
- Sending secure client data to the server

A visual representation of the communication process of the previously specified connection phases is shown in Figure 1 and is accompanied by explanatory steps. The sequence of messages shown in Figure 1 are actual representations of the types of requests and responses that are typical in setting up an RDP session.

The goal of this section is to provide readers with a basic sense of how the **RDP Client Test Suite** on the **Driver computer** (server) communicates with the **SUT computer** (client).

For example, in the **RDP Client Test Suite**, an RDP **Test Case** could invoke a method to obtain confirmation as to whether standard RDP security mechanisms are in place and if encryption is being enforced. If so, this could trigger sending security information to the server which in turn might generate session keys that are used to encrypt and validate the integrity of RDP traffic. Note that features such as these could be preconfigured, coded implementations on the **SUT computer**, against which some Test Cases are run from the Test Suite on the **Driver computer**.

The figure that follows illustrates the first 5 phases of the RDP connection sequence. The full RDP connection sequence is described in section 1.3.1.1 of MS-RDPBCGR.

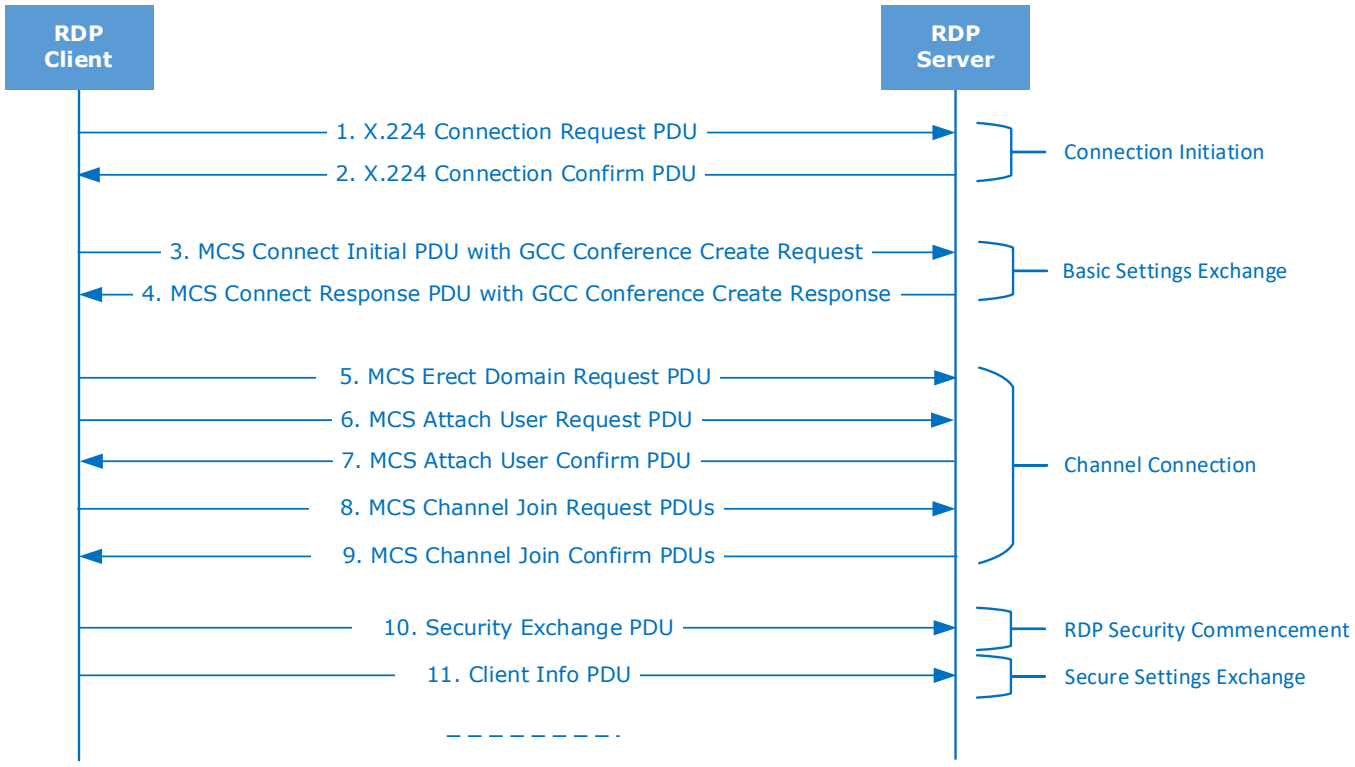


Figure 1. Partial RDP connection sequence : Example

▶ **The steps that follow describe the RDP connection sequence actions:**

1. The RDP client initiates the connection to the RDP server by sending an X.224 Connection Request **PDU**.
2. The server responds with an X.224 Connection Confirm PDU to affirm the connection request.

   Thereafter, all subsequent data sent between the client and server is wrapped in an X.224 data PDU.
3. The RDP client sends an MCS Connect Initial PDU with a GCC Conference Create Request to the RDP Server to facilitate exchange of basic settings such as core, security, and network data.

   For more information, see glossary definitions for **MCS** and **GCC**.
4. The RDP server reads the client data and sends the MCS Connect Response PDU with GCC Conference Create Response to finalize the data exchange.
5. The RDP client sends an MCS Erect Domain Request PDU and also performs the next step.
6. The RDP client sends the MCS Attach User Request PDU to attach the primary user identity to the MCS domain.
7. The server responds with an MCS Attach User Confirm PDU containing the User Channel ID.
8. The RDP client uses multiple MCS Channel Join Request PDUs to join the user channel, the input/output (I/O) channel, and all of the static virtual channels.
9. The server confirms each channel with an MCS Channel Join Confirm PDU.
10. The RDP client sends a Security Exchange PDU containing an encrypted 32-byte random number to the server; that is, if standard RDP security mechanisms are in place and encryption is enforced.

    This random number is encrypted with the public key of the server and the server generates another 32-byte random number. The client and server then utilize the two 32-byte random numbers to generate session keys which are used to encrypt and validate the integrity of subsequent RDP traffic.
11. The RDP client sends the Client Info PDU with secure client data; including username, password, auto-reconnect cookie, and so on; to the RDP server.
12. The RDP connection sequence continues in accordance with section 1.3.1.1 of MS-RDPBCGR.

To offer a simplified correlation to the foregoing, when the **RDP** Client 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 request and response message sequences shown in the previous figure.

After the RDP connection sequence and data exchange negotiations for the session are complete, the Test Suite can run a host of preconfigured Test Cases that utilize important features and functions of the RDP protocol **implementations** that comprise the system under test (SUT).

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

Test Environment Architecture

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

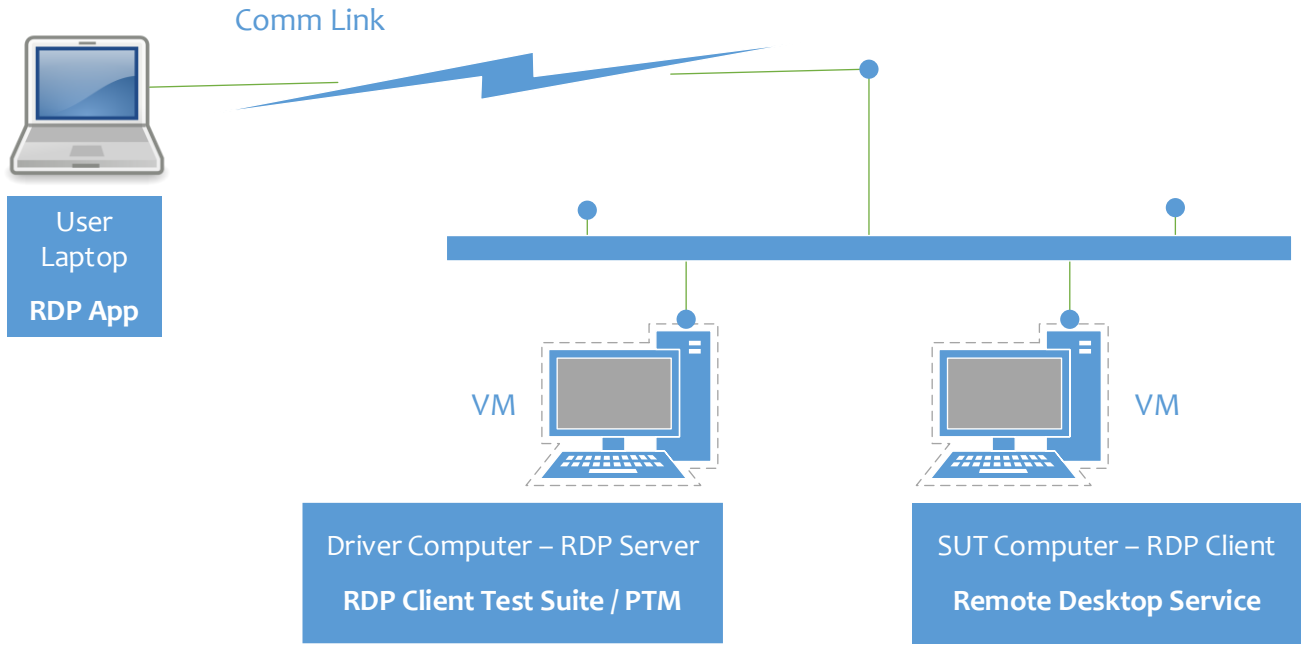


Figure 2. RDP Client Test Suite : Network Domain test environment

Note

The **RDP Client Test Suite** can also use the **Workgroup** test environment where there is no Domain Controller and the **Driver** and **SUT computer** function in a peer-to-peer configuration.

The components of the network Test Environment for the **RDPCETS** are described in the list that follows:

- **User laptop** — a Surface, workstation, or other computer from which you will **RDP** into the cloud to specified **virtual machines** in a Domain environment, where you will configure the **RDP Client Test Suite** as described in the section Configuring the RDP Test Suite for the system to be tested.
- **RDP app** — the common **Remote Desktop Protocol** application you will use to connect with the Domain environment consisting of the **Driver** and **SUT VMs**.
- **Driver computer** — a VM RDP Server that hosts the **RDP Client Test Suite** that you will configure for running **Test Cases** against the **SUT** configuration.
- **SUT computer** — a VM RDP Client computer that is pre-configured with the required RDP features to be tested by the Test Cases of the Test Suite.

Important

In this Test Environment, you will not be testing a proprietary RDP **implementation** on the **SUT computer**. Rather, the Microsoft **Remote Desktop Services (RDS)** component that normally runs on that computer will respond to the test **messages** generated by the Test Cases of the **RDP Client Test Suite**, which reside on the **Driver computer**, as shown in the figure that follows:

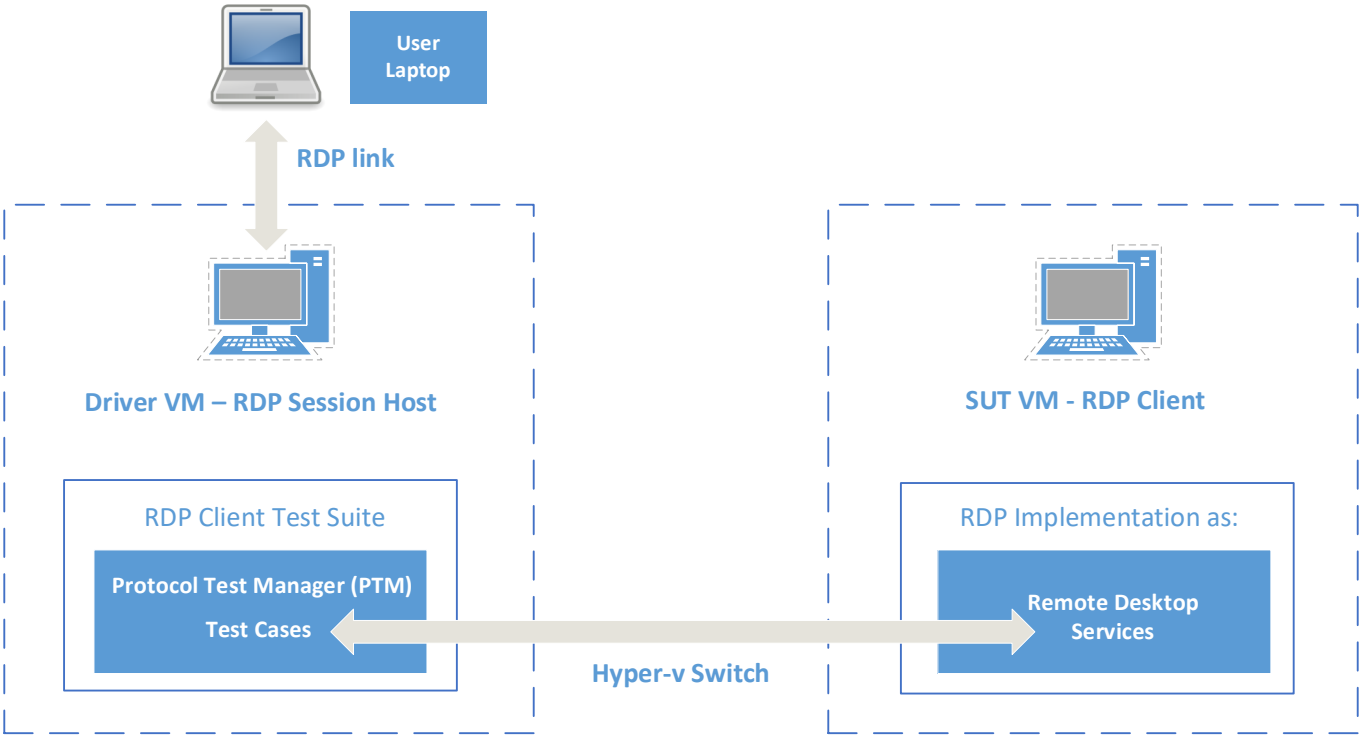


Figure 3. Test Environment : Test Cases communication path

In the section that follows, you will learn how to configure the **RDP Client Test Suite** with the use of the protocol test manager (PTM) or by loading a set of Test Cases previously saved as a **Profile**, which is described in Saving a Profile.

Configuring the RDP Test Suite

This section provides an overview of the tasks required to configure the Test Suite **Test Cases** by using the **Protocol Test Manager (PTM)** on the **Driver computer**. The list immediately below basically follows the steps of the PTM's **Configuration Wizard**, which can include performing the following tasks after starting the PTM:

- Reviewing the execution environment.
- Selecting the **Auto-Detection** option for configuring the test environment.
- Reviewing and validating user input information for connecting with the **SUT computer** (ie, the **SUT Name**, **SUT User Name**, **SUT Password**, **Trigger RDP Client by**, and so on.)
- Using PTM to detect, inspect, and validate the SUT environment for test readiness; results in reporting the protocols and features that are supported by the SUT, along with a default set of configurable properties.
- Modifying the output of the **Auto-Detection** process, by selecting/unselecting (filtering) the proposed default Test Cases that are based on assessment of the SUT environment, to create a unique test configuration.

  In this Tutorial, you will choose 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 Configure the Test Suite with PTM Configuration Wizard which follows. Thereafter, you will run the Test Cases and then perform simple analysis of test results, to obtain a basic understanding of the analysis features.

Optional Configuration Scenario

If you have already run through an execution of Test Cases and saved a **Profile** per the procedure in the section Saving a Profile, you can proceed to the section Configure the Test Suite by Loading a Profile to load a set of Test Cases with associated properties into the PTM prior to execution, instead of performing the configuration in the section that immediately follows. However, note that you will need to have the **PTM** open to the **Configure Method** tab to load an existing **Profile**.

Configure the Test Suite with PTM Configuration Wizard

The **Protocol Test Manager (PTM)** is a user interface (UI)-based tool that helps you configure and run **Test Cases**. To access the **RDP Client Test Suite** and the **PTM**, you will need to connect with the **Driver computer**. Thereafter, you can locate a PTM executable shortcut on the desktop of the **Driver computer** and start it. The procedures in this section will walk you through all the related processes.

▶ **To connect with the Driver computer VM, perform that steps that follow:**

1. Using the **Remote Desktop Connection** application on your laptop computer, connect with the **Driver computer** by entering the **Driver** computer name in the text portion of the **Computer** drop-down; also click **Show Options** and specify the domain\username in the **User name** text box.

Note

The computer and user names for these inputs should be provided to you by the Instructor of the RDP Client Test Suite Lab Tutorial.

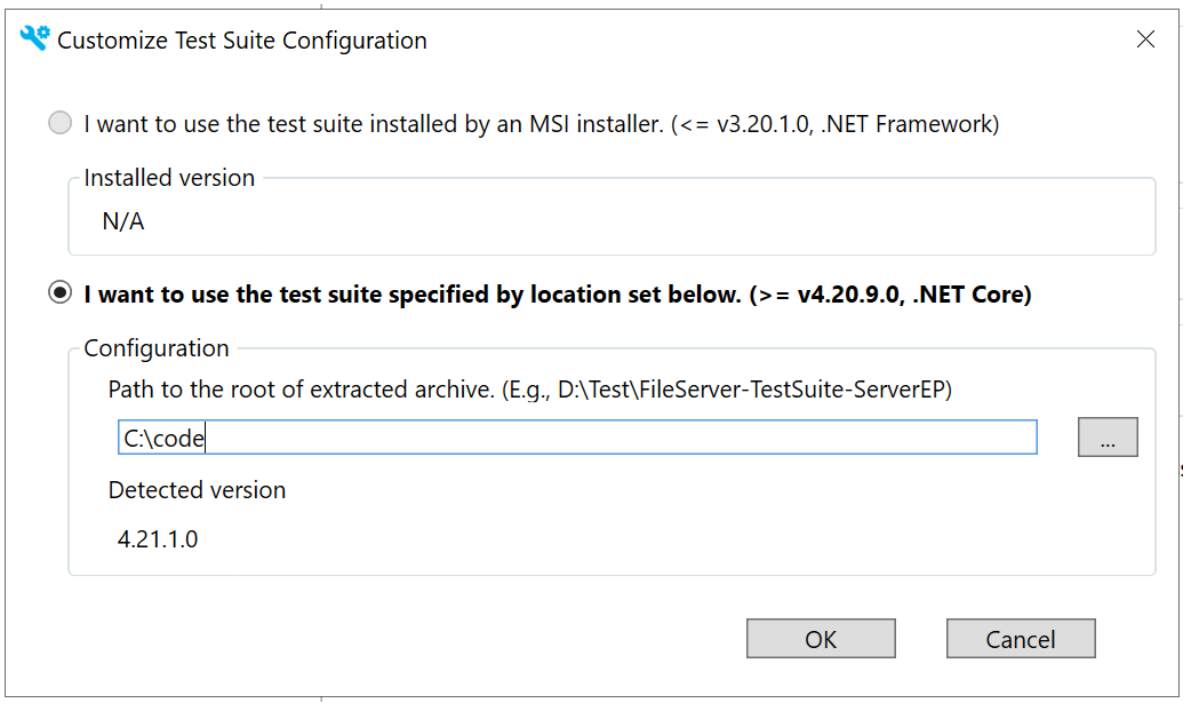
If you do not click the **Show Options** drop-down arrow to access the **User name** data entry point to provide input, you will be prompted for credentials when you connect.

2. If you are prompted for additional credentials, such as a password, enter the password that you received from the Instructor of this Tutorial.

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

1. After logging on to the **Driver computer**, double-click the **Protocol Test Manager** icon/shortcut on the desktop.

   The **Select Test Suite** tab of the PTM displays several Test Suites including **File Sharing**, **Security**, **Active Directory**, **RDP**, and so on. If you click the **Select** link in the **Remote Desktop** pane, the following **Customize Test Suite Configuration** dialog displays:



   In the dialog, the option that is selected depends upon which RDP Client Test Suite you installed:

   - **Option 1** - Version 3.20.1.0 or earlier, along with.NET Framework.
   - **Option 2** - Version 4.20.9.0 or later version of this Test Suite, along with .NET Core.

   If you installed the latest version of the RDP Client Test Suite v4.21.1.0, check that the path to the root of the extracted archive for the Test Suite is `c:\code` on the Driver computer, as shown above.

Note that the only active **Test Suite** on the opening tab of the PTM is **RDP**, as indicated by the live **Run** and **Configuration Wizard** links on the right side of the UI in the figure that follows.

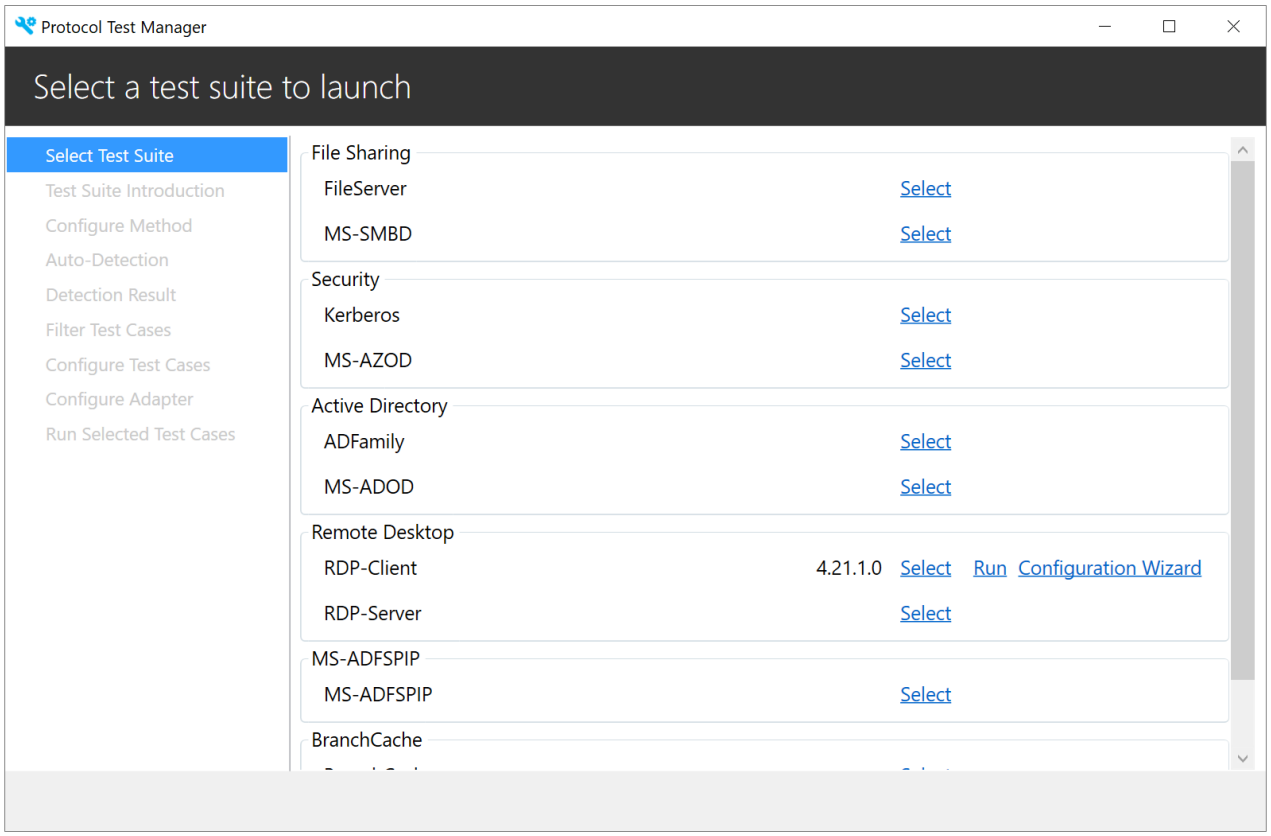


Figure 4. Protocol Test Manager : Launching the Configuration Wizard for RDP Client Test Suite

2. Click the **Configuration Wizard** link to begin the **Test Suite** configuration process.

Note

If this is for first time use, you will click the **Configuration Wizard** link to start the **RDP Test Suite** configuration process. Thereafter, while no **Test Cases** are running and if the **Test Suite** requires no further configuration changes, you can simply click **Run** to execute the existing (last) configuration of Test Cases while still retaining the existing **Test Suite** configuration settings.

At any time, you can resume configuration of the **Test Suite**, by clicking **Configuration Wizard**.

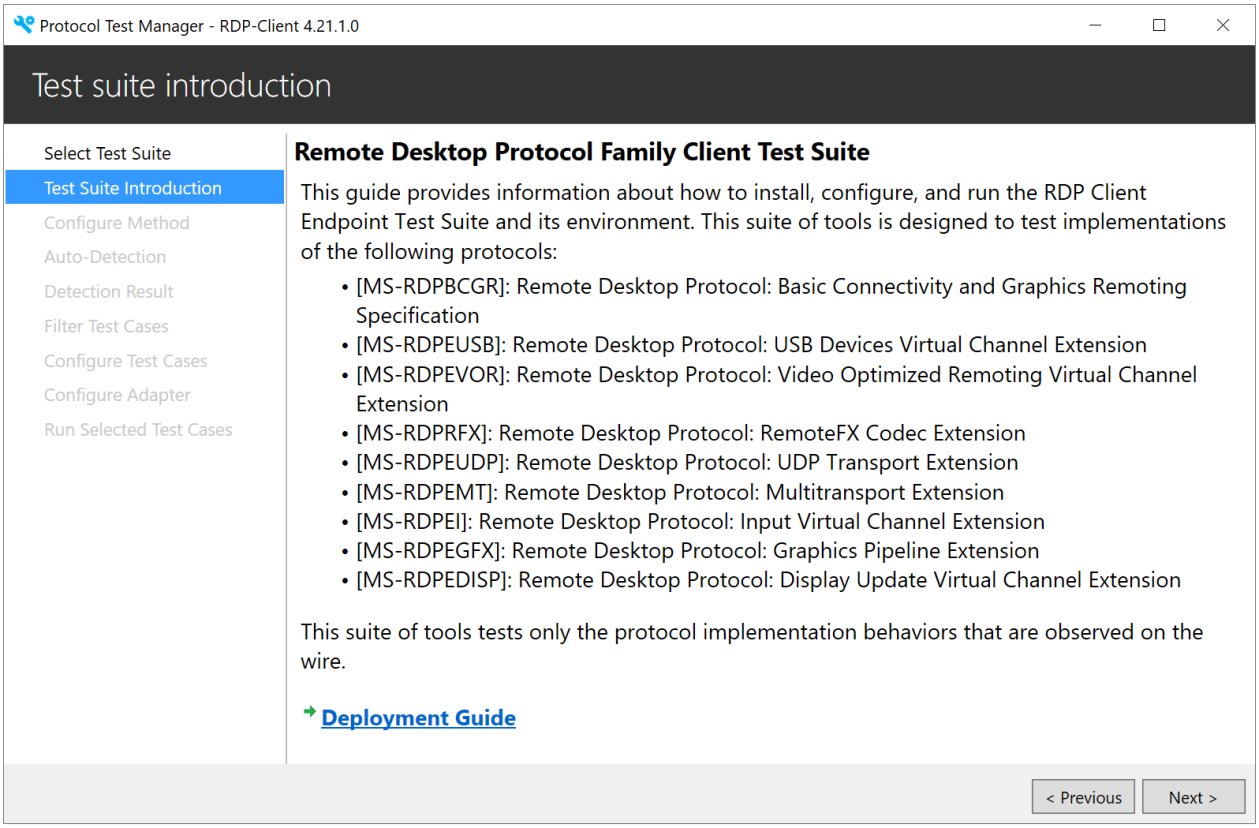


Figure 5. Protocol Test Manager : Reviewing the RDP Test Suite family of protocols

3. Read the information presented on the **Test Suite Introduction** page of the PTM Wizard and then click the **Deployment Guide** link to review general information about the test environment in which you will run your Test Cases.

   Note

   It is unnecessary to click the **Configure Environment** link after reviewing the test environment information, given that setup and configuration procedures will have already been completed by the Test Team running this RDP Lab session event.

4. Click **Next** to display the **Choose configuration method** page of the PTM Wizard.

5. On the **Choose configuration method** page of the PTM Wizard shown in the figure that follows, click **Run Auto-Detection** to start the process of retrieving the capabilities of the SUT **implementation** (also click **Yes** to the **Warning** message that displays).

   The retrieved capabilities are used to automatically configure the **Test Suite** and select a default set of Test Cases for the assessed **SUT** environment.

   Caution

   Unless you are adept at Microsoft RDP testing technologies and you understand how to assess the **SUT** environment with respect to manually choosing the correct Test Cases and configuring their properties, you are advised to *not use* the **Do Manual Configuration** option on the **Choose configuration method** page of the PTM Wizard.

However, note that you can use the **Load Profile** option to execute Test Cases from a **Profile**, as described in Configure the Test Suite by Loading a Profile.

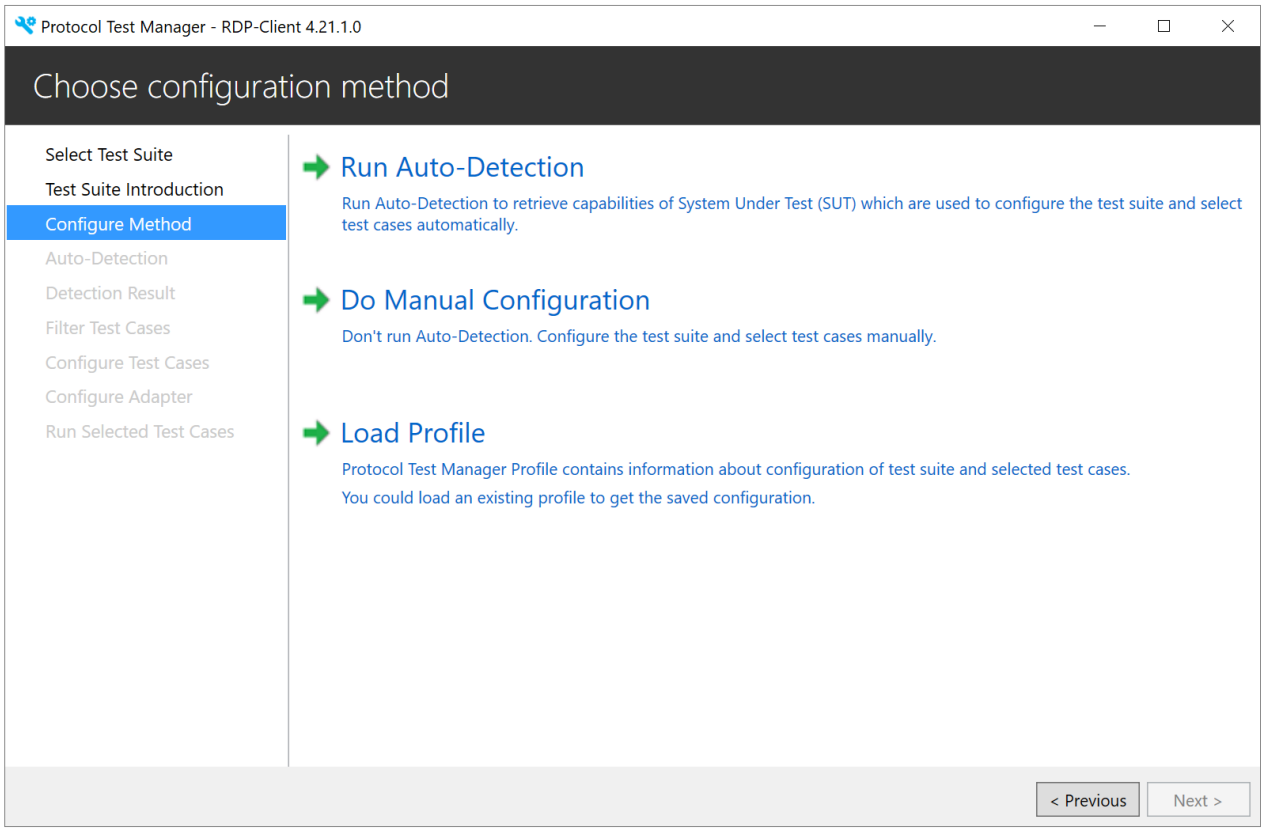


Figure 6. Protocol Test Manager : Choosing the Test Suite configuration method

After you click **Run Auto-Detection**, the **Auto-Detection** page of the PTM Wizard displays, as shown in the figure that follows.

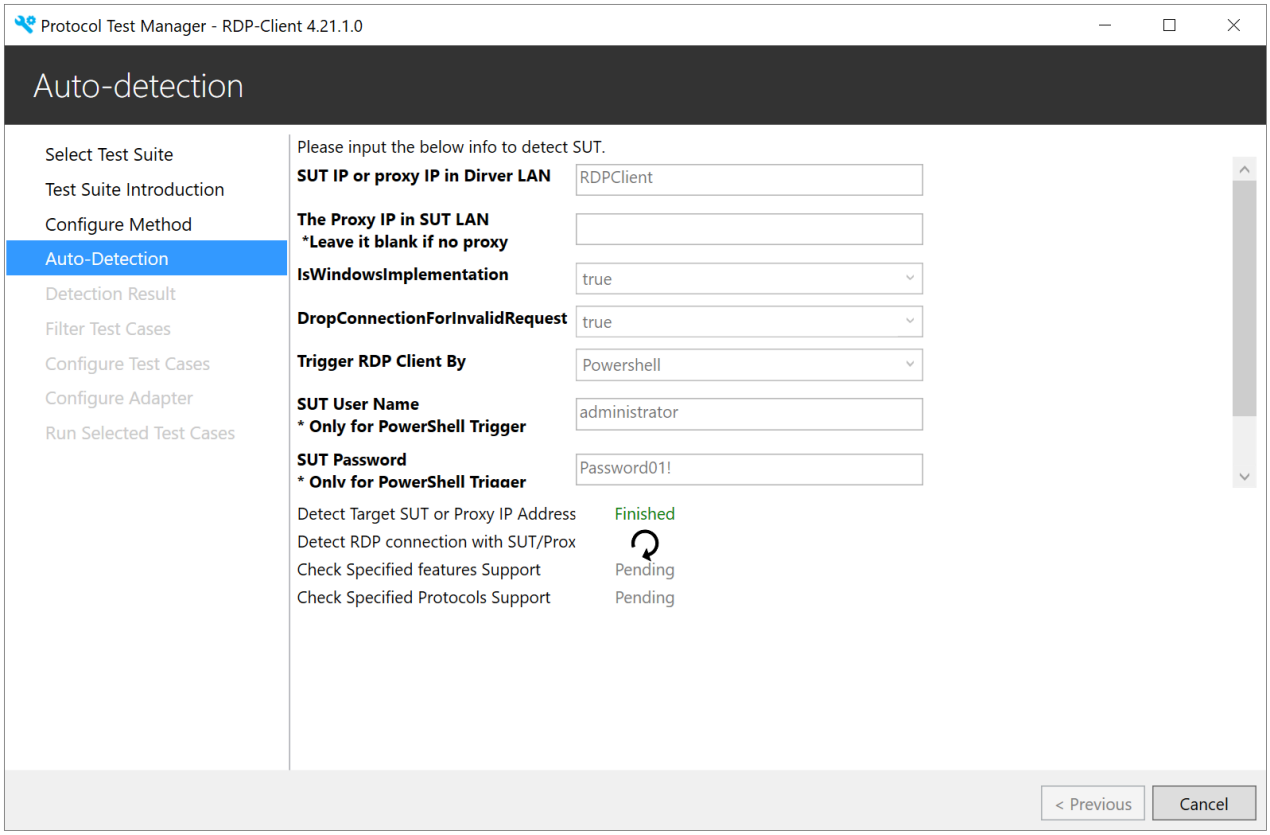


Figure 7. Protocol Test Manager : Verifying SUT connection parameters

6. On the **Auto-Detection** page of the PTM Wizard, verify that the values for such items as **SUT Name**, **SUT User Name**, **SUT Password**, and **Agent Listen Port** are correct in your environment, to ensure that the PTM can connect with the **SUT computer** and perform an accurate assessment of SUT capabilities.

   Important

   If there appears to be issues with these values, please consult the Instructor of this Tutorial, or send an email to protmail@microsoft.com.

7. When verification is complete, click the **Detect** button in the lower-right sector of the **Auto-Detection** page.

   As the detection process proceeds, you will see the **Pending** indications beneath the data entry fields change to **Finished** as detection progress continues.

8. After the initial detection process completes, click **Next** (formerly the **Detect** button) to display the **Detection Result** page of the PTM Wizard, as shown in the figure that follows:

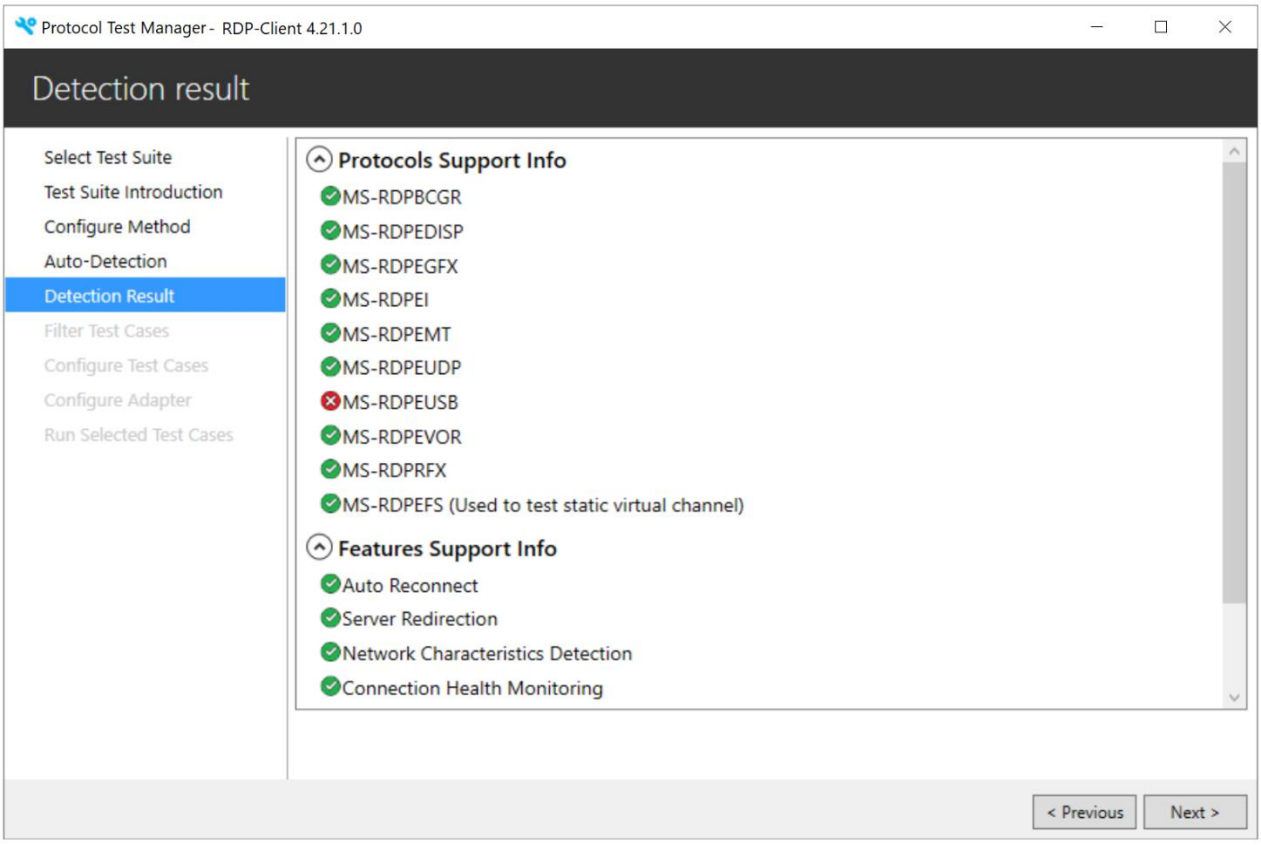


Figure 8. Protocol Test Manager : Reviewing the Detection Results support information

9. In the **Detection Result** page of the PTM Wizard, review the data that appears under the **Protocols Support Info** and **Feature Support Info** nodes and note the status indicators that display following the Auto-detection process.

Note

In the previous figure, **protocols** and features such as **MS-RDPEUSB**, will be flagged with an icon that indicates its detection status. If you click it, one of the following status descriptions is displayed in the lower sector of the UI:

- — "*<Protocol | Feature>* **is found not supported after detection**"
  An unsupported protocol or feature appears in italics and is flagged with this icon. If selected Test Cases attempt to test them, this status indicator will be used to indicate an error in the test results. **For more information**, see Test Results Output Status Indicators.
- — "*<Protocol | Feature>* **is found supported after detection**"
  Indicates that the assessed entity is supported by the Test Suite in the **SUT** environment.
- — "**Because of detection failure,** *<Protocol | Feature>* **is not detected successfully**"
  A failure to detect a protocol or feature prevents Test Case execution against those entities in the **SUT** environment.

10. On the **Filter Test Cases** page of the PTM Wizard shown in the figure that follows, review the selected Test Cases, as they are the suggested samples being used for the RDP Client Test Suite Lab Tutorial.

    If you are using this Tutorial outside the hands-on Lab Tutorial environment, you will need to select at least one of the sub-nodes in the **Priority** category and one in the **Protocol** category at a minimum, to activate and display associated Test Cases for a test run. The selection categories consist of the following:

    - **Protocol** — reflects the RDP protocols under test (for **SUT** implementation).
    - **Priority** — enables you to specify the type of tests to run (**BVT** or **NonBVT**).
    - **Enable Supported Feature** — activates Test Cases for supported features discovered in **Auto Detect**.

      Note

      Because a feature or protocol is supported does not necessarily mean you *must* enable it, as PTM allows you to test only the features and protocols you want to. Also, take note that if a Test Case belongs to multiple categories, it is listed in each category.
    - **Specific Requirements** — enables you to specify the *Interactive* testing mode. See *Test Run Pre-Configuration* topic in the RDP Client Test Suite User Guide for further information.

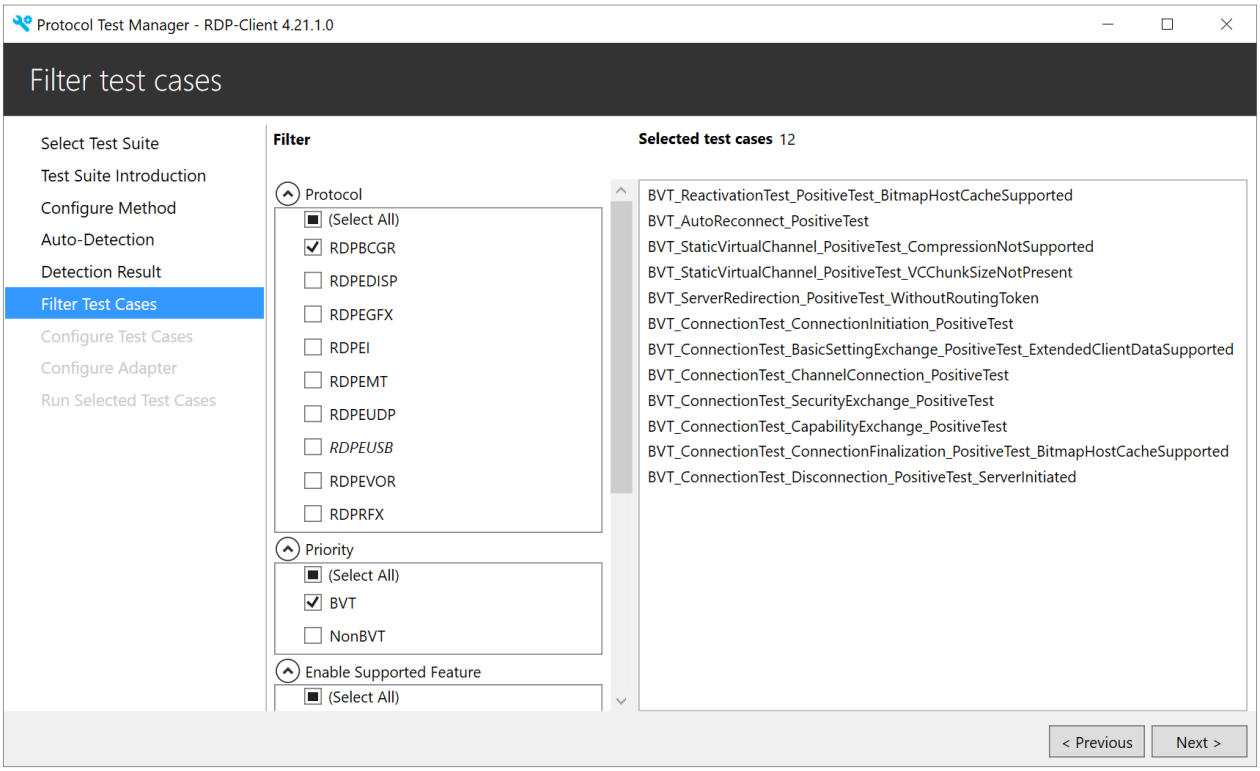


Figure 9. Protocol Test Manager : Selecting (filtering) Test Cases

11. Create the suggested Test Case configuration for this Tutorial in the steps that follow:

a. In the four categories described in the previous step, *unselect* the **Select All** node in each category.

b. In the **Protocol** category, select the protocol **RDPBCGR**.

c. In the **Priority** category, select the **BVT** check box.

   By selecting **BVT**, you will run verification Test Cases for the **Protocol** you selected, otherwise, **NonBVT** tests for that **Protocol** would apply.

   Note

   The **RDP Client Test Suite** includes a set of basic tests known as build verification tests (BVTs). The BVTs are a set of Scenario S1 Test Cases, as specified in the RDP Client Test Design Specification, that together perform verification tests that confirm whether the **Test Suite** is properly configured and ready to continue running Test Cases.

d. Select the following features in the **Enable Supported Feature** category:

   — AutoReconnect

   — ServerRedirection

   — StaticVirtualChannel

e. Observe that the Test Cases that apply to the selected protocol and features display in the Test Cases list view along with the number of tests indicated in the **Selected Test Cases** label above the list, as similar to the previous figure.

12. Click **Next** to display the **Configure Test Cases** page of the Wizard, as shown in the figure that follows.

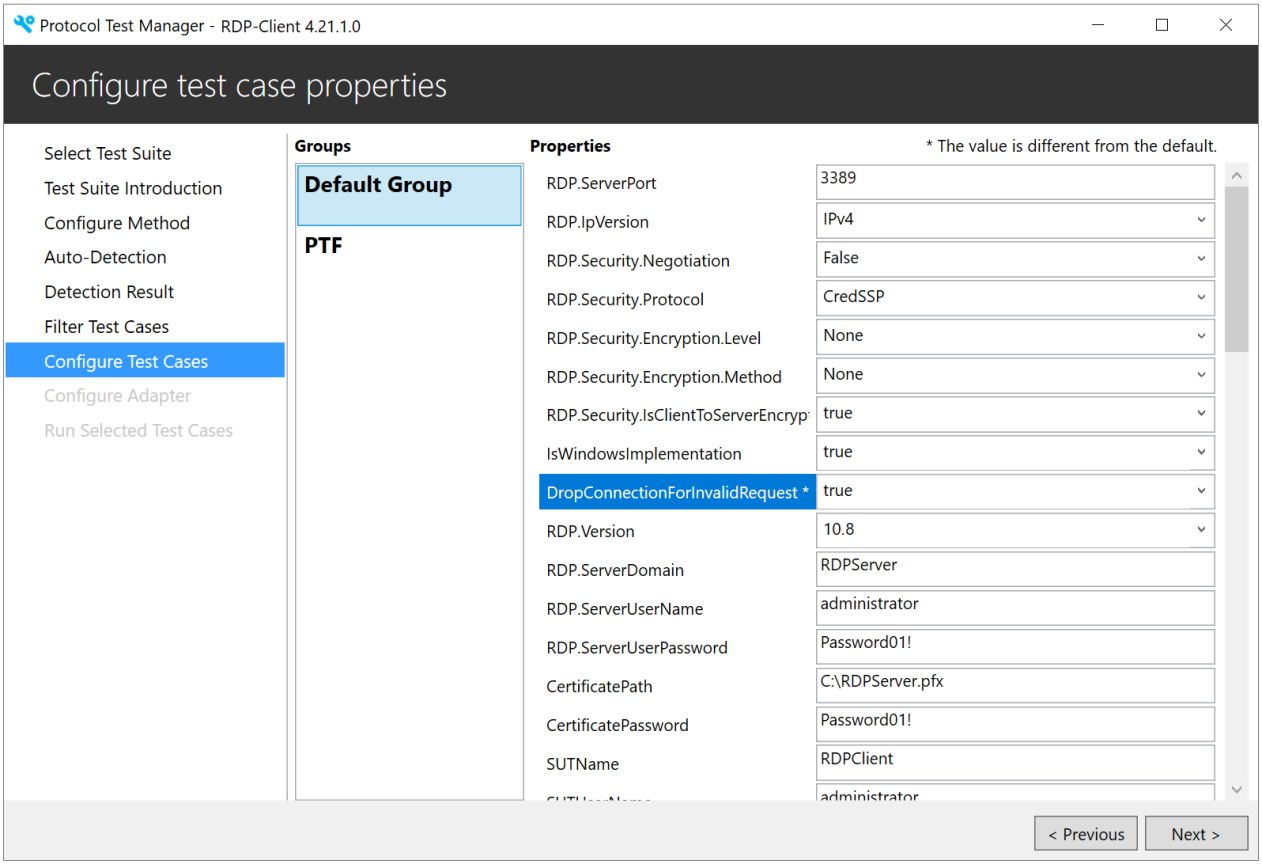


Figure 10. Protocol Test Manager : Reviewing/configuring Test Case properties

13. On the **Configure Test Cases** page, click the **Default Group** to display the RDP properties.
14. Scroll through the property values and consult with the Instructor of this Lab session if you encounter any value that obviously seems invalid.

    More Information

    To learn more about the meaning of RDP Test Case properties and their typical values, you may consult Configure the Test Suite in the RDP Client Test Suite User Guide for further information.

15. Click **Next** to display the **Configure Adapters** page of the Wizard, as shown in the figure that follows.

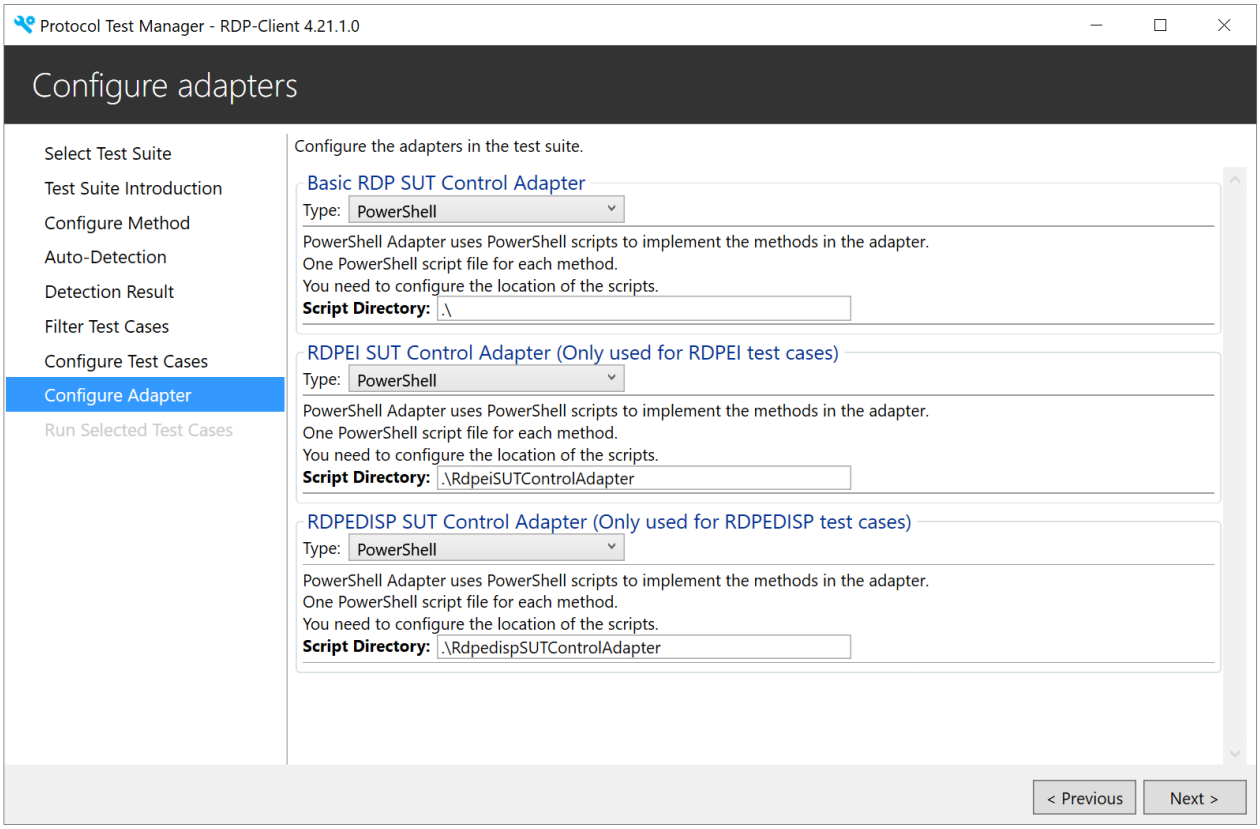


Figure 11. Protocol Test Manager : Configuring the Adapters

16. Click the drop-down arrow for each Adapter and select **PowerShell,** which is the default setting, if not already selected.

    This setting means that PowerShell scripts define the tasks performed by the SUT Control adapter for the RDPEI and RDPEDISP protocols and other basic processes.

17. When complete, click **Next** to display the **Run selected test cases** page of the PTM Wizard, as shown in the figure that follows.

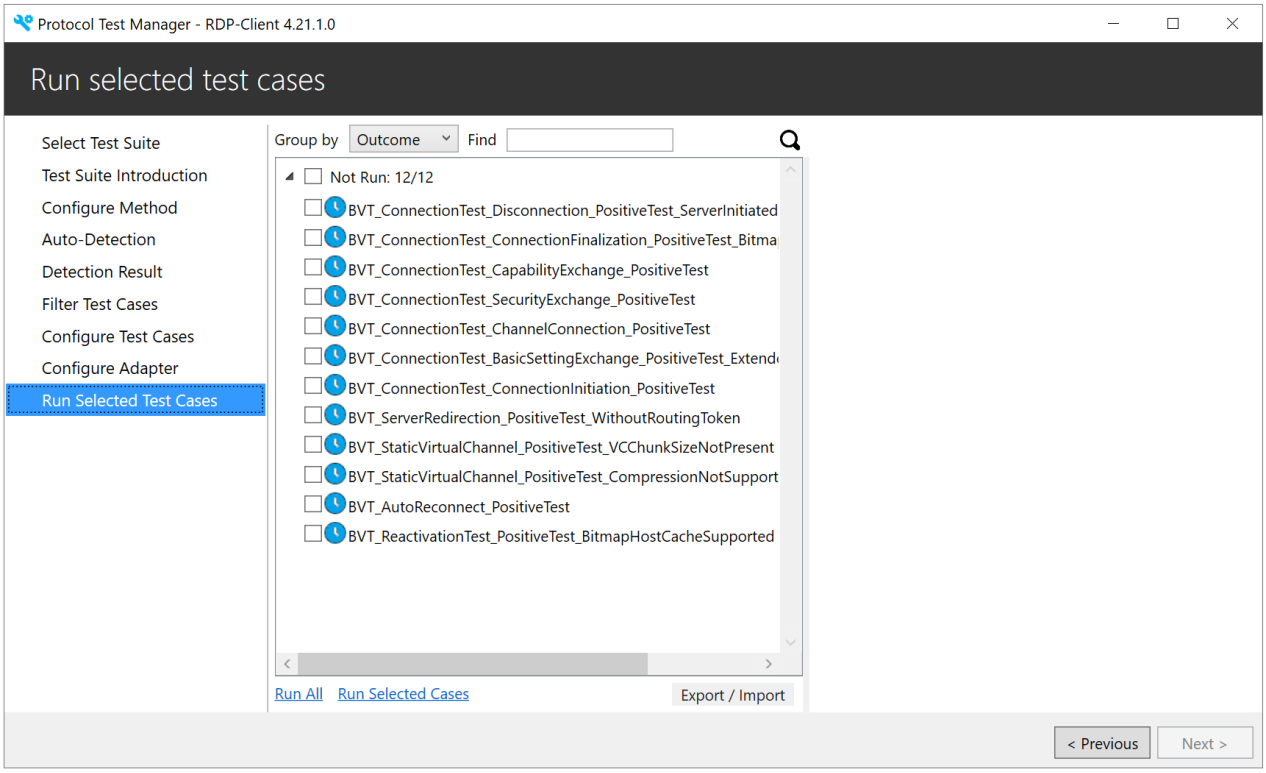


Figure 12. Protocol Test Manager : Test Case Execution Configuration

This completes the configuration phase of the PTM. If you are ready to run your Test Cases, go to Running the Test Cases : Options.

Configure the Test Suite by Loading a Profile

If you have previously saved one or more PTM **Profiles**, as described in Saving a Profile, follow the procedure in this section to load a **Profile** with the **Protocol Test Manager Load Profile** option.

This results in configuring the test environment with a specified **Test Case** configuration (including supporting property values) that you previously saved as a **Profile**. After you have loaded the Test Cases of a saved **Profile** into the PTM, you can execute the Test Case configuration as is, or modify it before executing.

- **To load a PTM Profile:**

1. On the **Configure Method** tab of the **PTM**, click **Load Profile**, select a target **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 Saving a Profile, navigate to that location.

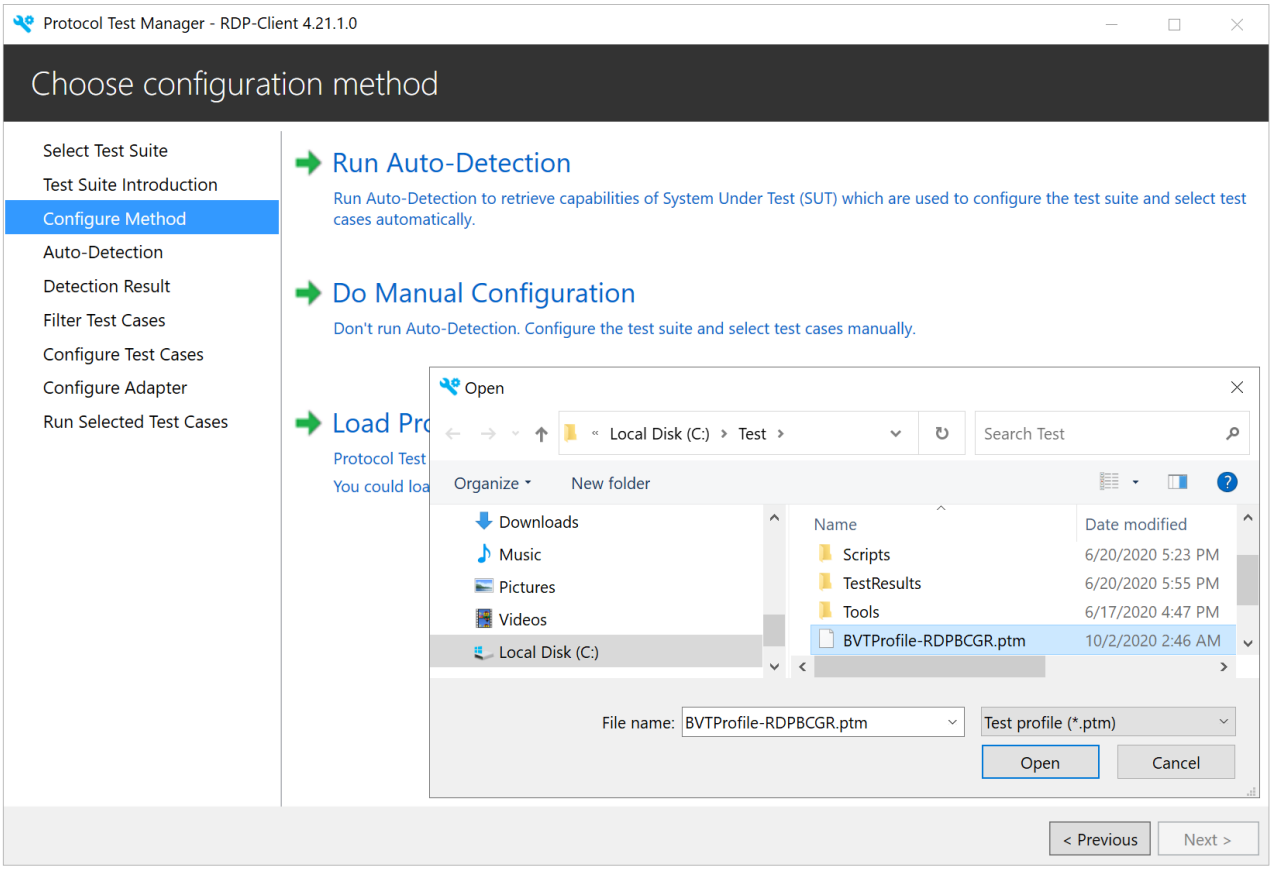


Figure 13. Protocol Test Manager : Loading an existing Profile

2. Observe that the PTM immediately opens to the **Run Selected Test Cases** tab with the **Not Run** checkbox selected, unexpanded, and showing the number of tests in the **Profile**.
3. Click the down arrow of the **Not Run** checkbox to expand it, as shown in the figure that follows, and confirm that the Test Cases of the loaded **Profile** appear as expected.
4. To run the **Profile** Test Cases, do one of the following:

   - Run all Test Cases from the **Profile** by selecting the **Not Run** checkbox and clicking the **Run All** command beneath the Test Case list view, as shown in the figure that follows.

     Note that all Test Cases will run whenever you click the **Run All** command, even if the cases are not specifically selected. Also, all Test Cases will run if you *select* all Test Cases and then click **Run Selected Cases**.

   - Run only specifically-selected Test Cases from the **Profile**, by first *selecting* the loaded Test Cases you want to run (or by *unselecting* those you do not) and then click the **Run Selected Test Cases** command beneath the Test Case list view, as shown in the figure that follows.

     Note

     The Profile Test Cases should run the same way as any other set of Test Cases do with all the status indications that occur during normal Test Case execution progress. When complete, you can proceed to **Analyzing the Test Results Data**.

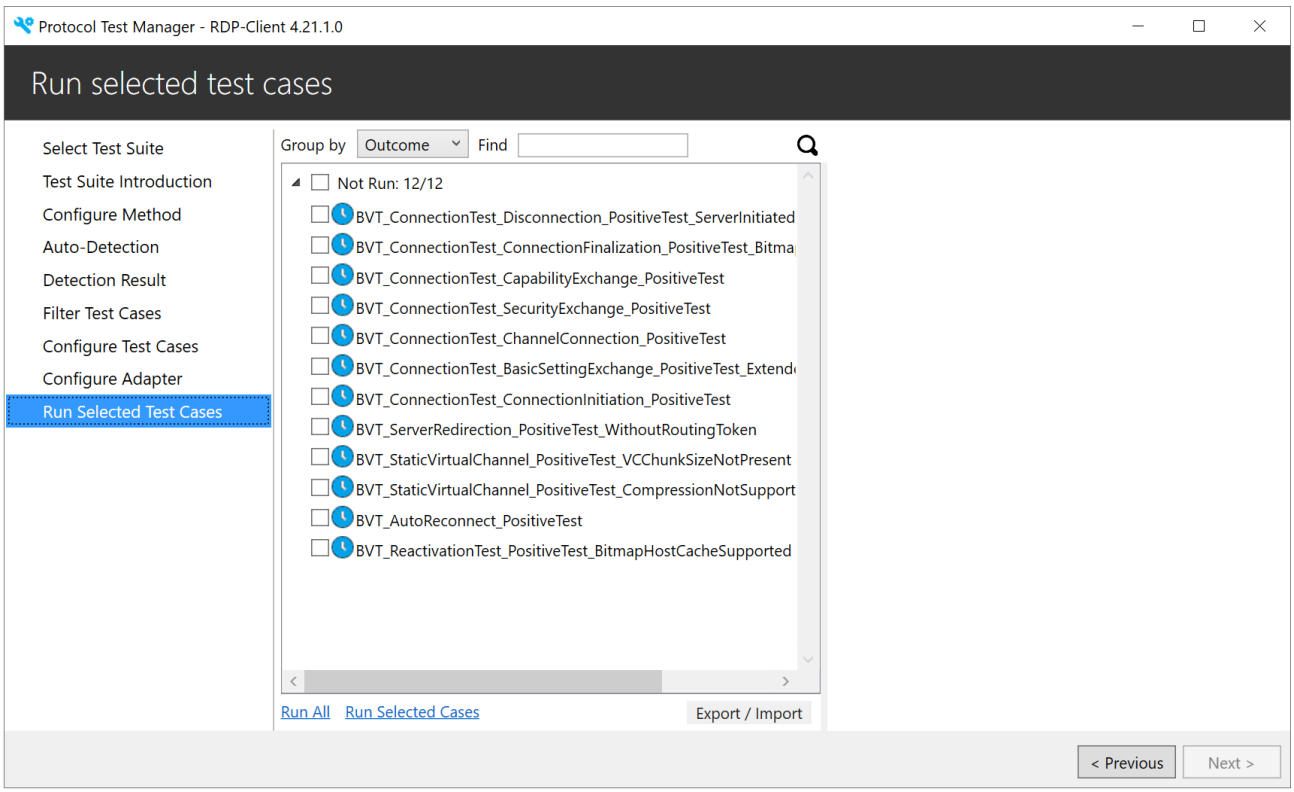


Figure 14. Protocol Test Manager: Confirming the validity of Profile test cases configuration

Important

If you want to preserve any *selection* or property changes you made to the list of Test Cases, you will need to save them by overwriting the existing **Profile** or by creating a new one. To do so, use the **Export / Import** feature under the Test Case list view and refer to the topic Saving a Profile for further procedural information.

Running the Test Cases : Options

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 option to run Test Cases:

Important
Do not execute Test Cases at this time.
Only perform execution from the procedure To run the Test Cases 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 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, given that you selected only some of them.

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

  If you select this option, only the Test Cases that are specifically *selected* will be executed.

Monitoring Test Case Execution Results Indicators

After you start **Test Case** execution, two command shells display test execution data in scrolling format, which includes success, failure, and other indications. At the same time the tests are displaying at the command line, you can also view high level results in the following three categories in the Test Cases list view of the PTM.

As test case execution progresses, you can observe these categories being incrementally updated:

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

  For example, if a property set from the Configure Test Cases tab of the PTM is incorrectly configured, or a Test Case conflicts with an unexpected or invalid property value, that Test Case can finish as **Inconclusive**.

Important
Whenever any of the core/preconfigured Test Cases in the test environment do not support certain features, errors will appear in the command console and PTM UI with respect to methods that failed when attempting to test those features.

Notwithstanding errors from actual failures, you can avoid this in an outside/proprietary test environment by running only Test Cases that support the features of that environment.

However, note that the RDP lab Tutorial environment in which you are running Test Cases in this session avoids feature errors 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 RDP Client Test Design Specification.

Using PTM to Execute the Test Cases

If you have completed setting up the test environment as described in Configure the Test Suite with the PTM Configuration Wizard, you can start **Test Case** execution by following the steps of the procedure that follows.

▶ **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 Cases** link below the Test Cases list view, as shown in the figure that follows.

   Given that the **Not Run** checkbox is now selected, all the Test Cases under the checkbox will also be selected and will run when you click the **Run Selected Cases** link, as described earlier in Running the Test Cases : Options.

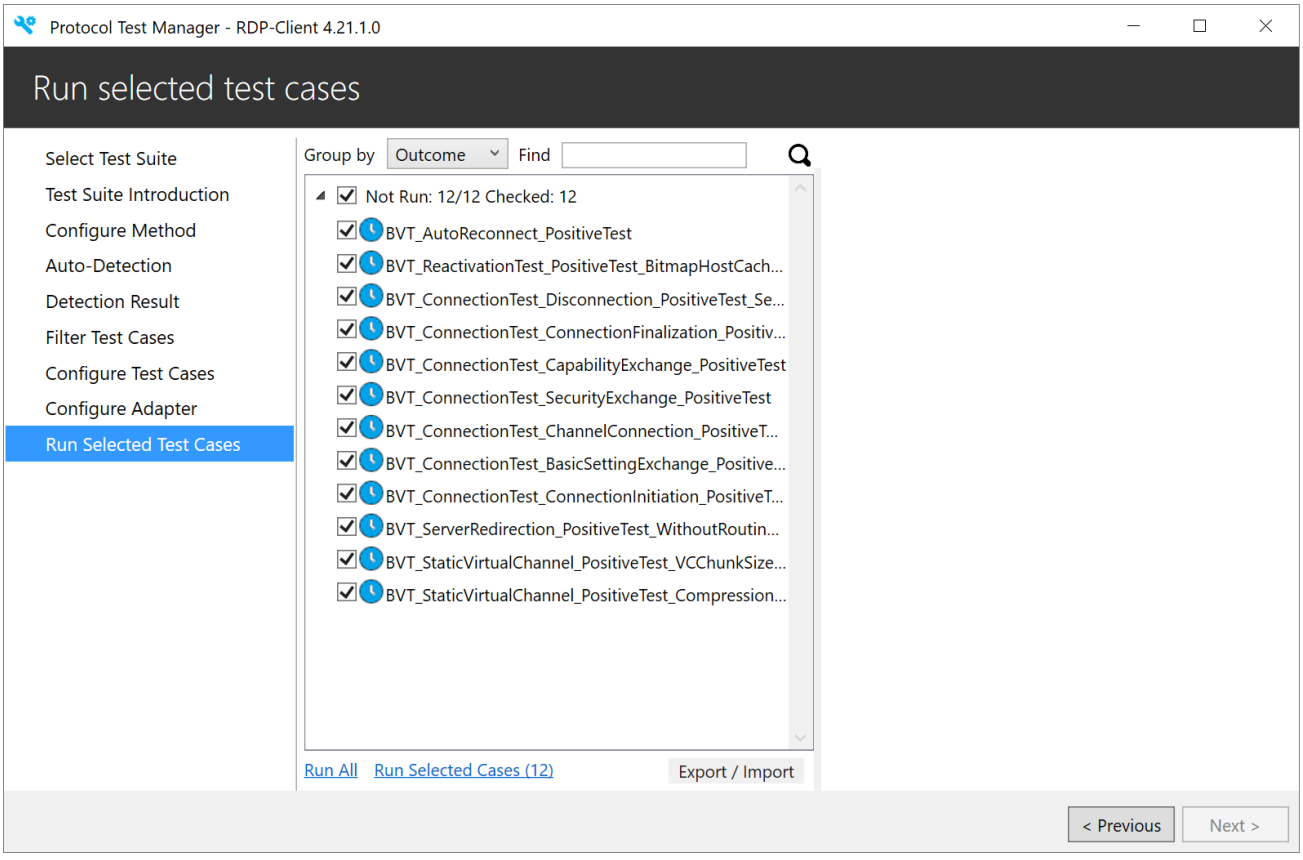


Figure 15. Protocol Test Manager : Starting Test Case execution

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 nullify all selections.

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 Cases** context menu command afterwards, all Test Cases listed under the **Not Run** checkbox will execute.

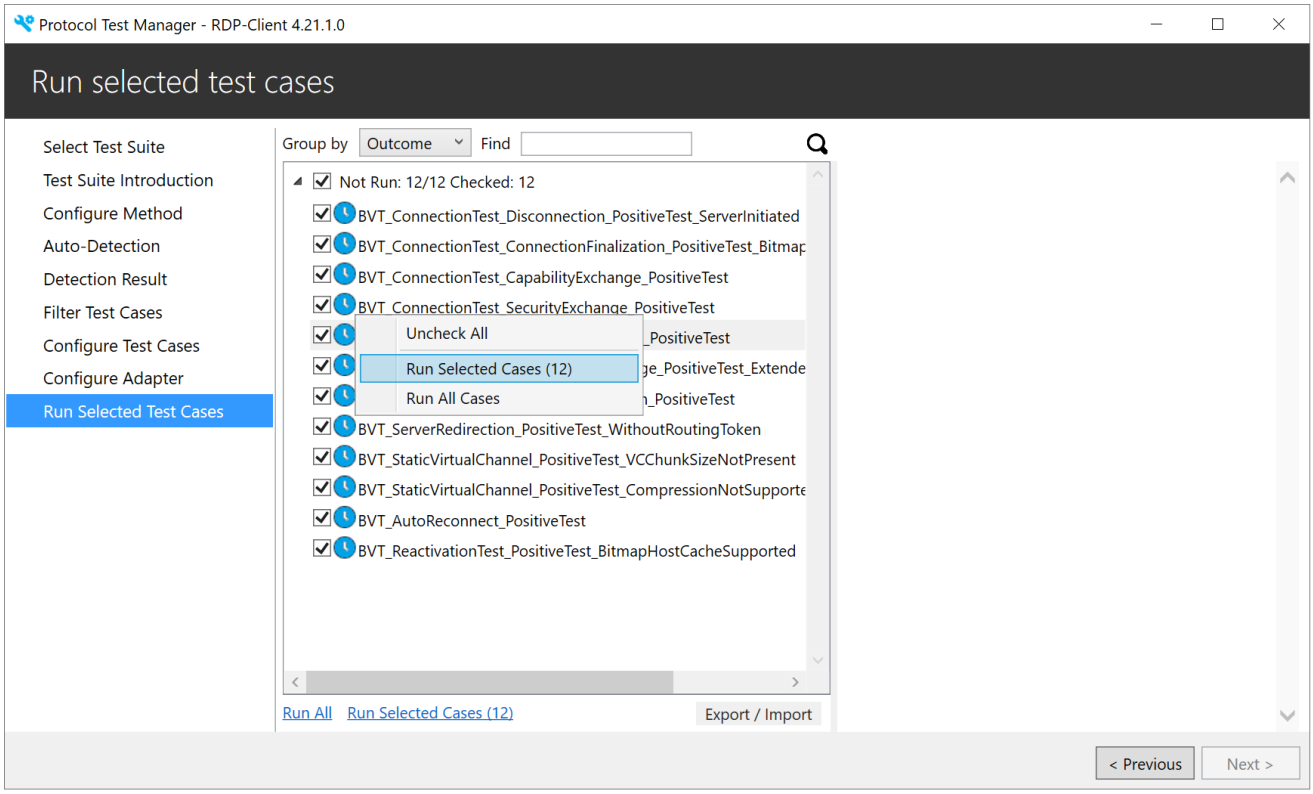


Figure 16. Protocol Test Manager : Test Case list view context menu commands

2. To display the context menu indicated in the preceding figure, click the expansion node (if not already expanded) 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** context menu item to begin execution of selected Test Cases.

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, as described in Analyzing the Test Results Data.

To learn about analyzing the results of Test Case execution, which includes descriptions of what was tested by any Test Case that you select, proceed to Analyzing the Test Results Data. Otherwise, review the topics that follow to learn about saving a **Profile** and using the **PTMCli** command line tool to execute the Test Cases of a **Profile**.

Saving a Profile

After you complete test execution based on a particular **Test Case** configuration, as described in Using PTM to Execute the Test Cases, you have the option to use the PTM to save the configuration as a **Profile** that you can re-run on demand by doing one of the following:

- Locating the **Profile** in a specified save directory and loading it directly into the PTM (see Configure the Test Suite by Loading a Profile).
- Executing the **Profile** from the command line (see Using a Command Line Tool to Execute Test Cases).

Thereafter, you can analyze the data in the PTM output results, command console, HTML, and/or text log formats.

Note

You can also optionally save a **Profile** *before* actually running your Test Case configuration, but in this instance, you will not have the advantage of knowing if the Test Case configuration performed well or not, 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 making changes in a **protocol implementation**.

In order to utilize a **Profile** in the recommended manner, you are advised to run the **RDP Client Test Suite** Test Cases at least once and then save a **Profile** that extracts the selected Test Cases and related property value information that you want.

Important

If you want to save specific Test Cases for a **Profile**, you must *select them* before your save the **Profile**, otherwise the **Profile** will not contain any Test Cases.

Running Test Cases on Demand — after you save a **Profile**, you can then use it in subsequent re-runs of the profiled test environment where you use the **Load Profile** option in PTM, as described in Configure the Test Suite by Loading a Profile; or you can use the `PTMCli` command line tool to execute the Profile's Test Cases from the command line (see Using a Command Line Tool to Execute Test Cases).

Saving Test Cases to a Profile — save a **Profile** by performing the steps that follow.

▶ **To save a Profile that encapsulates the currently selected Test Case configuration:**

1. In the **Run Selected Test Cases** page of the PTM Wizard, *select* the Test Cases that you want in your **Profile**.

   Important

   Please ensure that you *select* Test Cases before saving your **Profile**, or no Test Cases will execute. This is especially important if you intend to use the **PTMCli** command line tool with the -s switch, which specifies that only *selected* Test Cases will be executed in the **Profile**. See Using a Command Line Tool to Execute Test Cases for more information about **PTMCli**.

2. Just below the PTM Test Cases list view on the **Run Selected Test Cases** tab of the PTM Wizard, click the **Export/Import** link, as shown in the figure that follows, and then click the **Save Profile…** drop-down menu item to save the *selected* Test Cases of the current test configuration along with all the underlying related property values.

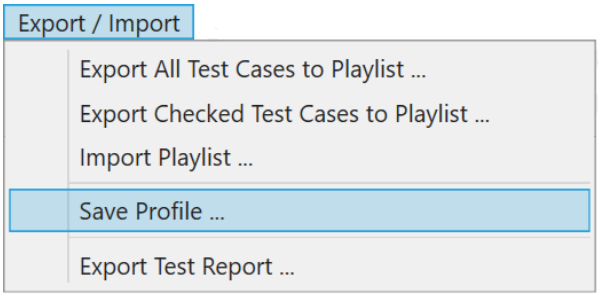


Figure 17. Protocol Test Manager : Saving a Test Cases Profile

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

3. Specify an appropriate name for the **Profile**, preferably a name that is associated with the type of tests in the **Profile**, for example, "BVT".

4. Close the open **Save** dialog.

Export / Import Menu Options

The other options in the **Export / Import** dialog are defined as follows:

- **Export All Test Cases to Playlist** — use to export *all* Test Cases to a playlist in a format that is supported by Microsoft Visual Studio.
- **Export Checked Test Cases to Playlist** — use to export *checked* Test Cases to a playlist in a format that is supported by Microsoft Visual Studio.
- **Import Playlist** — use to import a previously exported playlist.
- **Save Profile** — use to save all your Test Cases and properties for running on demand later.
- **Export Test Report** — use to export a Test Report for reviewing and or archiving test results.

Note
Test Cases can also be saved in text file format.

Using a Command Line Tool to Execute Test Cases

The **Protocol Test Manager** enables you execute the **Test Case** configuration of a previously saved **Profile** (see Saving a Profile) 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**, which you can use on a Windows, Linux, or macOS platform. You can download the archive here and extract the contents to a specified folder.

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

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

- **To load and execute 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 directory location you specified above on the **Driver** computer where the PtmCli application should reside.
3. At the command line, type the following command string:

```
dotnet PTMCli.dll -p <profilepath> -s
```

Note
The -p switch requires the directory path specification to a saved **Profile** and the -s switch in this command enforces execution of *only the Test Cases that were selected* at the time the **Profile** was saved.

4. Press the **Enter** key on your keyboard and 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:

```
dotnet PTMCli.dll --help
```

PtmCli Help

The help switches for the PtmCli application are described as follows:

-p --profile — a required argument that specifies the path to the Profile to run.

-t --testsuite — a required argument that specifies the path of the Test Suite to run.

-s --selected — default state is false where all test cases in the Profile are executed; otherwise, when this switch is specified, only the test cases selected when the Profile was saved are executed.

--categories — specifies the value-separated categories of test cases to run. This parameter overrides the Profile test cases.

-r --report - specifies the result file for a report that is to be written to.

-f --format — default format is plain text; valid values are: plain, json, and xunit.

--outcome — default values are Pass, Fail, and Inconclusive (comma or space-separated). Specifies the outcome value of test cases to be included in the report file.

-d --debug — default state is false, otherwise when specified, PtmCli debugging is enabled. Note that the log file can be found under the current directory where you are running PTMCli.

--help — displays this help screen.

--version — displays version information.

Multi-instance Support

PTMCli supports multi-instance use, meaning that you can run multiple PTMCli executables at the same time. For example, you an open two command consoles on a Windows platform and execute the commands in each window, as follows:

dotnet PTMCli.dll -p C:\test1.ptm -t C:\code\RDP-TestSuite-ClientEP

dotnet PTMCli.dll -p C:\test2.ptm -t C:\code\RDP-TestSuite-ClientEP

The outcome will be two sets of results under the following directory on the **Driver** computer:

```
C:\core\bin\HtmlTestResults\
```

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 analysis features that can quickly point you to what the causes of a failure might be, such as the following:

- Primary status indicators
- Logging information tags
- Test status icons
- Stack messages
- Error messages

Utilizing these analysis features of the PTM 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 test data that displays in the **PTM**.

Primary Status Indicators

The primary PTM status indicators for Test Case results appear as the **Passed**, **Failed**, and **Inconclusive** check box nodes (see Using PTM to Execute the Test Cases) that can each contain a portion of the total Test Cases that were run. You can obtain further Test Case results status details by selecting any Test Case, at which point you will see a display of Test Case result details line-by-line in the **StandardOut** category to the right of the Test Case list view. In the event that one or more Test Cases have errors, additional **Error** categories will appear when such a Test Case is selected.

The primary status indicators and Test Case result details are described in the sections that follow.

Note

In this section, the analysis will initially continue to use the `Profile#1_BVTScenario.ptm` **Profile**, the test results for which have already appeared in several figures in Using PTM to Execute the Test Cases.

However, when driving down into analysis of Test Cases that have **Failed** status, test results from the `TestProfile#2_FailScenario.ptm` **Profile** will be used in this Tutorial to facilitate an error analysis process example, as in section Evaluating a Failed Test Case in PTM.

Test Results Output Status Indicators

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

Table 2. Test results status indicators

Status Indicator	Description	UI Location
Test Case primary status indicators	▪ — indicates the Test Case status is **Not Run**.	These indicators appear immediately to the right of each **Test Case** check box *after* test execution to indicate the test result status, with exception of the following:

	▪ — indicates Test Case status as **Test In Progress**. ▪ — indicates the executed Test Case status as **Passed**. ▪ — indicates the executed Test Case status as **Inconclusive**. ▪ — indicates the executed Test Case status as **Failed**.	▪ The blue **Not Run** status indicator — appears *prior* to Test Case execution. ▪ The green **Test In Progress** status indicator — appears *during* Test Case execution.
Test results status display summary	Results display in either of these configurations: 1. The number of Test Cases that Passed, Failed, or were Inconclusive. 2. The number of Test Cases that executed in specific RDP categories, where the number of tests that were run in each category are identified.	1. The first results display configuration is shown as expandable **Passed**, **Failed**, and/or **Inconclusive** results category* check boxes. Appears in the Test Cases list view area of the PTM when selecting the **Outcome** item in the **Group by** drop-down list, as shown in next figure. 2. The second results display configuration is broken down into various categories such as RDP test type, protocol version, implementation being tested, and RDP feature, for example the **AutoReconnect** feature, where the number of Test Cases that were run are identified in each category. Appears in the Test Cases list view area of the PTM when selecting the **Category** item in the **Group by** drop-down list, as shown in the figure following the next.
Start Time and **End Time**	Exposes the overall duration of Test Cases execution.	Appears in the upper-left sector of the right-hand test results log 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 log 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 can appear in the **StandardOut** category of results.
Debug output data	Informative data that is displayed in tags in the	Includes data that displays in information tags that include the following:

	StandardOut category of results. See StandardOut Category Status Indicators for more information.	▪ **[TestInProgress]** ▪ **[Comment]** ▪ **[Debug]** ▪ **[CheckPoint]** ▪ **[CheckSucceeded]** ▪ **[CheckFailed]** ▪ **[TestStep]** ▪ **[TestPassed]** ▪ **[TestFailed]**
Error output data	Holds data that you can analyze to identify the source of Test Case failures.	Appears in the **ErrorStackTrace** and **ErrorMessage** category of results.

* 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.

Primary Test Case Result Indicators : Examples

This section provides examples that illustrate **Test Case** results in the primary **Passed** and **Failed** indicator categories. The view in each of these categories can be further manipulated by applying the **Outcome** or **Category** grouping configuration that is accessible just above the Test Case list view in the **PTM**.

You can also manipulate the data by isolating specific Test Cases or groups of Test Cases that can be discovered with the use of a common search term that matches part of the Test Case names. For further information, see Using the Find Feature to Locate Test Cases.

Test Case Results with Mixed Status: Outcome View

The figure that follows illustrates a test result status of **Passed** for 10 of the RDPBCGR BVT Test Cases contained in the PTM profile `Profile#1_BVTScenario_v4.20.1.0.ptm`, while 2 Test Cases returned with **Inconclusive** status. In this figure, the **Group By** drop-down list is set to the **Outcome** view.

Note

In scenarios where all Test Cases pass, only the **StandardOut** log result details to the right of the Test Case list view will appear for any selected Test Case. In other words, the **ErrorStackTrace** and **ErrorMessage** results log categories do not display in this case.

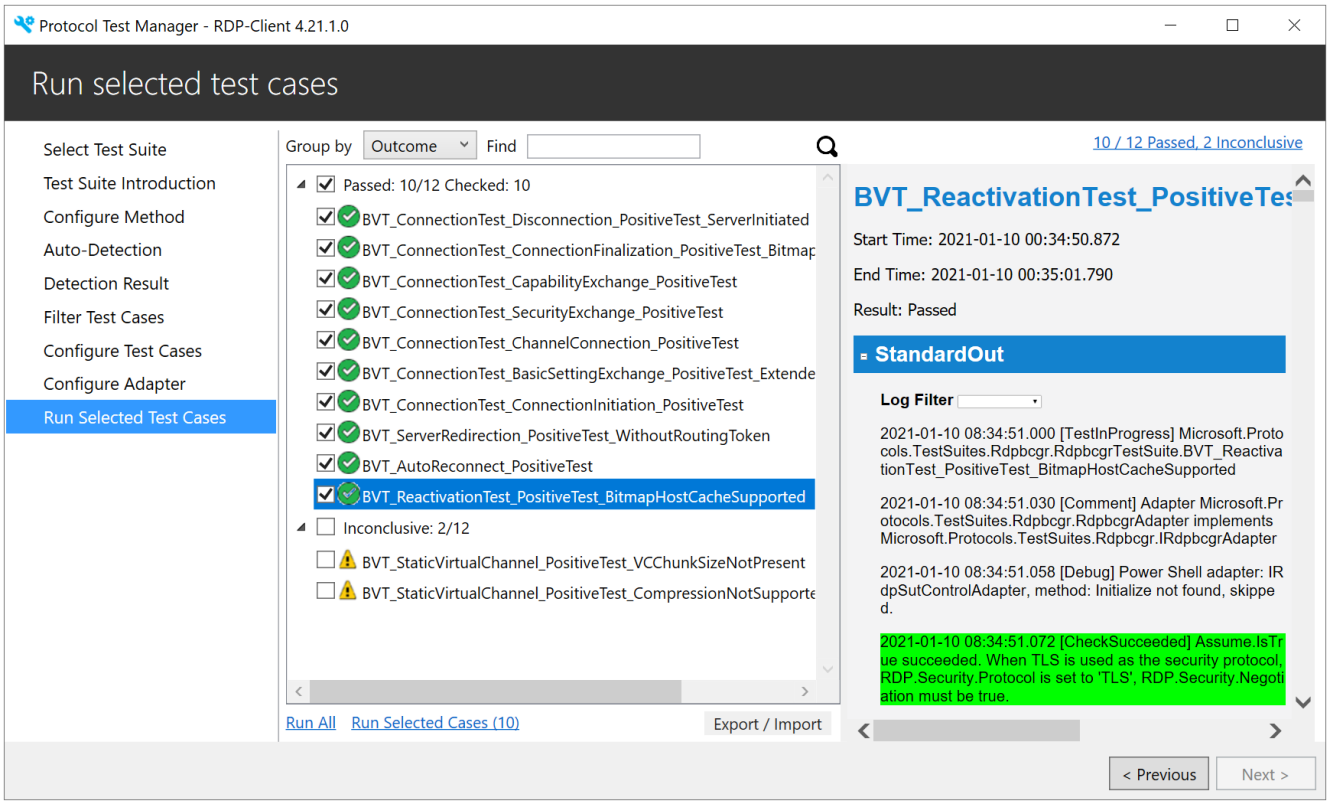


Figure 18. Protocol Test Manager : Test Results 'Outcome' grouping

Test Case Results with Mixed Status : Category View

The figure that follows is an example of the **Category** grouping of Test Cases — grouped by RDP features (autoreconnect, server redirection, etc), test type (BVT), RDP protocol version (RDP7.0), and protocol implementation (RDPBCGR); for the results of Test Cases contained in the PTM profile `Profile#1_BVTScenario_v4.20.1.0.ptm`.

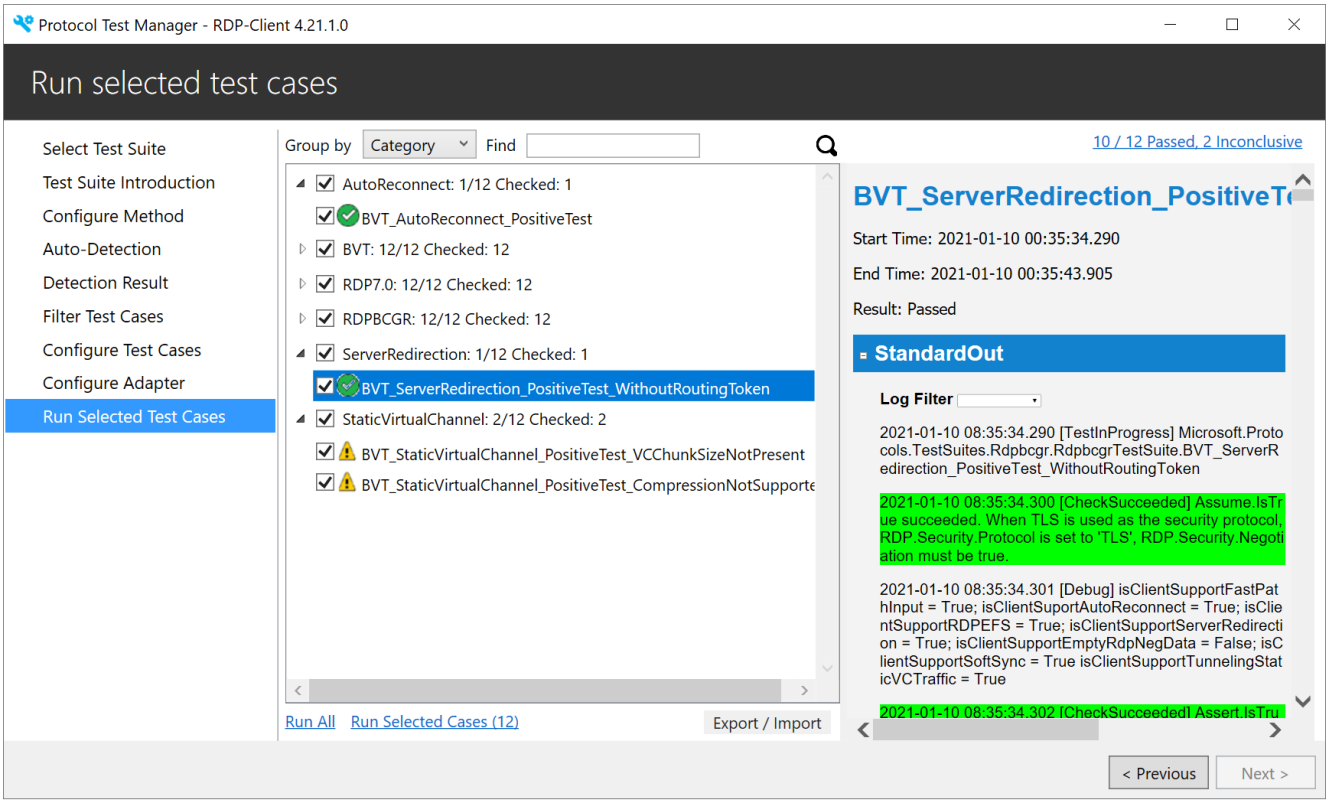


Figure 19. Protocol Test Manager : Test Results 'Category' grouping

Passed, Failed, and Inconclusive Tests : Outcome View

The figure that follows is an example of the **Outcome** grouping of Test Cases that are distributed-among and grouped-by the three primary categories for executed Test Cases.

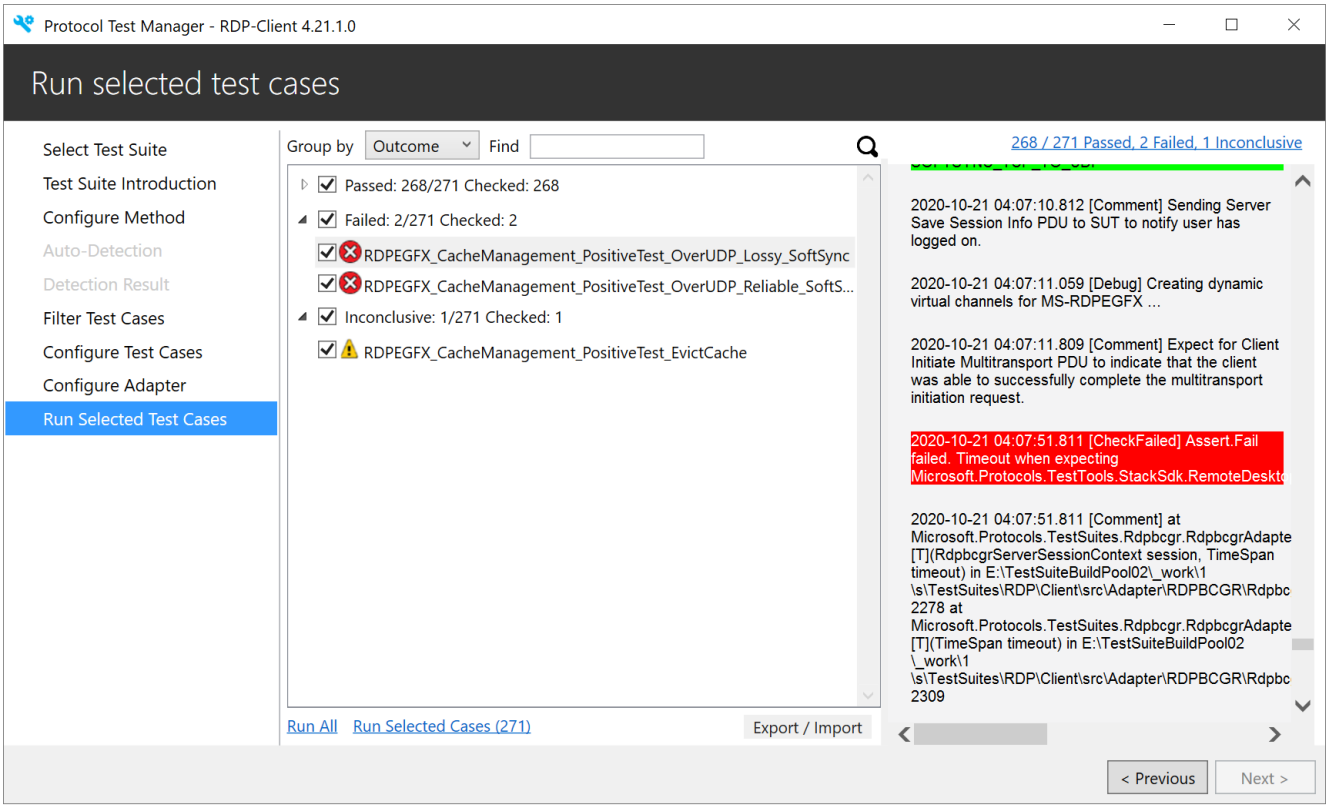


Figure 20. Protocol Test Manager : Passed, Failed, and Inconclusive Test Results 'Outcome' grouping

Viewing Tip

After selecting a completed Test Case in the PTM, you can adjust your view of output log results, such as **StandardOut**, by dragging the separator between the Test Case list view and the log pane to adjust the width of the window.

The sections that follow describe the error and standard output displays that contain line-by-line **ErrorStackTrace**, **ErrorMessage**, and **StandardOut** log data that can assist you in assessing why Test Cases finished execution with **Failed** or **Inconclusive** status.

The discussion begins with descriptions of the low-level **StandardOut** status indicators, also referred to in this document as *information tags*, that track the manner in which Test Cases were executed in terms of the specific test steps taken, the meaning of those steps, and the color codes used to represent the step types.

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 a transport, 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, for example, to 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 Status Indicators

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 Status Indicators

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 Microsoft RDP specification documentation, which may shed light on what the underlying failure may be related to, such as deviations from expected states, values, parameters, event data, and so on.

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

Evaluating a Failed Test Case in PTM

To begin evaluation of a failed Test Case as presented in the PTM, perform the procedure that follows. In this procedure, you will assess and diagnose one of several **Failed** Test Cases that were executed using the profile: `Profile#2_FailScenario_4.21.1.0.ptm`, which you will need to load into the PTM before you begin. This profile should be included with the Tutorial documentation on GitHub.

Important

Failures in this scenario are intentionally created by misconfiguring **Test Case** properties for the purpose of demonstrating the PTM features that enable the analysis of Test Case failures. In practice, Test Case failures are likely to be related to failure in a proprietary RDP implementation on the **SUT computer**, given that the RDP Client Test Suite test cases are designed to pass when run against default Microsoft **RDS** implementations in a Windows environment.

It is for this reason that Test Case failures during proprietary RDP implementation testing should strongly suggest that the failure is sourced in the implementation, not withstanding any unintended Test Case property misconfigurations.

▶ **To analyze a Test Case result with Failure status, perform the steps that follow:**

1. Restart the PTM from the desktop and navigate to the **Configure Method** page of the PTM.
2. Click **Load Profile,** navigate to the directory containing the profiles, and select `Profile#2_FailScenario_4.21.1.0.ptm`
3. Click **OK** to exit the **Open** dialog, at which time the Test Cases for the selected **Profile** are displayed on the **Run Selected Test Cases** page of the PTM.
4. Go to the **Configure Test Cases** page of the PTM and verify that the properties in the following table are set to the values listed in the **New Value** column.

   Note

   The new values should be set by the loaded **Profile** and will result in the generation of a common error for analysis in this procedure. When failure testing is complete, you are advised to reset these values to the original defaults.

   Table 3. Configure Test Cases page : Property values used to invoke an error scenario

Property Name	Default Value	Error Scenario Value
VerifySUTDisplay.Enable	false	true
TurnOffRDPRFXVerification	false	true
VerifySUTDisplay.IQA.AssessValueTh	0.98	1.0
Basic RDP SUT Control Adapter drop-down (on the **Configure Adapters** page)	PowerShell	PowerShell

5. Return to the **Run Selected Test Cases** page of the PTM.
6. Apply a filter to isolate specific Test Cases in the list for this scenario by entering the following text in the **Find** text box that is located above the Test Case list view. Do not use any quotes with this entry:

   "`rfxProgressiveCodec_PositiveTest`"
7. When complete, click the magnifying glass icon to the right of the **Find** box or press the **Enter** key on your keyboard.

   Observe that the Test Case list is filtered down to eleven (12) Test Cases.
8. Sequentially select four Test Cases, starting with the following Test Case: "RPEGFX_RfxProgressiveCodec_PositiveTest_NonProgressiveEncoding_SurfaceNotTileAligned"
9. Start the **RDPSUTControlAgent.exe** on the **SUT computer**.

If you do not have the SUT control agent file, download it and unpack the archive to a chosen SUT folder from here.

Notes

The control agent is required to trigger the **SUT computer** to initiate a connection with the server (**Driver computer**) as the Test Cases start running. For more information, see the RDP SUT Remote Control Protocol documentation.

The control agent also takes a screenshot of the test image on the SUT computer generated from image source data from the Driver computer and sends it back to the Driver computer for comparison, as described in Process Description.

10. Click **Run Selected Cases** below the Test Case list view on the **Run Selected Test Cases** page to start Test Case execution.

    After execution is complete, you should see one test results node showing 4 of 11 Test Cases **Failed** and another node showing 7 of 11 Test Cases **Not Run**. Note that by default the tests are grouped by **Outcome** in the **Group by** drop-down.

11. Click the expansion node next to the **Failed** and **Not Run** check boxes to expose the Test Cases in each of the displayed primary test results categories, as shown in the figure that follows.

12. Click the **Failed** Test Case **RPEGFX_RfxProgressiveCodec_PositiveTest_NonProgressiveEncoding_SurfaceNotTileAligned** and observe that the output log data to the right of the Test Case list view displays the **ErrorStackTrace**, **ErrorMessage**, and **StandardOut** log data categories.

    You will need to contract the first two categories to see the **StandardOut** (third) category.

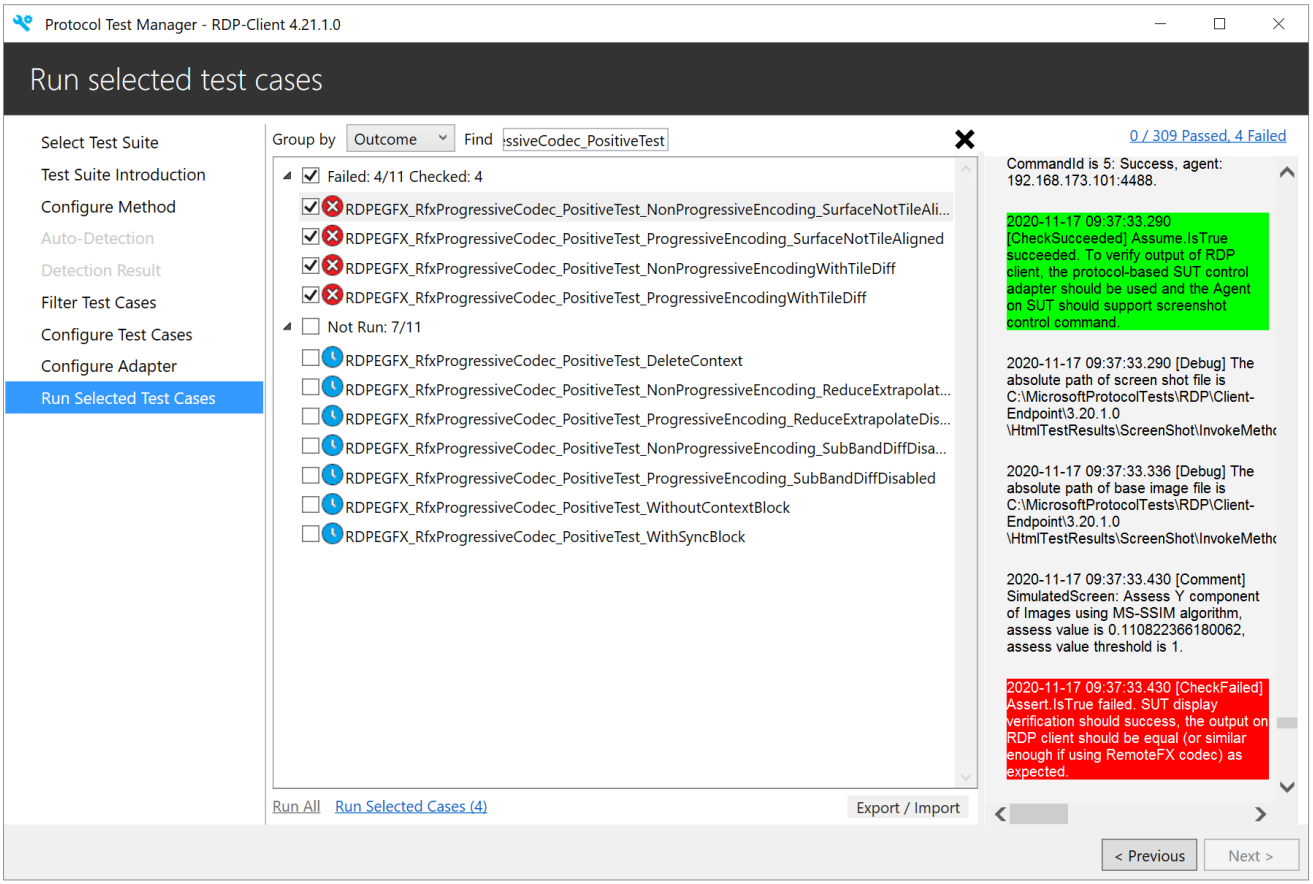


Figure 21. Protocol Test Manager : Test results evaluation of a selected Error case

13. While in the **StandardOut** category, scroll down to the **[CheckFailed]** information tag highlighted in red, as illustrated in the previous figure.

Tip

In many circumstances, the Test Cases that precede a **Failed** test result can expose the conditions that led to such an outcome, which in turn, can point to a potential cause of failure. The table that follows includes four steps that preceded the point of failure for the Test Case of interest in the **StandardOut** log results, as shown in the preceding figure.

In this Test Case analysis, the only information tag that indicates where this particular failure occurred is at step 4, which shows that it resulted from an **AssessValueThreshold** that fell below the threshold value that you configured earlier in Table 3. However, just as it is often the case with error text, the [Comments] tag information cannot necessarily explain the underlying cause of the failure. This is described further in Failure Diagnosis Summary, which offers some possible causes.

Table 4. Failed Test Case results data

Step Info Tag	Time	Log Description | Action	Reference
1. [CheckSucceeded]	2020-11-17 10:36:39.445	Assume.IsTrue succeeded. To verify output of RDP client, the protocol-based SUT control adapter should be used and the Agent on SUT should	The check verifies that the SUT control agent is functioning and should support screenshot control.

		support screenshot control command.	It also indicates the ProtocolBasedRdpSUTControlAdapter type should be used for the SUT Control Adapter methods.
2. [Debug]	2020-11-17 10:36:39.445	Absolute path of screen shot file is C:\MicrosoftProtocolTests\RDP\Client-Endpoint\3.20.1.0\HtmlTestResults\ScreenShot\InvokeMethod_1_ScreenShot.bmp	Indicates the location of the screen shot that was regenerated with an **SUT computer** algorithm from the **Driver computer** image data for test comparison.
3. [Debug]	2020-11-17 10:36:39.492	Absolute path of base image is C:\MicrosoftProtocolTests\RDP\Client-Endpoint\3.20.1.0\HtmlTestResults\ScreenShot\InvokeMethod_1_BaseImage.bmp	Indicates the location of the base source image on the **Driver computer** against which comparison is made to a regenerated SUT screenshot.
4. [Comment]	2020-11-17 10:36:39.586	SimulatedScreen: Assess Y component of images using MS-SSIM algorithm, the assess value is 0.110822366180062; the assess value threshold is 1.	Assessment resulting from algorithmic comparison of the regenerated **SUT** image with the original **Driver** base source image. The calculated assessed value using an **image quality assessment (IQA)** algorithm, falls below the similarity threshold, thus causing failure for image dissimilarities.
5. [CheckFailed]	2020-11-17 10:36:39.586	Assert.IsTrue failed. SUT display verification should success, the output on RDP client should be equal (or similar enough if using RemoteFX codec) as expected.	Point of failure for a process that should return the boolean value 'true' if the assess value threshold was met. The 'true' value did not succeed.

Process Description

Several Test Cases using the RfxCodec were run to compare original bitmap images (sent to the **SUT computer** as data messages) with regenerated bmp images (screenshots) to ascertain whether a high degree of similarity exists between them. The following four Test Cases were run and each resulted in Failure:

-- RDPEGFX_RfxProgressiveCodec_PositiveTest_NonProgessiveEncoding_SurfaceNotTileAligned
-- RDPEGFX_RfxProgressiveCodec_PositiveTest_ProgessiveEncoding_SurfaceNotTileAligned
-- RDPEGFX_RfxProgressiveCodec_PositiveTest_NonProgessiveEncodingWithTileDiff
-- RDPEGFX_RfxProgressiveCodec_PositiveTest_ProgessiveEncodingWithTileDiff

The following process description steps focus on only the first Test Case listed above, but apply similarly to all of them:

1. The RDP Test Case sends a frame of encoded data messages (of the bitmap image) to the SUT computer.
2. An SUT algorithm decodes the messages, regenerates the bmp image, and displays it on the SUT screen.
3. The SUT control agent captures a sample tile of the SUT screen image and sends it back to the **Driver computer**. There, it is compared to the original base image on the **Driver** computer that exists in the following directory location: C:\code\RDPClient-Endpoint\4.21.1.0\Bin\ RdpegfxRfxProgressiveCodecTestImage.bmp
4. If the comparison results in a similarity level that is less than the set **VerifySUTDisplay.IQA.AssessValueThreshold** value, which for this error evaluation scenario is equal to a value of 1, meaning 100%, the Test Case will fail.

Failure Diagnosis Summary

Possible causes of failure may include the following:

- The IQA algorithm threshold value is incorrect for the similarity level of images generated on the SUT. The threshold may need to be lowered.
- Implementation feature coding is faulty.
- The SUT decode algorithm is erroneous, resulting in a high degree of dissimilarity between the regenerated screen shot on the SUT and the original image on the Driver.
- The SUT control agent is improperly implemented.

Background

In the RDP Client Test Suite, the Threshold is set to the default value of 98% (0.98) for Windows to establish the minimum value quality measure of image similarity that must be met by the implementation under test to avoid generating an error, that is, for Windows implementations, as demonstrated in this example.

However, the minimum Threshold value is configurable and can be changed to accommodate varying encoding/compression algorithm processes, network speed, and data transmission loss in a particular implementation. The bottom line is that whatever the threshold value is set to, the assessed value must meet or exceed it or an error will occur.

If the threshold requirement is not met, it may signal that the decode algorithm on the SUT is invalid. As an aid to troubleshooting an implementation that is experiencing errors in this area, the example given here can provide a window into the underlying processes that are occurring. For example, if a sufficient image quality measurement cannot be obtained, it may be that the SUT implementation decoding algorithm needs some functional adjustments.

More Information

To learn more about the value ranges and other details for RDP protocol Test Suite property settings, see *Configuring the Test Suite* in the RDP Client Test Suite User Guide.

To learn more about Test Cases, including test scenarios and definitions, see the RDP_Overview_Client Test Design Specification.

To learn more about the RfxCodec, see [MS-RDPRFX]: Remote Desktop Protocol: RemoteFX Codec Extension document.

Using the Log Filter to Manipulate the Data View

In the PTM **StandardOut** log results, you can focus on only the data associated with specific status indicators (information tags) by making use of the **Log Filter** drop-down. For example, you might want to look at the data for the **[Debug]** information tags only, so you can isolate such information when it is important to your analysis. To use this feature, perform the steps that follow.

▶ **To filter StandardOut log results using the Log Filter drop-down:**

1. In the **Log Filter** drop-down just below the **StandardOut** log title, click the drop-down arrow to display the list of information tags that you can select to filter the log display, as shown in the figure that follows.

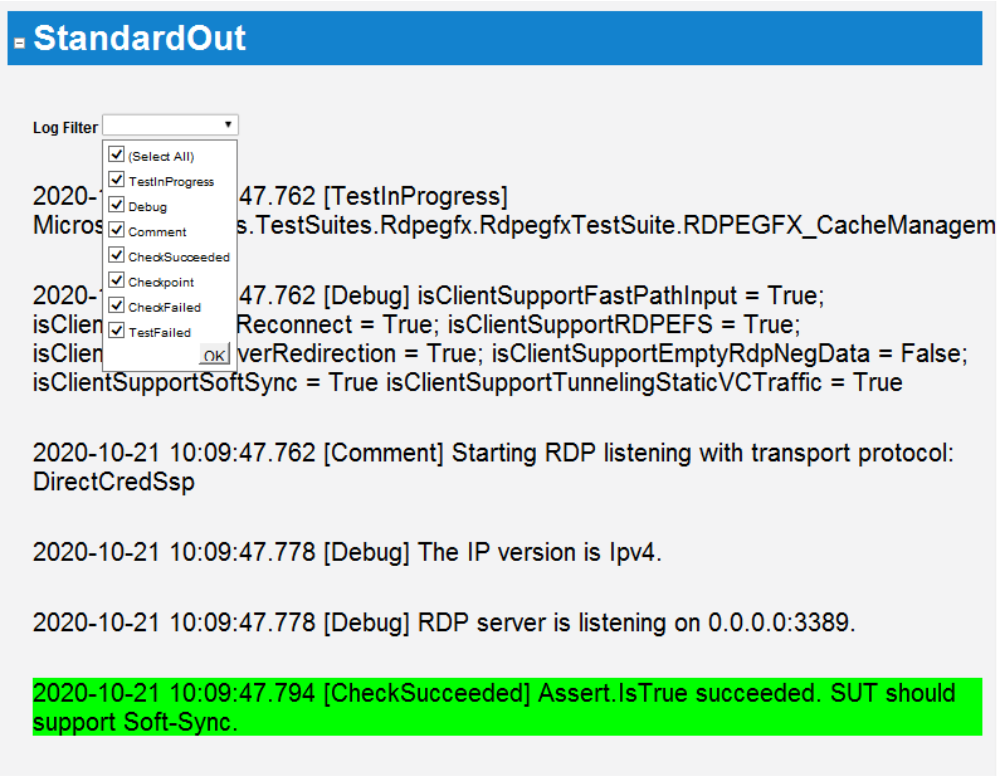


Figure 22. Protocol Test Manager : Applying Log Filtering to StandardOut test results data

2. Select one or more **Log Filter** check boxes, then click **OK** and observe that the **StandardOut** results show data for only the information tags that you selected.

Tip

Observe that as you scroll through **StandardOut** results, [Comments] information tags can provide some related background RDP documentation references that can be checked for

further clarification of issues. These can appear primarily for Test Cases that have an execution results status of **Inconclusive** or **Failed** and can help identify the causes of failures.

Using the Find Feature to Locate Test Cases

If you need to filter your list of Test Cases to isolate a particular Test Case or group of Test Cases containing similar names, you can launch a search by entering a text string that is part of a Test Case name in the **Find** text box above the Test Cases list view and pressing **Enter** on your keyboard. You can also click the search icon to the right of the **Find** box to obtain the same result.

For example, if you wanted to isolate all scenario S1 tests (SUT Verification tests), you could do that by entering the text "S1_" in the **Find** text box and clicking **Enter** on your keyboard. If you want to remove the search text, then click the **X** icon to the right of the **Find** text box.

Using HTML Format to Manipulate the StandardOut View

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 21. Protocol Test Manager : Test results evaluation of a selected Error case.

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.

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

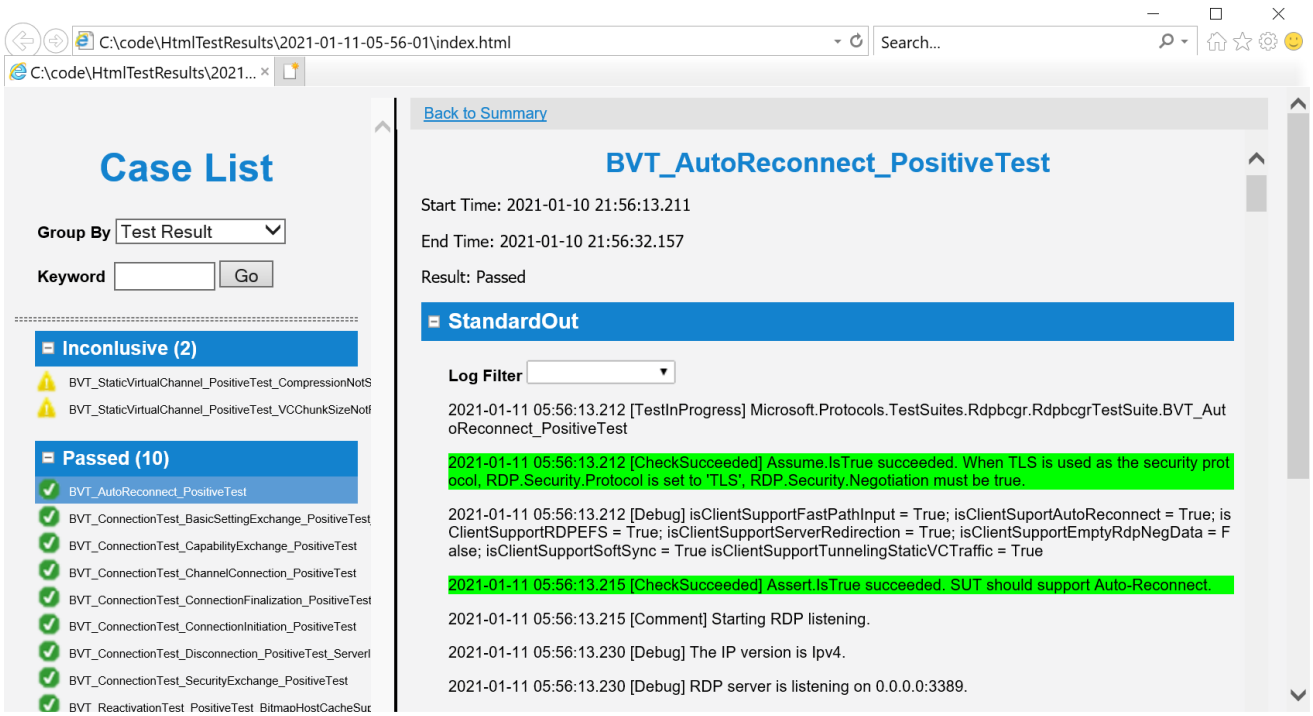


Figure 23. Protocol Test Manager : StandardOut results in separate HTML format

In the previous figure, the Test Case **BVT_AutoReconnect_PositiveTest** is selected and identified with a green icon, which indicates the primary result status of **Passed.** Note that you can select

either of the **Inconclusive** Test Cases and the log results in the right-hand pane of the HTML display will change to show the **ErrorStackTrace** and **ErrorMessage** log categories , where the latter provides further information about why the Test Cases returned as **Inconclusive**.

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 in the Lab session Tutorial.

Table 5. 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.	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.
	Errors appear in the Selected Test Cases pane in the **Group by Outcome** configuration.	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		

More Information

To learn more about potential Test Case failures, see the *Troubleshooting* section of the RDP Client Test Suite User Guide on GitHub.

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. Consult this information only if you are prepared to engage with more advanced and complex technologies:

- RDP Client Test Suite User Guide
- MS-RDPBCGR : Remote Desktop Protocol : Basic Connectivity and Graphics Remoting
- MS-RDSOD : Remote Desktop Services Protocols Overview
- [MS-RDPRFX]: Remote Desktop Protocol: RemoteFX Codec Extension
- RDP_Overview_Client Test Design Specification
- Getting Started Guide for PTF
- RDP SUT Remote Control Protocol