

# Calnex PTP Field Verifier (PFV) Remote Control Guide



**For Calnex Paragon Instruments**

**Version 8.0**

# Contents

Contents .....	2
Introduction.....	3
Generating Remote Control Scripts from the Script Recorder .....	4
Paragon-X .....	4
Using Remote Control from Tcl .....	5
Paragon-X: Location of the Tcl Module.....	5
Paragon-100G and Paragon-Neo: Location of the Tcl Module.....	5
Using the Tcl Module .....	5
Running Commands or Scripts .....	5
Using Remote Control from Python.....	6
Paragon-X: Location of the Python Module .....	6
Paragon-100G and Paragon-Neo: Location of the Python Module .....	6
Using the Python Module.....	6
Running Commands or Scripts .....	6
Using Remote Control from Perl.....	7
Connecting to the Instrument .....	7
Filename specification within scripts/commands.....	7
Error reporting.....	7
Firewall Settings .....	7
Extending Paragon-X Scripts to Control Paragon-100G and Paragon-neo .....	7
Paragon-100G and Paragon-neo Script Recorder .....	8
Using the RESTful API.....	9
Command Reference Concepts.....	10
PFV Concepts.....	10
Location of Rules Files .....	10
PTP Field Names .....	11
ToD Field Names.....	12
TLV Field Names .....	13
UDP Field Names .....	16
Settings.....	17
Command Reference Format.....	17
Special Commands .....	17
Command Reference – Contents .....	18
Control Commands .....	20
PFV Commands .....	21
Command List Index .....	69

## Introduction

Calnex's family of Paragon instruments allow you to automate the testing of your devices. To support this, remote control functionality is built-in to the Paragon instruments as a standard feature. This document details the commands used to control the Calnex PTP Field Verifier (PFV) in conjunction with the Paragon instruments. The commands detailed here can be used with the Paragon-X, Paragon-t, Paragon-100G and Paragon-neo.



Note: The functionality available in the PFV is determined by the functionality of the instrument used; not all commands are available with all products.

---

There are additional remote control commands that can be used specifically with Paragon-100G and Paragon-neo (using the REST API) – these are not documented here, however, online documentation is available in the instrument.

Details regarding the setup required to use Paragon remote control can be found in the Paragon Remote Control Manual. Minimum PC requirements (including supported OS's) are listed in Application Note CX5006: Paragon Remote Client Application Software - Minimum PC Requirements and PC Management Recommendations document.

### Overview

Paragon instruments (and associated remote control) consist of 3 main components:

- Hardware configuration and capture control
- Metrics analysis and visualization (using the CAT)
- Message field verification (using the PFV)

The PFV allows for the in-depth analysis of captured data, both on previously captured data (which does not require access to hardware) and in real-time whilst running a live capture on an instrument.

This document describes how to use remote control functionality for the PFV. Remote control for hardware configuration and capture control and for the CAT is detailed in separate documents. These documents can be accessed from the Paragon-X GUI (**Help/Remote Control**) and from the Windows **Start** menu under **Calnex/Documentation**.

Tcl and Python are supported and the commands for each are provided.

## Generating Remote Control Scripts from the Script Recorder

The Paragon-X, Paragon-100G and Paragon-Neo can record user operations and convert these into scripted commands. This makes script creation very simple – record keypresses in the GUI and then use the recorded script as part of your test program. The script recorder logs commands for the CAT and PFV as well as for instrument control.



Note: The commands recorded by the Paragon-X can be used as the basis for scripts to control Paragon-100G and Paragon-Neo but only a subset of these commands and associated parameters are supported since there are differences in the available functionality.

---

### Paragon-X

To use the script recorder:

1. Start the Paragon-X GUI and select the **Script Recorder...** option from the **Tools** drop-down menu. A new dialog will appear as follows:



2. Select the scripting language that you are using and then click **Start**. A splash screen and recording window will then be displayed.
3. Configure the Paragon and perform the operations you would like scripted.
4. When you have completed your chosen operations, select **View Recorded Script** from the **Tools** drop-down menu.
5. You can now save the script and run it when required. Alternatively, cut and paste the script snippet into your test program.
6. From the **Tools** drop-down you can now de-select the **Script Recorder...** option to stop recording.

## Using Remote Control from Tcl

A Tcl module is provided for remote control functionality. This module has been verified using ActiveState Tcl, version 8.5. It is recommended that this is the version you use.

The Tcl module provides a simple bridge between Tcl and the network interface protocol used to talk to the Paragon application.

### Paragon-X: Location of the Tcl Module

The Paragon Tcl module (`paragon.tcl`) is located under the **My Documents** folder. In Windows 10 for instance, the location is typically:

```
C:\Users\<<UserName>\Documents\Calnex\Paragon-X\RemoteControl\Tcl
```

For backward compatibility, the Tcl module is also located in the same directory as the Paragon-X software; this is normally one of:

```
C:\Program Files\Calnex\Paragon-X\
```

or

```
C:\Program Files (x86)\Calnex\Paragon-X\
```

### Paragon-100G and Paragon-Neo: Location of the Tcl Module

To use the commands listed in this document with Paragon-100G or Paragon-neo, you should use the `paragon.tcl` file distributed with Paragon-X (see above).

### Using the Tcl Module

The Tcl library must be referenced using the Tcl `source` command to read this file prior to running Tcl commands or scripting e.g.

```
source c:/Program Files/Calnex/Paragon-X/paragon.tcl
```

### Running Commands or Scripts

To run a Tcl command from the command line:

1. Start a Tcl shell e.g. `tclsh`
2. In the shell window, type: `source <path>/paragon.tcl`  
Where `<path>` is the path to the `paragon.tcl` file as detailed above.
3. Type your Tcl command e.g. `connect...` Note that you must connect to Paragon before executing other commands (see Connecting to the Instrument).

To run a Tcl script:

1. Start a Tcl shell e.g. `tclsh`
2. In the shell window, type: `source <path>/paragon.tcl`  
Where `<path>` is the path to the `paragon.tcl` file as detailed above.  
Alternatively, include the `source <path>/paragon.tcl` command in your script.
3. In the shell window, type `source` followed by the name of your script.

## Using Remote Control from Python

A Python module is provided for remote control functionality. This module has been verified using Python 3.4. It is recommended that the Python interpreter installed is the same version or newer otherwise Python functionality may not work correctly. The Python module provides a simple bridge between Python and the network interface protocol used to talk to the Paragon application.

### Paragon-X: Location of the Python Module

The Paragon Python module (`paragon.py`) is located under the **My Documents** folder. In Windows 7 for instance, the location is typically:

```
C:\Users\<<UserName>\Documents\Calnex\Paragon-X\RemoteControl\Python
```

In this folder, you will also find two example test scripts (`test_simple.py` and `test_replay.py`) that use this module.

### Paragon-100G and Paragon-Neo: Location of the Python Module

To use the commands listed in this document with Paragon-100G or Paragon-neo, you should use the `paragon.py` file distributed with Paragon-X (see above).

### Using the Python Module

The Python module must be imported before it can be used i.e.

```
import sys
sys.path.append(r'C:\Users\<<UserName>\Documents\Calnex\Paragon-
X\RemoteControl\Python')
import paragon as p
```

The path may need to be adjusted depending on the location of the module.

### Running Commands or Scripts

To run a Python command from the command line:

1. Start a Python shell: `Python`
2. In the shell window, type: `import sys`
3. Type the `sys.path.append` statement as detailed above.
4. Type `import paragon as p`
5. Type your python command e.g. `p.connect (...)`. Note that you must connect to Paragon before executing other commands (see Connecting to the Instrument).

To run a Python script:

1. Start a DOS command window
2. If you have not previously configured your path, then type the `sys.path.append` statement as described above.
3. Type `import paragon as p`. Alternatively, you can include the `import` statement in your script.
4. At the prompt navigate to the directory where your script is stored e.g.  
`C:\Users\<<UserName>\Documents\Calnex\RemoteControl\Python`
5. Type the name of the script including the `.py` filename extension e.g.  
`Test_simple.py`

## Using Remote Control from Perl

Perl is no longer supported.

## Connecting to the Instrument

To control a Paragon remotely, you first have to connect to it. For Paragon-X and Paragon-t, the Paragon GUI must be running on the controlling PC before any connection can be established. See the Paragon-X or Paragon-t Remote Control Manual for details.

For Paragon-X and Paragon-t, remote control for the CAT is also managed through the Paragon GUI. The GUI must be running on the controlling PC before any connection to the CAT can be established. For Paragon-100G and Paragon-neo, the instrument simply needs to be powered on.

## Filename specification within scripts/commands

Commands which take filenames must use a directory separator. The Windows directory separator '\ ' must be written as '/' and spaces must be written as '\ ' e.g .

```
recall "c:/Test\ Documents/sync-ethernet.cst".
```

## Error reporting

Errors are signalled using the usual 'error' call for the remote control interface being used i.e. Tcl or Python.

Often a command will have a dependency on an accompanying configuration or parameter; this will be reported in the error message.

## Firewall Settings

Remote control for Paragon-X operates using a TCP socket connection to a specified port (the default is port 9000). Any firewall must therefore be configured to allow connections on the specified TCP port.

## Extending Paragon-X Scripts to Control Paragon-100G and Paragon-neo

Some of the functionality in Paragon-100G and Paragon-neo may not be available using Paragon-X style commands. In this case, the RESTful API in Paragon-100G and Paragon-neo can be used in conjunction with the commands in this document.

For more details on the RESTful API go to the **Help** page on your Paragon-100G or Paragon-neo instrument.



Note: The use of the RESTful API may require you to install one or more additional packages for the script language you are using:

**ActiveTcl:** Requires the "REST" package. This can be installed using:

```
teacup update
```

**Python:** Requires the "requests" package. This can be installed using:

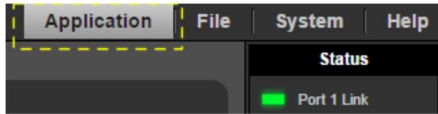
```
<Python install directory>/Scripts/pip install requests
```

---

## Paragon-100G and Paragon-neo Script Recorder

Paragon 100G and Paragon-neo web applications can record user operations and convert these into scripted (RESTful) commands. This makes script creation very simple – record actions in the browser and then use the recorded script as part of your test program.

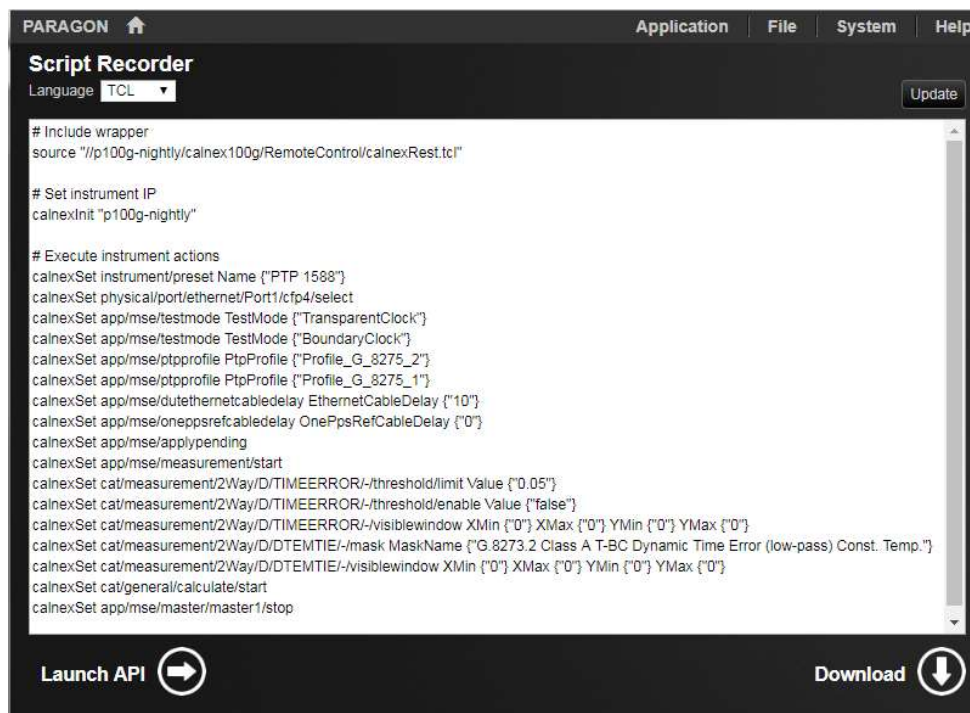
To use script recorder, open a web browser and enter the URL for your instrument. Then select **Application** from the menu bar:



To start recording operations, select **Start** from the **Script Recorder** panel in the bottom left-hand corner of the browser window:



Configure the Paragon and perform the operations you want scripted. When you have completed your chosen operations, click **Script**. You will now see a new page with your recorded script:



The default script language is Tcl. However, you can change this to Python from the **Language** pull-down. This is possible at any time even after the script has been recorded.

Clicking **Update** in the top-right hand corner of the recorded script window will update the script with any actions that have been recorded since the script window was last refreshed.

Click **Stop** on the main instrument page under **Script Recorder** to stop recording. The recorded script can be copied from the script window or downloaded to your local PC.



## Using the RESTful API

To use the Paragon-100G or Paragon-neo RESTful API in conjunction with the commands in this document, the appropriate `calnexRest` wrapper for your script language must be referenced by your script. Two wrappers are provided: `calnexRest.tcl` (Tcl) and `calnexRest.py` (Python).

The wrappers are located in `//<instrumentIpAddress>/calnex100G/RemoteControl/`. The wrapper must be referenced before it can be used. This is done in the same way as for the Paragon-X wrappers as described in *Using the Tcl Module and Using the Python Module* above. For example:

### Tcl:

```
source
"//<instrumentIpAddress>/calnex100G/RemoteControl/calnexRest.tcl"
```

### Python:

```
import sys
sys.path.append(r'//<instrumentIpAddress>/calnex100G/RemoteControl')
from calnexRest import calnexInit, calnexGet, calnexSet,
calnexCreate, calnexDel, calnexGetVal
```

The main functions provided by the wrappers are shown below. Any equivalent Paragon-X command is shown in brackets:

- `calnexInit`: must be called before any other function. The parameter is the IP address of the instrument.
- `calnexSet` (`paragonset`)
- `calnexGetVal` (`paragonget`): returns the value of a single specified setting.
- `calnexGet`: can return a single value or a set of values.

For more details on these commands, please consult the documentation on the **Help** page of your instrument.



Note: The wrapper functions and documentation for Paragon-100G and Paragon-neo have been revised. The information in this document is relevant only for Paragon-100G versions later than 06.03 and Paragon-neo versions later than 00.05.

---

## Command Reference Concepts

The following concepts are common themes throughout this Remote Control Guide. It is important to be aware of the level of control each command has, for example, modifying one parameter may affect multiple measurement parameters.

### PFV Concepts

The PFV can load:

- Timing capture files from Paragon-X, Paragon-100G and Paragon-neo.
- CCSA and G.8271 ToD capture files from Paragon-X.
- pcap files containing PTP messaging.

The contents of each of the above files types can be analyzed against a set of rules. A number of pre-defined rules are supplied with the PFV; alternatively, you can define your own. Verification against rules can only be performed in the following circumstances:

- The Paragon instrument used to capture the data has the appropriate option.
- The PC being used for analysis is licensed.

If no option/licence is detected, the the PFV will load the file and display its contents but no rules checking will be possible.

The commands in this section describe the commands used to control the PFV settings and behaviour.

### Location of Rules Files

Calnex provides a set of pre-defined rules files. For the PC-based PFV, these are installed by default in: C:\Program Files (x86)\Calnex\PFV\Rules\

In addition, users may create their own rules files. For the PC-based PFV, user-defined rules files should be stored in: C:\Users\<User>\Documents\Calnex\PFV\Rules\

For P100G, user-defined rules files should be stored in: \\<P100G>\Calnex100G\PFV\Rules

The application will populate the rules pull-down from all xml files in these folders. When selecting a rules file using remote control (i.e. using **PFV Rules PTP <filename>** or **PFV Rules ToD <filename>**), only the name of the file (as shown in the UI) is required – both the folders mentioned above will be searched for the matching file name.

## PTP Field Names

A number of the PFV remote commands take a PTP field name as a parameter. The field names used by remote control are the same as in the rules files and the PFV UI. The PTP field names are:

Message Type	All Except Announce/Management	Announce Only	Management Only
Field Names	alternateTimeTransmitterFlag	curUtcOffset	MgtTargetPortIdentity
	decodedcorrectionfieldps	gmClkAcc	MgtStartingBoundaryHops
	correctionField	gmClkClass	MgtBoundaryHops
	currentUtcOffsetValid	gmClockIdentity	MgtReserved0
	domainNumber	gmClkOslv	MgtActionField
	frequencyTraceable	gmPrior1	MgtReserved1
	leap59	gmPrior2	
	leap61	gmStepsRemoved	
	logMessageInterval	gmTimeSource	
	ptpTimescale		
	sourcePortIdentity OR sourceClockID		
	sourcePortNo		
	timeTraceable		
	twoStepFlag		
	unicastFlag		

## ToD Field Names

The ToD field names used by remote control are the same as in the rules files and the PFV UI. The G.8271 ToD field names are:

Message Type	Time Event	Time Announce	GNSS Status
Field Names	Length	Length	Length
	UTCTime	versionPTP	TimeSource
	Reserved0	domainNumber	Status
	Flags	FlagField	Alarms
	CurrentUTCOffset	sourceClockIdentity	Reserved0
	Reserved1	sourcePortNumber	FCS
	FCS	grandmasterPriority1	
		grandmasterPriority2	
		clockClass	
		clockAccuracy	
		offsetScaledLogVariance	
		grandmasterClockIdentity	
		stepsRemoved	
		timeSource	
	Reserved0		
	FCS		

The CCSA ToD field names are:

Message Type	Time Event	Time Information
Field Names	Length	Length
	UTCTime	TimeSourceType
	Reserved0	TypeSourceStatus
	Week	Alarms
	LeapS	Reserved0
	PPSstatus	Reserved1
	TAcc	Reserved2
	Reserved1	Reserved3
	Reserved2	Reserved4
	Reserved3	FCS
	FCS	

## TLV Field Names

TLV field names used by remote control are defined in the table below.

TLV	TLV Name	TLV Fields
Unicast Sync Request Signaling TLV	TlvUnicastReqSync	TlvUnicastReqSync_Type TlvUnicastReqSync_Length TlvUnicastReqSync_MsgType TlvUnicastReqSync_Reserved0 TlvUnicastReqSync_LogInterMsgPeriod TlvUnicastReqSync_DurationField
Unicast Sync Grant Signaling TLV	TlvUnicastGrantSync	TlvUnicastGrantSync_Type TlvUnicastGrantSync_Length TlvUnicastGrantSync_MsgType TlvUnicastGrantSync_Reserved0 TlvUnicastGrantSync_LogInterMsgPeriod TlvUnicastGrantSync_DurationField TlvUnicastGrantSync_Reserved1 TlvUnicastGrantSync_Reserved2 TlvUnicastGrantSync_RenewalInvited
Unicast Announce Request Signaling TLV	TlvUnicastReqAnnounce	TlvUnicastReqAnnounce_Type TlvUnicastReqAnnounce_Length TlvUnicastReqAnnounce_MsgType TlvUnicastReqAnnounce_Reserved0 TlvUnicastReqAnnounce_LogInterMsgPeriod TlvUnicastReqAnnounce_DurationField
Unicast Announce Grant Signaling TLV	TlvUnicastGrantAnnounce	TlvUnicastGrantAnnounce_Type TlvUnicastGrantAnnounce_Length TlvUnicastGrantAnnounce_MsgType TlvUnicastGrantAnnounce_Reserved0 TlvUnicastGrantAnnounce_LogInterMsgPeriod TlvUnicastGrantAnnounce_DurationField TlvUnicastGrantAnnounce_Reserved1 TlvUnicastGrantAnnounce_Reserved2 TlvUnicastGrantAnnounce_RenewalInvited
Unicast Del Resp Request Signaling TLV	TlvUnicastReqDelResp	TlvUnicastReqDelResp_Type TlvUnicastReqDelResp_Length TlvUnicastReqDelResp_MsgType TlvUnicastReqDelResp_Reserved0 TlvUnicastReqDelResp_LogInterMsgPeriod TlvUnicastReqDelResp_DurationField
Unicast Del Resp Grant Signaling TLV	TlvUnicastGrantDelResp	TlvUnicastGrantDelResp_Type TlvUnicastGrantDelResp_Length TlvUnicastGrantDelResp_MsgType TlvUnicastGrantDelResp_Reserved0 TlvUnicastGrantDelResp_LogInterMsgPeriod TlvUnicastGrantDelResp_DurationField TlvUnicastGrantDelResp_Reserved1 TlvUnicastGrantDelResp_Reserved2 TlvUnicastGrantDelResp_RenewalInvited

Unicast PDel Resp Request Signaling TLV	TlvUnicastReqPDelResp	TlvUnicastReqPDelResp_Type TlvUnicastReqPDelResp_Length TlvUnicastReqPDelResp_MsgType TlvUnicastReqPDelResp_Reserved0 TlvUnicastReqPDelResp_LogInterMsgPeriod TlvUnicastReqPDelResp_DurationField
Unicast PDel Resp Grant Signaling TLV	TlvUnicastGrantPDelResp	TlvUnicastGrantPDelResp_Type TlvUnicastGrantPDelResp_Length TlvUnicastGrantPDelResp_MsgType TlvUnicastGrantPDelResp_Reserved0 TlvUnicastGrantPDelResp_LogInterMsgPeriod TlvUnicastGrantPDelResp_DurationField TlvUnicastGrantPDelResp_Reserved1 TlvUnicastGrantPDelResp_Reserved2 TlvUnicastGrantPDelResp_RenewalInvited
CCSA / CMCC 5G High Precision TLV	TlvCmcc5GHighPrecision	TlvType TlvLength TlvOrganizationId TlvOrganizationSubType TlvCmcc5GAccuracyLevel1 TlvCmcc5GStepsRemovedLevel1 TlvCmcc5GAccuracyLevel2 TlvCmcc5GStepsRemovedLevel2 TlvCmcc5GAccuracyLevel3 TlvCmcc5GStepsRemovedLevel3
IEEE C37.238-2011 TLV	TlvC37238_2011	TlvC37238_2011_Type TlvC37238_2011_Length TlvC37238_2011_OrgId TlvC37238_2011_OrgSubType TlvC37238_2011_GrandmasterId TlvC37238_2011_GrandmasterTimelnaccuracy TlvC37238_2011_NetworkTimelnaccuracy TlvC37238_2011_Reserved
IEEE C37.238-2017 TLV	TlvC37238_2017	TlvC37238_2017_Type TlvC37238_2017_Length TlvC37238_2017_OrgId TlvC37238_2017_OrgSubType TlvC37238_2017_GrandmasterId TlvC37238_2017_Reserved1 TlvC37238_2017_TotalTimelnaccuracy TlvC37238_2017_Reserved2
IEEE 1588-2008 Alternate Time Offset Indicator TLV	TlvAlternateTimeOffsetIndicator	TlvATOI_Type TlvATOI_Length TlvATOI_KeyField TlvATOI_CurrentOffset TlvATOI_JumpSeconds TlvATOI_TimeOfNextJump TlvATOI_DisplayName TlvATOI_Pad
IEEE 802.1AS Followup Information TLV	Tlv8021ASFollowUpInfo	Tlv8021ASFollowUp_Type Tlv8021ASFollowUp_Length Tlv8021ASFollowUp_OrgId

		Tlv8021ASFollowUp_OrgSubType Tlv8021ASFollowUp_CSRO Tlv8021ASFollowUp_GmTimeBaseInd Tlv8021ASFollowUp_LastGmPhaseChange Tlv8021ASFollowUp_ScaledLastGmFreqChange
IEEE 802.1AS Message Interval Request TLV	Tlv8021ASMsgIntvlReq	Tlv8021ASMsgIntvlReq_Type Tlv8021ASMsgIntvlReq_Length Tlv8021ASMsgIntvlReq_OrgId Tlv8021ASMsgIntvlReq_OrgSubType Tlv8021ASMsgIntvlReq_LogLinkDelayIntvl Tlv8021ASMsgIntvlReq_LogTimeSyncIntvl Tlv8021ASMsgIntvlReq_LogAnnounceIntvl Tlv8021ASMsgIntvlReq_Flags Tlv8021ASMsgIntvlReq_Reserved0
IEEE 802.1AS gPTP Capable TLV	Tlv8021ASGtpCapable	Tlv8021ASGtpCapable_Type Tlv8021ASGtpCapable_Length Tlv8021ASGtpCapable_OrgId Tlv8021ASGtpCapable_OrgSubType Tlv8021ASGtpCapable_LogGtpCapMsgIntvl Tlv8021ASGtpCapable_Flags Tlv8021ASGtpCapable_Reserved0
IEEE 802.1AS gPTP Capable Message Interval Request TLV	Tlv8021ASGtpCapableMsg IntvlReq	Tlv8021ASGtpCapMsgIntvlReq_Type Tlv8021ASGtpCapMsgIntvlReq_Length Tlv8021ASGtpCapMsgIntvlReq_OrgId Tlv8021ASGtpCapMsgIntvlReq_OrgSubType Tlv8021ASGtpCapMsgIntvlReq_LogGtpCapMsgIntvl Tlv8021ASGtpCapMsgIntvlReq_Reserved0
IEEE 1588-2008 Path Trace TLV	TlvPathTrace	TlvPathTrace_Type TlvPathTrace_Length TlvPathTrace_PathSequence
IEEE 1588-2019 Port Communication Availability TLV	TlvPortCommAvail	TlvPortCommAvailMask TlvPortCommAvail_Type TlvPortCommAvail_Length TlvPortCommAvail_Sync TlvPortCommAvail_DelayResp TlvPortCommAvail_Reserved1 TlvPortCommAvail_Reserved2
IEEE 1588-2019 Protocol Address TLV	TlvProtocolAddress	TlvProtocolAddressMask TlvProtocolAddress_Type TlvProtocolAddress_Length TlvProtocolAddress_Address
SMPTE Synchronization Metadata 2015	TlvSmpteSm_2015	TlvSmpteSm_2015_Type TlvSmpteSm_2015_Length TlvSmpteSm_2015_OrgId TlvSmpteSm_2015_OrgSubType TlvSmpteSm_2015_DefaultSystemFrameRate TlvSmpteSm_2015_DefaultSystemFrameRateNumerator TlvSmpteSm_2015_DefaultSystemFrameRateDenominator TlvSmpteSm_2015_TimeTransmitterLockingStatus TlvSmpteSm_2015_TimeAddressFlags TlvSmpteSm_2015_Taf_DropFrameFlag TlvSmpteSm_2015_Taf_ColorFrameIdFlag TlvSmpteSm_2015_Taf_ReservedBit2

		TlvSmpteSm_2015_Taf_ReservedBit3 TlvSmpteSm_2015_Taf_ReservedBit4 TlvSmpteSm_2015_Taf_ReservedBit5 TlvSmpteSm_2015_Taf_ReservedBit6 TlvSmpteSm_2015_Taf_ReservedBit7 TlvSmpteSm_2015_CurrentLocalOffset TlvSmpteSm_2015_JumpSeconds TlvSmpteSm_2015_TimeOfNextJump TlvSmpteSm_2015_TimeOfNextJam TlvSmpteSm_2015_TimeOfPreviousJam TlvSmpteSm_2015_PreviousJamLocalOffset TlvSmpteSm_2015_DaylightSaving TlvSmpteSm_2015_Dsf_Current TlvSmpteSm_2015_Dsf_Next TlvSmpteSm_2015_Dsf_Previous TlvSmpteSm_2015_Dsf_ReservedBit3 TlvSmpteSm_2015_Dsf_ReservedBit4 TlvSmpteSm_2015_Dsf_ReservedBit5 TlvSmpteSm_2015_Dsf_ReservedBit6 TlvSmpteSm_2015_Dsf_ReservedBit7 TlvSmpteSm_2015_LeapSecondJump TlvSmpteSm_2015_Lsj_ReasonForChange TlvSmpteSm_2015_Lsj_ReservedBit1 TlvSmpteSm_2015_Lsj_ReservedBit2 TlvSmpteSm_2015_Lsj_ReservedBit3 TlvSmpteSm_2015_Lsj_ReservedBit4 TlvSmpteSm_2015_Lsj_ReservedBit5 TlvSmpteSm_2015_Lsj_ReservedBit6 TlvSmpteSm_2015_Lsj_ReservedBit7
--	--	---

### UDP Field Names

The UDP Header fields are:

- UdpChecksum



## Settings

Individual settings may be set or queried using remote control commands. Most settings can be set and queried individually, but some may only be queried. The general syntax for commands is as follows:

To **set** a setting:

```
paragonset <parameter name> <parameter value> [<parameter value>]
```

To **query** a setting:

```
paragonget <parameter name>
```

Also, note that:

- A query will return one or more *<parameter value>*s.
- The parameters taken by these commands are described in more detail later in this document. *<parameter name>* is made up of a space separated list.
- The *<parameter value>* type depends on the command. It may be a number, a Boolean, a string or an enumerated value. String parameter values containing spaces must be quoted.
- Certain commands take an index as their parameter. Index parameters are shown as *<index>*.
- The index value should be suffixed with a '#' character. If the index is a string parameter containing spaces, it must be quoted, including the '#' suffix.

## Command Reference Format

In the command reference below, most commands are detailed independently of the scripting language used. Each scripting language has a different calling mechanism but the CAT commands are the same. In addition, the majority of commands can be set (`paragonset`) and read (`paragonget`). Where appropriate, the set and get functions are detailed separately in the command reference.

Examples of set and get functions in each supported scripting language are shown below:

Command	PFV PTP Rules <filename>
<b>Tcl</b>	<pre>paragonset PFV PTP Rules "G.8275.1_Phase_Profile.xml" paragonget PFV PTP Rules</pre>
<b>Python</b>	<pre>paragonset("PFV PTP Rules", "G.8275.1_Phase_Profile.xml"); paragonget("PFV PTP Rules");</pre>
Command	PFV FieldErrorCount <sequenceId>
<b>Tcl</b>	<pre>paragonget PFV PTP FieldErrorCount sequenceId</pre>
<b>Python</b>	<pre>paragonget("PFV PTP FieldErrorCount sequenceId");</pre>

## Special Commands

Some commands do not use the `paragonset` and `paragonget` methods. These commands are listed in Control Commands below.

## Command Reference – Contents

connect.....	20
disconnect.....	20
PFV OpenFile.....	21
PFV Rules (Deprecated).....	21
PFV PTP Rules.....	21
PFV ToD Rules.....	22
PFV Show.....	22
PFV GenerateReport <filename>.....	23
PFV PassResult (Deprecated).....	23
PFV PTP PassResult.....	23
PFV ToD PassResult.....	24
PFV PassRate (Deprecated).....	24
PFV PTP PassRate.....	24
PFV ToD PassRate.....	24
PFV FieldErrorCount <fieldName> (Deprecated).....	24
PFV PTP FieldErrorCount <fieldName>.....	25
PFV ToD FieldErrorCount <messageType> <fieldName>.....	25
PFV PTP TLVFieldErrorCount <tlvName> <tlvField>.....	25
PFV Transitions <messageType> <fieldName>.....	25
PFV PTP Transitions <messageType> <fieldName>.....	26
PFV PTP HSRTransitions <messageType> <fieldName>.....	27
PFV PTP PRPTransitions <messageType> <fieldName>.....	27
PFV PTP TLVTransitions <messageType> <tlvName> <tlvField>.....	28
PFV ToD G8271Transitions <messageType> <fieldName>.....	29
PFV ToD Transitions <fieldName>.....	30
PFV PTP TwoTimeTransmitters Transitions <timeTransmitterId> <messageType> <fieldName>.....	31
PFV PTP TwoTimeTransmitters TLVTransitions <timeTransmitterId> <messageType> <tlvName> <tlvField>.....	32
PFV PTP TwoTimeTransmitters HSRTransitions <timeTransmitterId> <messageType> <fieldName> .....	33
PFV PTP TwoTimeTransmitters PRPTransitions <timeTransmitterId> <messageType> <fieldName> .....	34
PFV PTP MessageRate <messageType>.....	35
PFV PTP TwoTimeTransmitters MessageRate <timeTransmitterId> <messageType>.....	35
PFV PTP DirectionalMessageRate <captureDirection> <messageType>.....	36
PFV PTP TwoTimeTransmitters DirectionalMessageRate <timeTransmitterId> <captureDirection> <messageType>.....	37
PFV PTP MessageRatePass <messageType>.....	38
PFV PTP TwoTimeTransmitters MessageRatePass <timeTransmitterId> <messageType>.....	38
PFV PTP DirectionalMessageRatePass <captureDirection> <messageType>.....	39
PFV PTP TwoTimeTransmitters DirectionalMessageRatePass <timeTransmitterId> <messageType> .....	40
PFV PTP MessageIntervalArithmeticMean <communicationType> <messageType>.....	41
PFV PTP TwoTimeTransmitters MessageIntervalArithmeticMean <timeTransmitterId> <communicationType> <messageType>.....	42
PFV PTP MessageIntervalArithmeticMeanPass <communicationType> <messageType>.....	43
PFV PTP TwoTimeTransmitters MessageIntervalArithmeticMeanPass <timeTransmitterId> <communicationType> <messageType>.....	44
PFV PTP MessageIntervalPassPercentage <communicationType> <messageType>.....	45
PFV PTP TwoTimeTransmitters MessageIntervalPassPercentage <timeTransmitterId> <communicationType> <messageType>.....	46
PFV PTP MessageIntervalPassPercentagePass <communicationType> <messageType>.....	47
PFV PTP TwoTimeTransmitters MessageIntervalPassPercentagePass <timeTransmitterId> <communicationType> <messageType>.....	48
PFV PTP TwoTimeTransmitters MessageCount <timeTransmitterId> <captureDrection> <messageType>.....	49
PFV PTP MessageCountPass <captureDrection> <messageType>.....	50

PFV PTP TwoTimeTransmitters MessageCountPass <timeTransmitterId> <captureDrection> <messageType> .....	51
PFV PTP TwoMasters Transitions <masterId> <messageType> <fieldName> .....	52
PFV PTP TwoMasters TLVTransitions <masterId> <messageType> <tlvName> <tlvField>.....	53
PFV PTP TwoMasters HSRTransitions <masterId> <messageType> <fieldName> .....	54
PFV PTP TwoMasters PRPTransitions <masterId> <messageType> <fieldName>.....	55
PFV PTP TwoMasters MessageRate <masterId> <messageType> .....	56
PFV PTP TwoMasters DirectionalMessageRate <masterId> <captureDirection> <messageType> ..	57
PFV PTP TwoMasters MessageRatePass <masterId> <messageType> .....	58
PFV PTP TwoMasters DirectionalMessageRatePass <masterId> <messageType> .....	59
PFV PTP TwoMasters MessageIntervalArithmeticMean <masterId> <communicationType> <messageType> .....	60
PFV PTP TwoMasters MessageIntervalArithmeticMeanPass <masterId> <communicationType> <messageType> .....	61
PFV PTP TwoMasters MessageIntervalPassPercentage <masterId> <communicationType> <messageType> .....	62
PFV PTP TwoMasters MessageIntervalPassPercentagePass <masterId> <communicationType> <messageType> .....	63
PFV PTP TwoMasters MessageCount <masterId> <captureDrection> <messageType> .....	64
PFV PTP TwoMasters MessageCountPass <masterId> <captureDrection> <messageType> .....	65
PFV Table Count (Deprecated) .....	66
PFV PTP Table Count.....	66
PFV ToD Table Count.....	66
PFV Table Data <rowOffset> <rowCount> (Deprecated) .....	66
PFV PTP Table Data <rowOffset> <rowCount> .....	67
PFV ToD Table Data <rowOffset> <rowCount> .....	67
PFV PTP Table Diagnostic <enable> .....	67
PFV PTP Table Export <file> .....	68
PFV PTP Statistics Export <file> .....	68
PFV AutoRulesFileSelect <enable>.....	68
PFV Close .....	68

## Control Commands

The commands in this section control the basic operation of the Paragon/PFV remote control. These commands are the baseline for further operations.

Note: Several commands in this section do not use the paragonset / paragonget mechanism. The command format for these commands is listed explicitly for each supported scripting language.

---

### connect

<b>Description</b>	Makes a connection to the specified GUI and instrument.
<b>Tcl</b>	<code>connect &lt;instrIpAddress&gt; [&lt;guiHostname&gt; [&lt;instrPort&gt; [&lt;rmtPort&gt;]]]</code>
<b>Python</b>	<code>p.connect("&lt;instrIpAddress&gt; [&lt;guiHostname&gt; [&lt;rmtPort&gt; [&lt;instrPort&gt;]]]")</code>
<b>Parameters</b>	<p><b>&lt;instrIpAddress&gt;</b> <b>Paragon-X:</b> The IP address of the Paragon instrument.  <b>Paragon-100G / Paragon-neo:</b> Must be "localhost".</p> <p><b>&lt;guiHostname&gt;</b> <b>Paragon-X:</b> This is an optional parameter, which specifies the IP address of the PC that is hosting the GUI; if this parameter is not specified then a connection to the local machine is assumed.</p> <p><b>&lt;instrPort&gt;</b> <b>Paragon-100G / Paragon-neo:</b> The IP address of the instrument. This is an optional parameter, which specifies the TCP Port on the Paragon instrument; if this parameter is not specified then port 9990 is assumed.</p> <p><b>&lt;rmtPort&gt;</b> This is an optional parameter, which specifies the remote-control TCP Port used by the Paragon client application; if this parameter is not specified then port 9000 is assumed.</p>
<b>Result</b>	The command will only report on failure if an error occurs. This operation will block until a connection is made.

---

### disconnect

<b>Description</b>	Disconnects the currently connected GUI and instrument.
<b>Tcl</b>	<code>disconnect</code>
<b>Python</b>	<code>p.disconnect()</code>
<b>Result</b>	Return with text displaying status of connection.

## PFV Commands

The commands in this section are specific to the PFV application.

---

### PFV OpenFile

Set	
<b>Description</b>	Loads a file into the PFV and analyzes it.
<b>Command</b>	PFV OpenFile <fileName>
<b>Parameters</b>	<fileName> The full path to a capture file.

---

### PFV Rules (Deprecated)

*This command is deprecated and should be replaced with PFV PTP Rules. See PTP PVF Rules for description.*

---

### PFV PTP Rules

Set	
<b>Description</b>	Changes the rules file used for analysis of PTP captures. If there was a capture file already loaded, it will be re-analyzed.
<b>Command</b>	PFV PTP Rules <rulesName>
<b>Parameters</b>	<rulesName> The name of a rules file. Note, the path to the file should not be included, only its name, as displayed in the PFV.
<b>Pre-requisites</b>	The data file must have been captured on an instrument with an appropriate option or the PC running PFV must be licensed.
<b>Result</b>	If no option or license is detected, then an error will be generated. If the rules file cannot be found, then an error will be generated.
Get	
<b>Description</b>	Returns the name of currently used rules file.
<b>Command</b>	PFV PTP Rules
<b>Pre-requisites</b>	See above.
<b>Result</b>	Returns the name of currently used rules file or an empty string if no file has been selected.

---

**PFV ToD Rules**

<b>Set</b>	
<b>Description</b>	Changes the rules file used for analysis of ToD captures. If there was a ToD capture file already loaded, it will be re-analyzed.
<b>Command</b>	PFV ToD Rules <rulesName>
<b>Parameters</b>	<rulesName> The name of a rules file. Note, the path to the file should not be included, only its name, as displayed in the PFV.
<b>Pre-requisites</b>	The data file must have been captured on an instrument with an appropriate option or the PC running PFV must be licensed.
<b>Result</b>	If no option or license is detected, then an error will be generated. If the rules file cannot be found, then an error will be generated.
<b>Get</b>	
<b>Description</b>	Returns the name of currently used rules file.
<b>Command</b>	PFV ToD Rules
<b>Pre-requisites</b>	See above.
<b>Result</b>	Returns the name of currently used rules file or an empty string if no file has been selected.

---

**PFV Show**

<b>Set</b>	
<b>Description</b>	Launches the PFV application.
<b>Command</b>	PFV Show <visible>
<b>Parameters</b>	<visible> Set to TRUE to launch the PFV.

---

**PFV GenerateReport <filename>**

Set	
<b>Description</b>	Generates a report in either pdf or html formats based on the PFV analysis results. The report is automatically saved in <Documents>\Calnex\PFV\Reports.
<b>Command</b>	PFV GenerateReport <filename>
<b>Parameters</b>	<p>&lt;filename&gt; The path and filename of the report to be generated. The extension should be either “.pdf” or “.html”. This is an optional parameter. If omitted, default file name and path will be used.</p> <p>You can include a page size as part of the filename when generating a pdf report. For example “C:/Directory/report.pdf (Letter)” will generate a report named “report.pdf” with a ‘Letter’ page size of 215.9mm by 279.4mm.</p> <p>Sizes A4 and Letter are supported, A4 will be used if no size is specified.</p>
<b>Pre-requisites</b>	The data file must have been captured on an instrument with an appropriate option or the PC running PFV must be licensed. A capture file must be already loaded by the PFV and a rules file selected.

---

**PFV PassResult (Deprecated)**

*This command is deprecated and should be replaced with PFV PTP PassResult. See PTP PFV PassResult for description.*

---

**PFV PTP PassResult**

Get	
<b>Description</b>	Returns the overall pass/fail result of the PFV analysis of the currently-loaded PTP capture file.
<b>Command</b>	PFV PTP PassResult
<b>Pre-requisites</b>	A PTP file must be loaded and have been analyzed by the PFV.
<b>Result</b>	Returns the result of the PFV analysis. If TRUE, the analysis passed all rules checks; if FALSE, then at least one rules check has failed.

---

**PFV ToD PassResult**

Get	
<b>Description</b>	Returns the overall pass/fail result of the PFV analysis of the currently-loaded ToD capture file.
<b>Command</b>	PFV ToD PassResult
<b>Pre-requisites</b>	A ToD file must be loaded and have been analyzed by the PFV.
<b>Result</b>	Returns the result of the PFV analysis. If TRUE, the analysis passed all rules checks; if FALSE, then at least one rules check has failed.

---

**PFV PassRate (Deprecated)**

*This command is deprecated and should be replaced with PFV PTP PassRate.  
See PFV PTP PassRate for description.*

---

**PFV PTP PassRate**

Get	
<b>Description</b>	Returns the percentage of packets that have passed the PFV analysis for the currently-loaded PTP capture.
<b>Command</b>	PFV PTP PassRate
<b>Pre-requisites</b>	A PTP file must be loaded and have been analyzed by the PFV.
<b>Result</b>	Returns the percentage of packets that have passed the analysis.

---

**PFV ToD PassRate**

Get	
<b>Description</b>	Returns the percentage of packets that have passed the PFV analysis for the currently-loaded ToD capture.
<b>Command</b>	PFV ToD PassRate
<b>Pre-requisites</b>	A ToD file must be loaded and have been analyzed by the PFV.
<b>Result</b>	Returns the percentage of packets that have passed the analysis.

---

**PFV FieldErrorCount <fieldName> (Deprecated)**

*This command is deprecated and should be replaced with PFV PTP FieldErrorCount.  
See PFV PTP FieldErrorCount for description.*



---

**PFV PTP FieldErrorCount <fieldName>**

Get	
<b>Description</b>	Returns the number of failures for the specified message field.
<b>Command</b>	PFV PTP FieldErrorCount <fieldName>
<b>Parameters</b>	<fieldName> The message field name. The field name is the same as is used in the PFV rules files and in the PFV UI. See the <i>PFV Getting Started Guide</i> for a complete list of field names.
<b>Pre-requisites</b>	A PTP file must be loaded and have been analyzed by the PFV.
<b>Result</b>	Returns an integer value equal to the number of detected failures in the specified field.

---

**PFV ToD FieldErrorCount <messageType> <fieldName>**

Get	
<b>Description</b>	Returns the number of failures for the specified message field.
<b>Command</b>	PFV FieldErrorCount <messageType> <fieldName>
<b>Parameters</b>	<messageType> The ToD message type. This must be one of the following: CcsaTimeInformation CcsaTimeEvent G8271TimeEvent G8271TimeAnnounce G8271GnssStatus  <fieldName> The message field name. The field name is the same as is used in the PFV rules files and in the PFV UI. See the <i>PFV Getting Started Guide</i> for a complete list of field names.
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.
<b>Result</b>	Returns an integer value equal to the number of detected failures in the specified field.

---

**PFV PTP TLVFieldErrorCount <tlvName> <tlvField>**

Get	
<b>Description</b>	Returns the number of failures for the specified TLV field.
<b>Command</b>	PFV PTP TLVFieldErrorCount <tlvName > <tlvField>
<b>Parameters</b>	<tlvName> The TLV being queried. The name must come from the "TLV Name" in the table in section "TLV Field Names".  <tlvField> The TLV field name. Valid field names can be found in the table in section "TLV Field Names".
<b>Pre-requisites</b>	A PTP file must be loaded and have been analyzed by the PFV.
<b>Result</b>	Returns an integer value equal to the number of detected failures in the specified field.

---

**PFV Transitions <messageType> <fieldName>**

*This command is deprecated and should be replaced with PFV PTP Transitions. See PFV PTP Transitions for description.*

**PFV PTP Transitions <messageType> <fieldName>**

<b>Get</b>										
<b>Description</b>	Returns a set of comma-separated values detailing how a field changed its value throughout the duration of the PTP capture. Each range has a set of delimiting packet indexes and timestamps associated with it.									
<b>Command</b>	PFV PTP Transitions <messageType> <fieldName>									
<b>Parameters</b>	<p>&lt;messageType&gt; The PTP message type. This must be one of the following:</p> <table style="margin-left: 40px; border: none;"> <tr> <td>ANNOUNCE</td> <td>SYNC</td> <td>FOLLOWUP</td> </tr> <tr> <td>DELREQ</td> <td>DELRESP</td> <td>MANAGEMENT</td> </tr> <tr> <td>PDELREQ</td> <td>PDELRESP</td> <td>PDELRESPFUP</td> </tr> </table> <p>&lt;fieldname&gt; The message field name. The field names are listed in "PTP Field Names" above.</p>	ANNOUNCE	SYNC	FOLLOWUP	DELREQ	DELRESP	MANAGEMENT	PDELREQ	PDELRESP	PDELRESPFUP
ANNOUNCE	SYNC	FOLLOWUP								
DELREQ	DELRESP	MANAGEMENT								
PDELREQ	PDELRESP	PDELRESPFUP								
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.									
<b>Result</b>	<p>Returns a set of comma-separated values. Example output:</p> <p><i>Value, PacketNumberFrom, PacketNumberTo, PacketTimestampFrom, PacketTimestampTo</i></p> <p><i>6, 2, 68, 0.02, 2.2</i></p> <p>The above example would mean that the field value was 6 from the 2<sup>nd</sup> to 68<sup>th</sup> packet or from 0.02s to 2.2s.</p>									

**PFV PTP HSRTransitions <messageType> <fieldName>**

<b>Get</b>												
<b>Description</b>	Returns a set of comma-separated values detailing how an HSR (High availability Seamless Redundancy protocol) field changed its value throughout the duration of the PTP capture. Each range has a set of delimiting packet indexes and timestamps associated with it.											
<b>Command</b>	PFV PTP HSRTransitions <messageType> <fieldName>											
<b>Parameters</b>	<p>&lt;messageType&gt; The PTP message type. This must be one of the following:</p> <table style="margin-left: 40px;"> <tr> <td>ANNOUNCE</td> <td>SYNC</td> <td>FOLLOWUP</td> </tr> <tr> <td>DELREQ</td> <td>DELRESP</td> <td>MANAGEMENT</td> </tr> <tr> <td>PDELREQ</td> <td>PDELRESP</td> <td>PDELRESPFUP</td> </tr> </table> <p>&lt;fieldname&gt; The message field name. This must be one of the following:</p> <table style="margin-left: 40px;"> <tr> <td>NetId</td> <td>LanId</td> </tr> </table>	ANNOUNCE	SYNC	FOLLOWUP	DELREQ	DELRESP	MANAGEMENT	PDELREQ	PDELRESP	PDELRESPFUP	NetId	LanId
ANNOUNCE	SYNC	FOLLOWUP										
DELREQ	DELRESP	MANAGEMENT										
PDELREQ	PDELRESP	PDELRESPFUP										
NetId	LanId											
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.											
<b>Result</b>	<p>Returns a set of comma-separated values. Example output:</p> <p><i>Value, PacketNumberFrom, PacketNumberTo, PacketTimestampFrom, PacketTimestampTo, 6, 2, 68, 0.02, 2.2</i></p> <p>The above example would mean that the field value was 6 from the 2<sup>nd</sup> to 68<sup>th</sup> packet or from 0.02s to 2.2s.</p>											

**PFV PTP PRPTransitions <messageType> <fieldName>**

<b>Get</b>											
<b>Description</b>	Returns a set of comma-separated values detailing how a PRP (Paralled Redundancy Protocol) field changed its value throughout the duration of the PTP capture. Each range has a set of delimiting packet indexes and timestamps associated with it.										
<b>Command</b>	PFV PTP PRPTransitions <messageType> <fieldName>										
<b>Parameters</b>	<p>&lt;messageType&gt; The PTP message type. This must be one of the following:</p> <table style="margin-left: 40px;"> <tr> <td>ANNOUNCE</td> <td>SYNC</td> <td>FOLLOWUP</td> </tr> <tr> <td>DELREQ</td> <td>DELRESP</td> <td>MANAGEMENT</td> </tr> <tr> <td>PDELREQ</td> <td>PDELRESP</td> <td>PDELRESPFUP</td> </tr> </table> <p>&lt;fieldname&gt; The message field name. This must be one of the following:</p> <table style="margin-left: 40px;"> <tr> <td>LanId</td> </tr> </table>	ANNOUNCE	SYNC	FOLLOWUP	DELREQ	DELRESP	MANAGEMENT	PDELREQ	PDELRESP	PDELRESPFUP	LanId
ANNOUNCE	SYNC	FOLLOWUP									
DELREQ	DELRESP	MANAGEMENT									
PDELREQ	PDELRESP	PDELRESPFUP									
LanId											
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.										
<b>Result</b>	<p>Returns a set of comma-separated values. Example output:</p> <p><i>Value, PacketNumberFrom, PacketNumberTo, PacketTimestampFrom, PacketTimestampTo, 6, 2, 68, 0.02, 2.2</i></p> <p>The above example would mean that the field value was 6 from the 2<sup>nd</sup> to 68<sup>th</sup> packet or from 0.02s to 2.2s.</p>										

**PFV PTP TLVTransitions <messageType> <tlvName> <tlvField>**

<b>Get</b>										
<b>Description</b>	Returns a set of comma-separated values detailing how a TLV field changed its value throughout the duration of the PTP capture. Each range has a set of delimiting packet indexes and timestamps associated with it. The range also indicates the zero-based index of the TLV in the message (which may contain multiple TLVs).									
<b>Command</b>	PFV PTP TLVTransitions <messageType> <tlvName> <tlvField>									
<b>Parameters</b>	<p>&lt;messageType&gt; The PTP message type. This must be one of the following:</p> <table style="margin-left: 40px; border: none;"> <tr> <td>ANNOUNCE</td> <td>SYNC</td> <td>FOLLOWUP</td> </tr> <tr> <td>DELREQ</td> <td>DELRESP</td> <td>MANAGEMENT</td> </tr> <tr> <td>PDELREQ</td> <td>PDELRESP</td> <td>PDELRESPFUP</td> </tr> </table> <p>&lt;tlvName&gt; The TLV being queried. The name must come from the "TLV Name" in the table in section "TLV Field Names".</p> <p>&lt;tlvField&gt; The TLV field name. Valid field names can be found in the table in section "TLV Field Names".</p>	ANNOUNCE	SYNC	FOLLOWUP	DELREQ	DELRESP	MANAGEMENT	PDELREQ	PDELRESP	PDELRESPFUP
ANNOUNCE	SYNC	FOLLOWUP								
DELREQ	DELRESP	MANAGEMENT								
PDELREQ	PDELRESP	PDELRESPFUP								
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.									
<b>Result</b>	<p>Returns a set of comma-separated values. Example output:</p> <pre>PacketNumberFrom,PacketNumberTo,PacketTimestampFrom,PacketTimestampTo, IndexOfTlv1,TlvValue1\n7,32,7E-05,0.0625,0,3,\n</pre> <p>The above example would mean that the TLV was found in messages from the 7<sup>th</sup> to the 32<sup>nd</sup> packet in the capture, or from time 70µs to 62.5ms, and that:</p> <ul style="list-style-type: none"> <li>• The TLV was found at index 0 throughout this time, and</li> <li>• The value of the requested field was 3</li> </ul>									

**PFV ToD G8271Transitions <messageType> <fieldName>**

<b>Get</b>	
<b>Description</b>	Returns a set of comma-separated values detailing how a field changed its value throughout the duration of the G.8271 ToD capture. Each range has a set of delimiting message indexes and timestamps associated with it.
<b>Command</b>	PFV ToD G8271Transitions <messageType> <fieldName>
<b>Parameters</b>	<p><i>&lt;messageType&gt;</i> The message type. One of the following:            TimeEvent            TimeAnnounce            GnssStatus</p> <p><i>&lt;fieldName&gt;</i> The message field name. The available field names for specific message types are listed below.</p> <p>TimeEvent:            Flags    CurrentUtcOffset</p> <p>TimeAnnounce:            VersionPtp    DomainNumber    FlagField            SourceClockIdentity    GmPriority1    GmPriority2            ClockClass    ClockAccuracy    Oslv    GmClockIdentity            StepsRemoved    TimeSource</p> <p>GnssStatus:            TimeSource    Status    Alarms</p>
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.
<b>Result</b>	<p>Returns a set of comma-separated values. Example output:</p> <p><i>Value, PacketNumberFrom, PacketNumberTo, PacketTimestampFrom, PacketTimestampTo, 6, 2, 68, 2, 68</i></p> <p>The above example would mean that the field value was 6 from the 2<sup>nd</sup> to 68<sup>th</sup> packet or from 2s to 68s.</p>

---

**PFV ToD Transitions <fieldName>**

<b>Get</b>	
<b>Description</b>	Returns a set of comma-separated values detailing how a field changed its value throughout the duration of the ToD capture. Each range has a set of delimiting message indexes and timestamps associated with it. This command is specific to CCSA messages for backwards compatibility.
<b>Command</b>	PFV ToD Transitions <fieldName>
<b>Parameters</b>	<i>&lt;fieldName&gt;</i> The message field name. Available values are: Week LeapS PPSStatus TAcc TimeSourceType TimesSourceStatus Alarms
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.
<b>Result</b>	Returns a set of comma-separated values. Example output: <i>Value, PacketNumberFrom, PacketNumberTo, PacketTimestampFrom, PacketTimestampTo, 6, 2, 68, 2, 68</i> The above example would mean that the field value was 6 from the 2 <sup>nd</sup> to 68 <sup>th</sup> packet or from 2s to 68s.

**PFV PTP TwoTimeTransmitters Transitions <timeTransmitterId> <messageType> <fieldName>**

<b>Get</b>	
<b>Description</b>	<p>This extended version of the PTP Transitions command is provided to query the analysis of Paragon-neo 'Two TimeTransmitters' captures where message captures from two disitinct timeTransmitter devices are present in a single capture file.</p> <p>Returns a set of comma-separated values detailing how a field changed its value throughout the duration of the PTP capture. Each range has a set of delimiting packet indexes and timestamps associated with it.</p>
<b>Command</b>	<p>PFV PTP TwoTimeTransmitters Transitions &lt;timeTransmitterId&gt; &lt;messageType&gt; &lt;fieldName&gt;</p>
<b>Parameters</b>	<p>&lt;timeTransmitterId&gt; The PTP timeTransmitter identifier. This must be one of the following:  TIMETRANSMITTER1 TIMETRANSMITTER2</p> <p>&lt;messageType&gt; The PTP message type. This must be one of the following:  ANNOUNCE SYNC FOLLOWUP  DELREQ DELRESP MANAGEMENT  PDELREQ PDELRESP PDELRESFPUP</p> <p>&lt;fieldname&gt; The message field name. The field names are listed in "PTP Field Names" above.</p>
<b>Pre-requisites</b>	<p>A Paragon-neo 'Two TimeTransmitters' PTP capture file must be loaded and have been analyzed by the PFV.</p>
<b>Result</b>	<p>Returns a set of comma-separated values. Example output:</p> <p><i>Value, PacketNumberFrom, PacketNumberTo, PacketTimestampFrom, PacketTimestampTo, 6, 2, 68, 0.02, 2.2</i></p> <p>If we assume that the example output above was returned by specifying 'TIMETRANSMITTER1' as the &lt;timeTransmitterId&gt; this would mean that the field value was 6 from 0.02s to 2.2s for 'TIMETRANSMITTER1' and the first and last packet numbers in this range were 2 and 68 respectively. It does not mean that every packet between 2 and 68 (exclusive) originated at 'TIMETRANSMITTER1'.</p>

**PFV PTP TwoTimeTransmitters TLVTransitions <timeTransmitterId> <messageType> <tlvName> <tlvField>**

<b>Get</b>	
<b>Description</b>	<p>This extended version of the PTP TLVTransitions command is provided to query the analysis of Paragon-neo 'Two TimeTransmitters' or 'Two TimeRecievers' captures where message captures from two disitinct timeTransmitter or timeReciever devices are present in a single capture file.</p> <p>Returns a set of comma-separated values detailing how a TLV field changed its value throughout the duration of the PTP capture. Each range has a set of delimiting packet indexes and timestamps associated with it. The range also indicates the zero-based index of the TLV in the message (which may contain multiple TLVs).</p>
<b>Command</b>	PFV PTP TwoTimeTransmitters TLVTransitions <timeTransmitterId> <messageType> <tlvName> <tlvField>
<b>Parameters</b>	<p><i>&lt;timeTransmitterId&gt;</i> The PTP timeTransmitter identifier. This must be one of the following: TIMETRANSMITTER1 TIMETRANSMITTER2</p> <p><i>&lt;messageType&gt;</i> The PTP message type. This must be one of the following: ANNOUNCE SYNC FOLLOWUP DELREQ DELRESP MANAGEMENT PDELREQ PDELRESP PDELRESPFUP</p> <p><i>&lt;tlvName&gt;</i> The TLV being queried. The name must come from the "TLV Name" in the table in section "TLV Field Names".</p> <p><i>&lt;tlvField&gt;</i> The TLV field name. Valid field names can be found in the table in section "TLV Field Names".</p>
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.
<b>Result</b>	<p>Returns a set of comma-separated values. Example output: PacketNumberFrom,PacketNumberTo,PacketTimestampFrom,PacketTimestampTo,IndexOfTlv1,TlvValue1\n7,32,7E-05,0.0625,0,3,\n</p> <p>The above example would mean that the TLV was found in messages from the 7<sup>th</sup> to th 32ns packet in the capture, or from time 70μs to 62.5ms, and that:</p> <ul style="list-style-type: none"> <li>• The TLV was found at index 0 throughout this time, and</li> <li>• The value of the requested field was 3</li> </ul>



**PFV PTP TwoTimeTransmitters HSRTransitions <timeTransmitterId> <messageType> <fieldName>**

<b>Get</b>	
<b>Description</b>	<p>This extended version of the PTP HSRTransitions command is provided to query the analysis of Paragon-neo 'Two TimeTransmitters' captures where message captures from two disitinct timeTransmitter devices are present in a single capture file.</p> <p>Returns a set of comma-separated values detailing how an HSR (High availability Seamless Redundnacy protocol) field changed its value throughout the duration of the PTP capture. Each range has a set of delimiting packet indexes and timestamps associated with it.</p>
<b>Command</b>	PFV PTP TwoTimeTransmitters HSRTransitions <timeTransmitterId> <messageType> <fieldName>
<b>Parameters</b>	<p>&lt;timeTransmitterId&gt; The PTP timeTransmitter identifier. This must be one of the following:  TIMETRANSMITTER1 TIMETRANSMITTER2</p> <p>&lt;messageType&gt; The PTP message type. This must be one of the following:  ANNOUNCE SYNC FOLLOWUP  DELREQ DELRESP MANAGEMENT  PDELREQ PDELRESP PDELRESPFUP</p> <p>&lt;fieldname&gt; The message field name. This must be one of the following:  NetId LanId</p>
<b>Pre-requisites</b>	A Paragon-neo 'Two TimeTransmitters' PTP capture file must be loaded and have been analyzed by the PFV.
<b>Result</b>	<p>Returns a set of comma-separated values. Example output:</p> <p><i>Value, PacketNumberFrom, PacketNumberTo, PacketTimestampFrom, PacketTimestampTo, 6, 2, 68, 0.02, 2.2</i></p> <p>If we assume that the example output above was returned by specifying 'TIMETRANSMITTER1' as the &lt;timeTransmitterId&gt; this would mean that the field value was 6 from 0.02s to 2.2s for 'TIMETRANSMITTER1' and the first and last packet numbers in this range were 2 and 68 respectively. It does not mean that every packet between 2 and 68 (exclusive) originated at 'TIMETRANSMITTER1'.</p>

**PFV PTP TwoTimeTransmitters PRPTransitions <timeTransmitterId> <messageType> <fieldName>**

<b>Get</b>	
<b>Description</b>	<p>This extended version of the PTP PRPTransitions command is provided to query the analysis of Paragon-neo 'Two TimeTransmitters' captures where message captures from two disitinct timeTransmitter devices are present in a single capture file.</p> <p>Returns a set of comma-separated values detailing how a PRP (Parallel Redundnacy Protocol) field changed its value throughout the duration of the PTP capture. Each range has a set of delimiting packet indexes and timestamps associated with it.</p>
<b>Command</b>	PFV PTP TwoTimeTransmitters PRPTransitions <timeTransmitterId> <messageType> <fieldName>
<b>Parameters</b>	<p>&lt;timeTransmitterId&gt; The PTP timeTransmitter identifier. This must be one of the following:  TIMETRANSMITTER1 TIMETRANSMITTER2</p> <p>&lt;messageType&gt; The PTP message type. This must be one of the following:  ANNOUNCE SYNC FOLLOWUP  DELREQ DELRESP MANAGEMENT  PDELREQ PDELRESP PDELRESPFUP</p> <p>&lt;fieldname&gt; The message field name. This must be one of the following:  LanId</p>
<b>Pre-requisites</b>	A Paragon-neo 'Two TimeTransmitters' PTP capture file must be loaded and have been analyzed by the PFV.
<b>Result</b>	<p>Returns a set of comma-separated values. Example output:</p> <p><i>Value, PacketNumberFrom, PacketNumberTo, PacketTimestampFrom, PacketTimestampTo</i>  6, 2, 68, 0.02, 2.2</p> <p>If we assume that the example output above was returned by specifying 'TIMETRANSMITTER1' as the &lt;timeTransmitterId&gt; this would mean that the field value was 6 from 0.02s to 2.2s for 'TIMETRANSMITTER1' and the first and last packet numbers in this range were 2 and 68 respectively. It does not mean that every packet between 2 and 68 (exclusive) originated at 'TIMETRANSMITTER1'.</p>

---

**PFV PTP MessageRate <messageType>**

<b>Get</b>													
<b>Description</b>	Returns the message rate for the given message type.												
<b>Command</b>	PFV PTP MessageRate <messageType>												
<b>Parameters</b>	<p>&lt;messageType&gt; The PTP message type. This must be one of the following:</p> <table style="margin-left: 40px; border: none;"> <tr> <td>SYNC</td> <td>FOLLOWUP</td> <td></td> </tr> <tr> <td>DELREQ</td> <td>DELRESP</td> <td></td> </tr> <tr> <td>PDELREQ</td> <td>PDELRESP</td> <td>PDELRESPFUP</td> </tr> <tr> <td>ANNOUNCE</td> <td>SIGNALING</td> <td>MANAGEMENT</td> </tr> </table>	SYNC	FOLLOWUP		DELREQ	DELRESP		PDELREQ	PDELRESP	PDELRESPFUP	ANNOUNCE	SIGNALING	MANAGEMENT
SYNC	FOLLOWUP												
DELREQ	DELRESP												
PDELREQ	PDELRESP	PDELRESPFUP											
ANNOUNCE	SIGNALING	MANAGEMENT											
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.												
<b>Result</b>	Returns the rate of the requested <messageType> in packets per second. If the requested <messageType> is not present in the capture file "N/A" will be returned.												

---

**PFV PTP TwoTimeTransmitters MessageRate <timeTransmitterId> <messageType>**

<b>Get</b>															
<b>Description</b>	<p>This extended version of the MessageRate command is provided to query the analysis of Paragon-neo 'Two TimeTransmitters' captures where message captures from two disitinct timeTransmitter devices are present in a single capture file.</p> <p>Returns the message rate for the given message type and timeTransmitter device identifier.</p>														
<b>Command</b>	PFV PTP TwoTimeTransmitters MessageRate <timeTransmitterId> <messageType>														
<b>Parameters</b>	<p>&lt;timeTransmitterId&gt; The PTP timeTransmitter identifier. This must be one of the following:</p> <table style="margin-left: 40px; border: none;"> <tr> <td>TIMETRANSMITTER1</td> <td>TIMETRANSMITTER2</td> </tr> </table> <p>&lt;messageType&gt; The PTP message type. This must be one of the following:</p> <table style="margin-left: 40px; border: none;"> <tr> <td>SYNC</td> <td>FOLLOWUP</td> <td></td> </tr> <tr> <td>DELREQ</td> <td>DELRESP</td> <td></td> </tr> <tr> <td>PDELREQ</td> <td>PDELRESP</td> <td>PDELRESPFUP</td> </tr> <tr> <td>ANNOUNCE</td> <td>SIGNALING</td> <td>MANAGEMENT</td> </tr> </table>	TIMETRANSMITTER1	TIMETRANSMITTER2	SYNC	FOLLOWUP		DELREQ	DELRESP		PDELREQ	PDELRESP	PDELRESPFUP	ANNOUNCE	SIGNALING	MANAGEMENT
TIMETRANSMITTER1	TIMETRANSMITTER2														
SYNC	FOLLOWUP														
DELREQ	DELRESP														
PDELREQ	PDELRESP	PDELRESPFUP													
ANNOUNCE	SIGNALING	MANAGEMENT													
<b>Pre-requisites</b>	A Paragon-neo 'Two TimeTransmitters capture file must be loaded and have been analyzed by the PFV.														
<b>Result</b>	Returns the rate of the requested <messageType> in packets per second for the specified <timeTransmitterId>. If the requested <messageType> is not present in the capture file for the specified <timeTransmitterId> "N/A" will be returned.														

**PFV PTP DirectionalMessageRate <captureDirection> <messageType>**

<b>Get</b>	
<b>Description</b>	Returns the message rate for the given message type.
<b>Command</b>	PFV PTP DirectionalMessageRate <captureDirection> <messageType>
<b>Parameters</b>	<p><i>&lt;captureDirection&gt;</i> The direction the message was captured at the measurement port. Tx is a transmitted message at the port, Rx is a received message at the port.</p> <p style="text-align: center;">Tx                      Rx                      Any</p> <p><i>&lt;messageType&gt;</i> The PTP message type. This must be one of the following:</p> <p style="text-align: center;">SYNC                      FOLLOWUP  DELREQ                      DELRESP  PDELREQ                      PDELRESP                      PDELRESPFUP  ANNOUNCE                      SIGNALING                      MANAGEMENT</p>
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.
<b>Result</b>	Returns the rate of the requested <messageType> in packets per second. If the requested <messageType> is not present in the capture file "N/A" will be returned.

**PFV PTP TwoTimeTransmitters DirectionalMessageRate <timeTransmitterId> <captureDirection> <messageType>**

<b>Get</b>	
<b>Description</b>	This extended version of the MessageRate command is provided to query the analysis of Paragon-neo 'Two TimeTransmitters' captures where message captures from two disitinct timeTransmitter devices are present in a single capture file.  Returns the message rate for the given message type and timeTransmitter device identifier.
<b>Command</b>	PFV PTP TwoTimeTransmitters DirectionalMessageRate <timeTransmitterId> <captureDirection> <messageType>
<b>Parameters</b>	<p><i>&lt;timeTransmitterId&gt;</i> The PTP timeTransmitter identifier. This must be one of the following: TIMETRANSMITTER1 TIMETRANSMITTER2</p> <p><i>&lt;captureDirection&gt;</i> The direction the message was captured at the selected port. Tx is a transmitted message at the port, Rx is a received message at the port. Tx                      Rx                      Any</p> <p><i>&lt;messageType&gt;</i> The PTP message type. This must be one of the following: SYNC                      FOLLOWUP DELREQ                      DELRESP PDELREQ                      PDELRESP                      PDELRESPFUP ANNOUNCE                      SIGNALING                      MANAGEMENT</p>
<b>Pre-requisites</b>	A Paragon-neo 'Two TimeTransmitters' capture file must be loaded and have been analyzed by the PFV.
<b>Result</b>	Returns the rate of the requested <messageType> in packets per second for the specified <timeTransmitterId>. If the requested <messageType> is not present in the capture file for the specified <timeTransmitterId> "N/A" will be returned.

**PFV PTP MessageRatePass <messageType>**

<b>Get</b>													
<b>Description</b>	Returns a Boolean result indicating the success or failure of a message rate rule for the specified message type.												
<b>Command</b>	PFV PTP MessageRatePass <messageType>												
<b>Parameters</b>	<p>&lt;messageType&gt; The PTP message type. This must be one of the following:</p> <table style="margin-left: 40px;"> <tr> <td>SYNC</td> <td>FOLLOWUP</td> <td></td> </tr> <tr> <td>DELREQ</td> <td>DELRESP</td> <td></td> </tr> <tr> <td>PDELREQ</td> <td>PDELRESP</td> <td>PDELRESPFUP</td> </tr> <tr> <td>ANNOUNCE</td> <td>SIGNALING</td> <td>MANAGEMENT</td> </tr> </table>	SYNC	FOLLOWUP		DELREQ	DELRESP		PDELREQ	PDELRESP	PDELRESPFUP	ANNOUNCE	SIGNALING	MANAGEMENT
SYNC	FOLLOWUP												
DELREQ	DELRESP												
PDELREQ	PDELRESP	PDELRESPFUP											
ANNOUNCE	SIGNALING	MANAGEMENT											
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.												
<b>Result</b>	<p>Returns the result of the relevant PFV analysis 'averageMessageRate' rule. If TRUE, the message rate passed the rule associated with the &lt;messageType&gt;; if FALSE then the rules check has failed.</p> <p>Note that if the specified &lt;messageType&gt; has no rules associated with it the result returned will be FALSE.</p>												

**PFV PTP TwoTimeTransmitters MessageRatePass <timeTransmitterId> <messageType>**

<b>Get</b>															
<b>Description</b>	<p>This extended version of the MessageRatePass command is provided to query the analysis of Paragon-neo 'Two TimeTransmitters' captures where message captures from two distinct timeTransmitter devices are present in a single capture file.</p> <p>Returns a Boolean result indicating the success or failure of a message rate rule for the specified message type and timeTransmitter.</p>														
<b>Command</b>	PFV PTP TwoTimeTransmitters MessageRatePass <timeTransmitterId> <messageType>														
<b>Parameters</b>	<p>&lt;timeTransmitterId&gt; The PTP timeTransmitter identifier. This must be one of the following:</p> <table style="margin-left: 40px;"> <tr> <td>TIMETRANSMITTER1</td> <td>TIMETRANSMITTER2</td> </tr> </table> <p>&lt;messageType&gt; The PTP message type. This must be one of the following:</p> <table style="margin-left: 40px;"> <tr> <td>SYNC</td> <td>FOLLOWUP</td> <td></td> </tr> <tr> <td>DELREQ</td> <td>DELRESP</td> <td></td> </tr> <tr> <td>PDELREQ</td> <td>PDELRESP</td> <td>PDELRESPFUP</td> </tr> <tr> <td>ANNOUNCE</td> <td>SIGNALING</td> <td>MANAGEMENT</td> </tr> </table>	TIMETRANSMITTER1	TIMETRANSMITTER2	SYNC	FOLLOWUP		DELREQ	DELRESP		PDELREQ	PDELRESP	PDELRESPFUP	ANNOUNCE	SIGNALING	MANAGEMENT
TIMETRANSMITTER1	TIMETRANSMITTER2														
SYNC	FOLLOWUP														
DELREQ	DELRESP														
PDELREQ	PDELRESP	PDELRESPFUP													
ANNOUNCE	SIGNALING	MANAGEMENT													
<b>Pre-requisites</b>	A Paragon-neo 'Two TimeTransmitters' capture file must be loaded and have been analyzed by the PFV.														
<b>Result</b>	<p>Returns the result of the relevant PFV analysis 'averageMessageRate' rule. If TRUE, the message rate passed the rule associated with the &lt;messageType&gt; and &lt;timeTransmitterId&gt;; if FALSE then the rules check has failed.</p> <p>Note that if the specified &lt;messageType&gt; and &lt;timeTransmitterId&gt; has no rules associated with it the result returned will be FALSE.</p>														

**PFV PTP DirectionalMessageRatePass <captureDirection> <messageType>**

<b>Get</b>	
<b>Description</b>	Returns a Boolean result indicating the success or failure of a message rate rule for the specified message type and capture direction.
<b>Command</b>	PFV PTP DirectionalMessageRatePass <captureDirection> <messageType>
<b>Parameters</b>	<p><i>&lt;captureDirection&gt;</i> The direction the message was captured at the measurement port. Tx is a transmitted message at the port, Rx is a received message at the port.</p> <p style="padding-left: 40px;">Tx                      Rx                      Any</p> <p><i>&lt;messageType&gt;</i> The PTP message type. This must be one of the following:</p> <p style="padding-left: 40px;">SYNC                      FOLLOWUP  DELREQ                      DELRESP  PDELREQ                      PDELRESP                      PDELRESPFUP  ANNOUNCE                      SIGNALING                      MANAGEMENT</p>
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.
<b>Result</b>	Returns the result of the relevant PFV analysis 'averageMessageRate' rule. If TRUE, the message rate passed the rule associated with the <messageType>; if FALSE then the rules check has failed. Note that if the specified <messageType> has no rules associated with it the result returned will be FALSE.

**PFV PTP TwoTimeTransmitters DirectionalMessageRatePass <timeTransmitterId> <messageType>**

<b>Get</b>	
<b>Description</b>	<p>This extended version of the DirectionalMessageRatePass command is provided to query the analysis of Paragon-neo 'Two TimeTransmitters' captures where message captures from two disitinct timeTransmitter devices are present in a single capture file.</p> <p>Returns a Boolean result indicating the success or failure of a message rate rule for the specified message type, capture direction and timeTransmitter.</p>
<b>Command</b>	<p>PFV PTP TwoTimeTransmitters DirectionalMessageRatePass &lt;timeTransmitterId&gt; &lt;captureDirection&gt; &lt;messageType&gt;</p>
<b>Parameters</b>	<p>&lt;timeTransmitterId&gt; The PTP timeTransmitter identifier. This must be one of the following:  TIMETRANSMITTER1 TIMETRANSMITTER2</p> <p>&lt;captureDirection&gt; The direction the message was captured at the selected port. Tx is a transmitted message at the port, Rx is a received message at the port.  Tx Rx Any</p> <p>&lt;messageType&gt; The PTP message type. This must be one of the following:  SYNC FOLLOWUP  DELREQ DELRESP  PDELREQ PDELRESP PDELRESPFUP  ANNOUNCE SIGNALING MANAGEMENT</p>
<b>Pre-requisites</b>	<p>A Paragon-neo 'Two TimeTransmitters' capture file must be loaded and have been analyzed by the PFV.</p>
<b>Result</b>	<p>Returns the result of the relevant PFV analysis 'averageMessageRate' rule. If TRUE, the message rate passed the rule associated with the &lt;messageType&gt;, &lt;captureDirection&gt; and &lt;timeTransmitterId&gt;; if FALSE then the rules check has failed. Note that if the specified &lt;messageType&gt;, &lt;captureDirection&gt; and &lt;timeTransmitterId&gt; has no rules associated with it the result returned will be FALSE.</p>



**PFV PTP MessageIntervalArithmeticMean <communicationType> <messageType>**

<b>Get</b>	
<b>Description</b>	Returns the arithmetic mean, in seconds to a picosecond resolution, of all inter-message intervals for the specified communication type and message type.
<b>Command</b>	PFV PTP MessageIntervalArithmeticMean <communicationType> <messageType>
<b>Parameters</b>	<p>&lt;communicationType&gt; PTP messages will be 'Unicast' or 'Multicast' determined by the value of the 'Unicast' flag in the common message header.</p> <p style="text-align: center;">UNICAST      MULTICAST</p> <p>&lt;messageType&gt; The PTP message type. This must be one of the following:</p> <p style="text-align: center;">SYNC              ANNOUNCE DELREQ          PDELREQ</p>
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.
<b>Result</b>	Returns the arithmetic mean of all inter-message intervals measured for the requested <communicationType> and <messageType> combination. The result returned is expressed in seconds to a maximum of 12 decimal places. If the requested <communicationType> and <messageType> is not present in the capture file "N/A" will be returned.

**PFV PTP TwoTimeTransmitters MessageIntervalArithmeticMean <timeTransmitterId>  
<communicationType> <messageType>**

<b>Get</b>	
<b>Description</b>	<p>This extended version of the MessageIntervalArithmeticMean command is provided to query the analysis of Paragon-neo 'Two TimeTransmitters' captures where message captures from two disitinct timeTransmitter devices are present in a single capture file.</p> <p>Returns the arithmetic mean, in seconds to a picosecond resolution, of all inter-message intervals for the specified communication type and message type on the specified timeTransmitter device identifier.</p>
<b>Command</b>	<p>PFV PTP TwoTimeTransmitters MessageIntervalArithmeticMean &lt;timeTransmitterId&gt; &lt;communicationType&gt; &lt;messageType&gt;</p>
<b>Parameters</b>	<p><i>&lt;timeTransmitterId&gt;</i> The PTP timeTransmitter identifier. This must be one of the following: TIMETRANSMITTER1 TIMETRANSMITTER2</p> <p><i>&lt;communicationType&gt;</i> PTP messages will be 'Unicast' or 'Multicast' determined by the value of the 'Unicast' flag in the common message header. UNICAST MULTICAST</p> <p><i>&lt;messageType&gt;</i> The PTP message type. This must be one of the following: SYNC ANNOUNCE DELREQ PDELREQ</p>
<b>Pre-requisites</b>	<p>A Paragon-neo 'Two TimeTransmitters' capture file must be loaded and have been analyzed by the PFV.</p>
<b>Result</b>	<p>Returns the arithmetic mean of all inter-message intervals measured for the requested &lt;communicationType&gt; and &lt;messageType&gt; combination on the requested &lt;timeTransmitterId&gt; capture flow.</p> <p>The result returned is expressed in seconds to a maximum of 12 decimal places. If the requested &lt;communicationType&gt; and &lt;messageType&gt; is not present in the capture file for the specified &lt;timeTransmitterId&gt; "N/A" will be returned.</p>

**PFV PTP MessageIntervalArithmeticMeanPass <communicationType> <messageType>**

<b>Get</b>	
<b>Description</b>	Returns a Boolean result indicating the success or failure of a 'meanInterMessageInterval' rule for the specified communication type and message type.
<b>Command</b>	PFV PTP MessageIntervalArithmeticMeanPass <communicationType> <messageType>
<b>Parameters</b>	<p>&lt;communicationType&gt; PTP messages will be 'Unicast' or 'Multicast' determined by the value of the 'Unicast' flag in the common message header.</p> <p style="text-align: center;">UNICAST      MULTICAST</p> <p>&lt;messageType&gt; The PTP message type. This must be one of the following:</p> <p style="text-align: center;">SYNC              ANNOUNCE DELREQ            PDELREQ</p>
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV. An appropriate rules file must also be active.
<b>Result</b>	Returns the result of the relevant PFV analysis 'meanInterMessageInterval' rule. If TRUE, the calculated 'MessageIntervalArithmeticMean' value passed the rule associated with the <communicationType> and <messageType>; if FALSE then the rule check has failed. Note that if the specified <communicationType> and <messageType> has no rule associated with it the result returned will be FALSE.

**PFV PTP TwoTimeTransmitters MessageIntervalArithmeticMeanPass <timeTransmitterId>  
<communicationType> <messageType>**

<b>Get</b>	
<b>Description</b>	<p>This extended version of the MessageIntervalArithmeticMeanPass command is provided to query the analysis of Paragon-neo 'Two TimeTransmitters' captures where message captures from two disitinct timeTransmitter devices are present in a single capture file.</p> <p>Returns a Boolean result indicating the success or failure of a 'meanInterMessageInterval' rule for the specified timeTransmitter identifier, communication type and message type.</p>
<b>Command</b>	<p>PFV PTP TwoTimeTransmitters MessageIntervalArithmeticMeanPass &lt;timeTransmitterId&gt; &lt;communicationType&gt; &lt;messageType&gt;</p>
<b>Parameters</b>	<p>&lt;timeTransmitterId&gt; The PTP timeTransmitter identifier. This must be one of the following: TIMETRANSMITTER1 TIMETRANSMITTER2</p> <p>&lt;communicationType&gt; PTP messages will be 'Unicast' or 'Multicast' determined by the value of the 'Unicast' flag in the common message header. UNICAST MULTICAST</p> <p>&lt;messageType&gt; The PTP message type. This must be one of the following: SYNC ANNOUNCE DELREQ PDELREQ</p>
<b>Pre-requisites</b>	<p>A Paragon-neo 'Two TimeTransmitters' capture file must be loaded and have been analyzed by the PFV. An appropriate rules file must also be active.</p>
<b>Result</b>	<p>Returns the result of the relevant PFV analysis 'meanInterMessageInterval' rule. If TRUE, the calculated 'MessageIntervalArithmeticMean' value passed the rule associated with the &lt;timeTransmitterId&gt;, &lt;communicationType&gt; and &lt;messageType&gt;; if FALSE then the rule check has failed.</p> <p>Note that if the specified &lt;timeTransmitterId&gt;, &lt;communicationType&gt; and &lt;messageType&gt; has no rule associated with it the result returned will be FALSE.</p>

**PFV PTP MessageIntervalPassPercentage <communicationType> <messageType>**

<b>Get</b>	
<b>Description</b>	Returns the percentage of inter-message intervals measured for the specified communication type and message type that lie within +/- 30% of the corresponding log message interval. This metric is calculated in order to test against inter-message interval expectations as detailed in the 2019 revision of the IEEE 1588 standards.
<b>Command</b>	PFV PTP MessageIntervalPassPercentage <communicationType> <messageType>
<b>Parameters</b>	<p>&lt;communicationType&gt; PTP messages will be 'Unicast' or 'Multicast' determined by the value of the 'Unicast' flag in the common message header.</p> <p style="text-align: center;">UNICAST      MULTICAST</p> <p>&lt;messageType&gt; The PTP message type. This must be one of the following:</p> <p style="text-align: center;">SYNC              ANNOUNCE DELREQ</p>
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.
<b>Result</b>	Returns the pass percentage all inter-message intervals measured for the requested <communicationType> and <messageType> combination that lie within +/- 30% of the corresponding log message interval. If the requested <communicationType> and <messageType> is not present in the capture file "N/A" will be returned.

**PFV PTP TwoTimeTransmitters MessageIntervalPassPercentage <timeTransmitterId>  
<communicationType> <messageType>**

<b>Get</b>	
<b>Description</b>	<p>This extended version of the MessageIntervalPassPercentage command is provided to query the analysis of Paragon-neo 'Two TimeTransmitters' captures where message captures from two disitinct timeTransmitter devices are present in a single capture file.</p> <p>Returns the percentage of inter-message intervals measured for the specified communication type and message type that lie within +/- 30% of the corresponding log message interval. This metric is calculated in order to test against inter-message interval expectations as detailed in the 2019 revision of the IEEE 1588 standards.</p>
<b>Command</b>	PFV PTP TwoTimeTransmitters MessageIntervalPassPercentage <timeTransmitterId> <communicationType> <messageType>
<b>Parameters</b>	<p><i>&lt;timeTransmitterId&gt;</i> The PTP timeTransmitter identifier. This must be one of the following: TIMETRANSMITTER1 TIMETRANSMITTER2</p> <p><i>&lt;communicationType&gt;</i> PTP messages will be 'Unicast' or 'Multicast' determined by the value of the 'Unicast' flag in the common message header. UNICAST MULTICAST</p> <p><i>&lt;messageType&gt;</i> The PTP message type. This must be one of the following: SYNC ANNOUNCE DELREQ</p>
<b>Pre-requisites</b>	A Paragon-neo 'Two TimeTransmitters' capture file must be loaded and have been analyzed by the PFV.
<b>Result</b>	Returns the pass percentage all inter-message intervals measured for the requested <timeTransmitterId>, <communicationType> and <messageType> combination that lie within +/- 30% of the corresponding log message interval. If the requested <timeTransmitterId>, <communicationType> and <messageType> is not present in the capture file "N/A" will be returned.

**PFV PTP MessageIntervalPassPercentagePass <communicationType> <messageType>**

<b>Get</b>	
<b>Description</b>	Returns a Boolean result indicating the success or failure of a 'interMessageIntervalPassPercentage' rule for the specified communication type and message type.
<b>Command</b>	PFV PTP MessageIntervalPassPercentagePass <communicationType> <messageType>
<b>Parameters</b>	<p>&lt;communicationType&gt; PTP messages will be 'Unicast' or 'Multicast' determined by the value of the 'Unicast' flag in the common message header.</p> <p style="text-align: center;">UNICAST      MULTICAST</p> <p>&lt;messageType&gt; The PTP message type. This must be one of the following:</p> <p style="text-align: center;">SYNC              ANNOUNCE DELREQ</p>
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV. An appropriate rules file must also be active.
<b>Result</b>	<p>Returns the result of the relevant PFV analysis 'interMessageIntervalPassPercentage' rule. If TRUE, the calculated 'MessageIntervalPassPercentage' value passed the rule associated with the &lt;communicationType&gt; and &lt;messageType&gt;; if FALSE then the rule check has failed.</p> <p>Note that if the specified &lt;communicationType&gt; and &lt;messageType&gt; has no rule associated with it the result returned will be FALSE.</p>

**PFV PTP TwoTimeTransmitters MessageIntervalPassPercentagePass <timeTransmitterId>  
<communicationType> <messageType>**

<b>Get</b>	
<b>Description</b>	<p>This extended version of the MessageIntervalPassPercentagePass command is provided to query the analysis of Paragon-neo 'Two TimeTransmitters' captures where message captures from two disitinct timeTransmitter devices are present in a single capture file.</p> <p>Returns a Boolean result indicating the success or failure of a 'interMessageIntervalPassPercentage' rule for the specified timeTransmitter identifier, communication type and message type.</p>
<b>Command</b>	<p>PFV PTP TwoTimeTransmitters MessageIntervalPassPercentagePass &lt;timeTransmitterId&gt; &lt;communicationType&gt; &lt;messageType&gt;</p>
<b>Parameters</b>	<p>&lt;timeTransmitterId&gt; The PTP timeTransmitter identifier. This must be one of the following: TIMETRANSMITTER1 TIMETRANSMITTER2</p> <p>&lt;communicationType&gt; PTP messages will be 'Unicast' or 'Multicast' determined by the value of the 'Unicast' flag in the common message header. UNICAST MULTICAST</p> <p>&lt;messageType&gt; The PTP message type. This must be one of the following: SYNC ANNOUNCE DELREQ</p>
<b>Pre-requisites</b>	<p>A Paragon-neo 'Two TimeTransmitters' capture file must be loaded and have been analyzed by the PFV. An appropriate rules file must also be active.</p>
<b>Result</b>	<p>Returns the result of the relevant PFV analysis 'interMessageIntervalPassPercentage' rule. If TRUE, the calculated 'MessageIntervalPassPercentage' value passed the rule associated with the &lt;timeTransmitterId&gt;, &lt;communicationType&gt; and &lt;messageType&gt;; if FALSE then the rule check has failed.</p> <p>Note that if the specified &lt;timeTransmitterId&gt;, &lt;communicationType&gt; and &lt;messageType&gt; has no rule associated with it the result returned will be FALSE.</p>



**PFV PTP TwoTimeTransmitters MessageCount <timeTransmitterId> <captureDirection> <messageType>**

<b>Get</b>	
<b>Description</b>	This extended version of the MessageCount command is provided to query the analysis of Paragon-neo 'Two TimeTransmitters' captures where message captures from two disitinct timeTransmitter devices are present in a single capture file. Returns the message count for the specified message type, capture direction and timeTransmitter identifier.
<b>Command</b>	PFV PTP TwoTimeTransmitters MessageCount <timeTransmitterId> <captureDirection> <messageType>
<b>Parameters</b>	<p><i>&lt;timeTransmitterId&gt;</i> The PTP timeTransmitter identifier. This must be one of the following:  TIMETRANSMITTER1 TIMETRANSMITTER2</p> <p><i>&lt;captureDirection&gt;</i> The direction the message was captured at the measurement port. Tx is a transmitted message at the port, Rx is a received message at the port.  Tx Rx Any</p> <p><i>&lt;messageType&gt;</i> The PTP message type. This must be one of the following:  SYNC FOLLOWUP  DELREQ DELRESP  PDELREQ PDELRESP PDELRESPFUP  ANNOUNCE SIGNALING MANAGEMENT</p>
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.
<b>Result</b>	Returns the message count of the requested <messageType> in the requested <captureDirection> for the specified <timeTransmitterId>. Use <captureDirection> 'Any' to retrieve the total number of <messageType> messages captured. If the requested <messageType> is not present in the capture file 0 will be returned.

**PFV PTP MessageCountPass <captureDrection> <messageType>**

<b>Get</b>	
<b>Description</b>	Returns a Boolean result indicating the success or failure of a message count rule for the specified message type and capture direction.
<b>Command</b>	PFV PTP MessageCountPass <captureDirection> <messageType>
<b>Parameters</b>	<p>&lt;captureDirection&gt;    The direction the message was captured at the measurement port. Tx is a transmitted message at the port, Rx is a received message at the port.</p> <p style="padding-left: 100px;">Tx                      Rx                      Any</p> <p>&lt;messageType&gt;        The PTP message type. This must be one of the following:</p> <p style="padding-left: 100px;">SYNC                    FOLLOWUP</p> <p style="padding-left: 100px;">DELREQ                DELRESP</p> <p style="padding-left: 100px;">PDELREQ              PDELRESP      PDELRESPFUP</p> <p style="padding-left: 100px;">ANNOUNCE            SIGNALING      MANAGEMENT</p>
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.
<b>Result</b>	<p>Returns the result of the relevant PFV analysis 'messageCount' rule. If TRUE, the message count passed the rule associated with the &lt;messageType&gt; and &lt;captureDirection&gt;; if FALSE then the rule check has failed.</p> <p>Note that if the specified &lt;messageType&gt; and &lt;captureDirection&gt; has no rule associated with it the result returned will be FALSE.</p>

**PFV PTP TwoTimeTransmitters MessageCountPass <timeTransmitterId> <captureDrection>  
<messageType>**

<b>Get</b>	
<b>Description</b>	This extended version of the MessageCountPass command is provided to query the analysis of Paragon-neo 'Two TimeTransmitters' captures where message captures from two disitinct timeTransmitter devices are present in a single capture file.  Returns a Boolean result indicating the success or failure of a message count rule for the specified message type, capture direction and timeTransmitter identifier.
<b>Command</b>	PFV PTP TwoTimeTransmitters MessageCountPass <timeTransmitterId> <captureDirection> <messageType>
<b>Parameters</b>	<p>&lt;timeTransmitterId&gt; The PTP timeTransmitter identifier. This must be one of the following: TIMETRANSMITTER1 TIMETRANSMITTER2</p> <p>&lt;captureDirection&gt; The direction the message was captured at the measurement port. Tx is a transmitted message at the port, Rx is a received message at the port. Tx Rx Any</p> <p>&lt;messageType&gt; The PTP message type. This must be one of the following: SYNC FOLLOWUP DELREQ DELRESP PDELREQ PDELRESP PDELRESPFUP ANNOUNCE SIGNALING MANAGEMENT</p>
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.
<b>Result</b>	Returns the result of the relevant PFV analysis 'messageCount' rule. If TRUE the message count passed the rule associated with the <messageType>, <captureDirection> and <timeTransmitterId>; if FALSE then the rule check has failed.  Note that if the specified <messageType>, <captureDirection> and <timeTransmitterId> has no rule associated with it the result returned will be FALSE.

PFV PTP TwoMasters Transitions <masterId> <messageType> <fieldName>

Get	
<b>Description</b>	<p><b>"In support of the IEEE 1588:2022 standard amendment 2 although this command will continue to be supported we suggest you use the equivalent 'TimeTransmitter' command."</b></p> <p>This extended version of the PTP Transitions command is provided to query the analysis of Paragon-neo 'Two Masters' captures where message captures from two disitinct master devices are present in a single capture file.</p> <p>Returns a set of comma-separated values detailing how a field changed its value throughout the duration of the PTP capture. Each range has a set of delimiting packet indexes and timestamps associated with it.</p>
<b>Command</b>	PFV PTP TwoMasters Transitions <masterId> <messageType> <fieldName>
<b>Parameters</b>	<p>&lt;masterId&gt;      The PTP master identifier. This must be one of the following:                   MASTER1      MASTER2</p> <p>&lt;messageType&gt;    The PTP message type. This must be one of the following:                   ANNOUNCE    SYNC            FOLLOWUP                   DELREQ        DELRESP        MANAGEMENT                   PDELREQ      PDELRESP      PDELRESFPUP</p> <p>&lt;fieldname&gt;      The message field name. The field names are listed in "PTP Field Names" above.</p>
<b>Pre-requisites</b>	A Paragon-neo 'Two Masters' PTP capture file must be loaded and have been analyzed by the PFV.
<b>Result</b>	<p>Returns a set of comma-separated values. Example output:</p> <p><i>Value, PacketNumberFrom, PacketNumberTo, PacketTimestampFrom, PacketTimestampTo, 6, 2, 68, 0.02, 2.2</i></p> <p>Assuming that the example output above was returned by specifying 'MASTER1' as the &lt;masterId&gt; this would mean that the field value was 6 from 0.02s to 2.2s for 'MASTER1' and the first and last packet numbers in this range were 2 and 68 respectively. It does not mean that every packet between 2 and 68 (exclusive) originated at 'MASTER1'.</p>

PFV PTP TwoMasters TLVTransitions <masterId> <messageType> <tlvName> <tlvField>

Get	
<b>Description</b>	<p>"In support of the IEEE 1588:2022 standard amendment 2 although this command will continue to be supported we suggest you use the equivalent 'TimeTransmitter' command."</p> <p>This extended version of the PTP TLVTransitions command is provided to query the analysis of Paragon-neo 'Two Masters' or 'Two Subordinates' captures where message captures from two disitinct master or subordinate devices are present in a single capture file.</p> <p>Returns a set of comma-separated values detailing how a TLV field changed its value throughout the duration of the PTP capture. Each range has a set of delimiting packet indexes and timestamps associated with it. The range also indicates the zero-based index of the TLV in the message (which may contain multiple TLVs).</p>
<b>Command</b>	PFV PTP TwoMasters TLVTransitions <masterId> <messageType> <tlvName> <tlvField>
<b>Parameters</b>	<p>&lt;masterId&gt;      The PTP master identifier. This must be one of the following:                   MASTER1      MASTER2</p> <p>&lt;messageType&gt;    The PTP message type. This must be one of the following:                   ANNOUNCE    SYNC      FOLLOWUP                   DELREQ      DELRESP    MANAGEMENT                   PDELREQ     PDELRESP   PDELRESPFUP</p> <p>&lt;tlvName&gt;        The TLV being queried. The name must come from the "TLV Name" in the table in section "TLV Field Names".</p> <p>&lt;tlvField&gt;        The TLV field name. Valid field names can be found in the table in section "TLV Field Names".</p>
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.
<b>Result</b>	<p>Returns a set of comma-separated values. Example output:</p> <p>PacketNumberFrom,PacketNumberTo,PacketTimestampFrom,PacketTimestampTo, IndexOfTlv1,TlvValue1\n7,32,7E-05,0.0625,0,3,\n</p> <p>The above example would mean that the TLV was found in messages from the 7<sup>th</sup> to the 32<sup>nd</sup> packet in the capture, or from time 70μs to 62.5ms, and that:</p> <ul style="list-style-type: none"> <li>• The TLV was found at index 0 throughout this time, and</li> <li>• The value of the requested field was 3</li> </ul>

PFV PTP TwoMasters HSRTransitions <masterId> <messageType> <fieldName>

Get	
<b>Description</b>	<p><b>"In support of the IEEE 1588:2022 standard amendment 2 although this command will continue to be supported we suggest you use the equivalent 'TimeTransmitter' command."</b></p> <p>This extended version of the PTP HSRTransitions command is provided to query the analysis of Paragon-neo 'Two Masters' captures where message captures from two disitinct master devices are present in a single capture file.</p> <p>Returns a set of comma-separated values detailing how an HSR (High availability Seamless Redundnacy protocol) field changed its value throughout the duration of the PTP capture. Each range has a set of delimiting packet indexes and timestamps associated with it.</p>
<b>Command</b>	PFV PTP TwoMasters HSRTransitions <masterId> <messageType> <fieldName>
<b>Parameters</b>	<p>&lt;masterId&gt;      The PTP master identifier. This must be one of the following:                            MASTER1      MASTER2</p> <p>&lt;messageType&gt;    The PTP message type. This must be one of the following:                            ANNOUNCE    SYNC            FOLLOWUP                            DELREQ        DELRESP        MANAGEMENT                            PDELREQ      PDELRESP      PDELRESPFUP</p> <p>&lt;fieldname&gt;      The message field name. This must be one of the following: "                            NetId            LanId</p>
<b>Pre-requisites</b>	A Paragon-neo 'Two Masters' PTP capture file must be loaded and have been analyzed by the PFV.
<b>Result</b>	<p>Returns a set of comma-separated values. Example output:</p> <p><i>Value, PacketNumberFrom, PacketNumberTo, PacketTimestampFrom, PacketTimestampTo, 6, 2, 68, 0.02, 2.2</i></p> <p>Assuming that the example output above was returned by specifying 'MASTER1' as the &lt;masterId&gt; this would mean that the field value was 6 from 0.02s to 2.2s for 'MASTER1' and the first and last packet numbers in this range were 2 and 68 respectively. It does not mean that every packet between 2 and 68 (exclusive) originated at 'MASTER1'.</p>

PFV PTP TwoMasters PRPTransitions <masterId> <messageType> <fieldName>

Get	
<b>Description</b>	<p><b>"In support of the IEEE 1588:2022 standard amendment 2 although this command will continue to be supported we suggest you use the equivalent 'TimeTransmitter' command."</b></p> <p>This extended version of the PTP PRPTransitions command is provided to query the analysis of Paragon-neo 'Two Masters' captures where message captures from two disitinct master devices are present in a single capture file.</p> <p>Returns a set of comma-separated values detailing how a PRP (Parallel Redundnacy Protocol) field changed its value throughout the duration of the PTP capture. Each range has a set of delimiting packet indexes and timestamps associated with it.</p>
<b>Command</b>	PFV PTP TwoMasters PRPTransitions <masterId> <messageType> <fieldName>
<b>Parameters</b>	<p>&lt;masterId&gt;      The PTP master identifier. This must be one of the following:                            MASTER1      MASTER2</p> <p>&lt;messageType&gt;    The PTP message type. This must be one of the following:                            ANNOUNCE    SYNC            FOLLOWUP                            DELREQ        DELRESP        MANAGEMENT                            PDELREQ      PDELRESP      PDELRESFPUP</p> <p>&lt;fieldname&gt;      The message field name. This must be one of the following: "                            LanId</p>
<b>Pre-requisites</b>	A Paragon-neo 'Two Masters' PTP capture file must be loaded and have been analyzed by the PFV.
<b>Result</b>	<p>Returns a set of comma-separated values. Example output:</p> <p><i>Value, PacketNumberFrom, PacketNumberTo, PacketTimestampFrom, PacketTimestampTo, 6, 2, 68, 0.02, 2.2</i></p> <p>Assuming that the example output above was returned by specifying 'MASTER1' as the &lt;masterId&gt; this would mean that the field value was 6 from 0.02s to 2.2s for 'MASTER1' and the first and last packet numbers in this range were 2 and 68 respectively. It does not mean that every packet between 2 and 68 (exclusive) originated at 'MASTER1'.</p>

PFV PTP TwoMasters MessageRate <masterId> <messageType>

Get	
<b>Description</b>	<p>"In support of the IEEE 1588:2022 standard amendment 2 although this command will continue to be supported we suggest you use the equivalent 'TimeTransmitter' command."</p> <p>This extended version of the MessageRate command is provided to query the analysis of Paragon-neo 'Two Masters' captures where message captures from two disitinct master devices are present in a single capture file.</p> <p>Returns the message rate for the given message type and master device identifier.</p>
<b>Command</b>	PFV PTP TwoMasters MessageRate <masterId> <messageType>
<b>Parameters</b>	<p>&lt;masterId&gt;      The PTP master identifier. This must be one of the following:                            MASTER1      MASTER2</p> <p>&lt;messageType&gt;    The PTP message type. This must be one of the following:                            SYNC            FOLLOWUP                            DELREQ        DELRESP                            PDELREQ      PDELRESP    PDELRESFPUP                            ANNOUNCE    SIGNALING    MANAGEMENT</p>
<b>Pre-requisites</b>	A Paragon-neo 'Two Masters' capture file must be loaded and have been analyzed by the PFV.
<b>Result</b>	Returns the rate of the requested <messageType> in packets per second for the specified <masterId>. If the requested <messageType> is not present in the capture file for the specified <masterId> "N/A" will be returned.



PFV PTP TwoMasters DirectionalMessageRate <masterId> <captureDirection> <messageType>

Get	
<b>Description</b>	<p>"In support of the IEEE 1588:2022 standard amendment 2 although this command will continue to be supported we suggest you use the equivalent 'TimeTransmitter' command."</p> <p>This extended version of the MessageRate command is provided to query the analysis of Paragon-neo 'Two Masters' captures where message captures from two disitinct master devices are present in a single capture file.</p> <p>Returns the message rate for the given message type and master device identifier.</p>
<b>Command</b>	PFV PTP TwoMasters DirectionalMessageRate <masterId> <captureDirection> <messageType>
<b>Parameters</b>	<p>&lt;masterId&gt;            The PTP master identifier. This must be one of the following:                                   MASTER1        MASTER2</p> <p>&lt;captureDirection&gt;    The direction the message was captured at the selected port. Tx is a transmitted message at the port, Rx is a received message at the port.                                   Tx                Rx                Any</p> <p>&lt;messageType&gt;        The PTP message type. This must be one of the following:                                   SYNC            FOLLOWUP                                   DELREQ        DELRESP                                   PDELREQ      PDELRESP     PDELRESPFUP                                   ANNOUNCE     SIGNALING     MANAGEMENT</p>
<b>Pre-requisites</b>	A Paragon-neo 'Two Masters' capture file must be loaded and have been analyzed by the PFV.
<b>Result</b>	Returns the rate of the requested <messageType> in packets per second for the specified <masterId>. If the requested <messageType> is not present in the capture file for the specified <masterId> "N/A" will be returned.

**PFV PTP TwoMasters MessageRatePass <masterId> <messageType>**

<b>Get</b>	
<b>Description</b>	<p><b>"In support of the IEEE 1588:2022 standard amendment 2 although this command will continue to be supported we suggest you use the equivalent 'TimeTransmitter' command."</b></p> <p>This extended version of the MessageRatePass command is provided to query the analysis of Paragon-neo 'Two Masters' captures where message captures from two disitinct master devices are present in a single capture file.</p> <p>Returns a Boolean result indicating the success or failure of a message rate rule for the specified message type and master.</p>
<b>Command</b>	PFV PTP TwoMasters MessageRatePass <masterId> <messageType>
<b>Parameters</b>	<p><i>&lt;masterId&gt;</i>      The PTP master identifier. This must be one of the following:                                   MASTER1      MASTER2</p> <p><i>&lt;messageType&gt;</i>    The PTP message type. This must be one of the following:                                   SYNC            FOLLOWUP                                   DELREQ        DELRESP                                   PDELREQ      PDELRESP    PDELRESFPUP                                   ANNOUNCE    SIGNALING    MANAGEMENT</p>
<b>Pre-requisites</b>	A Paragon-neo 'Two Masters' capture file must be loaded and have been analyzed by the PFV.
<b>Result</b>	<p>Returns the result of the relevant PFV analysis 'averageMessageRate' rule. If TRUE, the message rate passed the rule associated with the &lt;messageType&gt; and &lt;masterId&gt;; if FALSE then the rules check has failed.</p> <p>Note that if the specified &lt;messageType&gt; and &lt;masterId&gt; has no rules associated with it the result returned will be FALSE.</p>

PFV PTP TwoMasters DirectionalMessageRatePass <masterId> <messageType>

Get	
<b>Description</b>	<p>"In support of the IEEE 1588:2022 standard amendment 2 although this command will continue to be supported we suggest you use the equivalent 'TimeTransmitter' command."</p> <p>This extended version of the DirectionalMessageRatePass command is provided to query the analysis of Paragon-neo 'Two Masters' captures where message captures from two disitinct master devices are present in a single capture file.</p> <p>Returns a Boolean result indicating the success or failure of a message rate rule for the specified message type, capture direction and master.</p>
<b>Command</b>	PFV PTP TwoMasters DirectionalMessageRatePass <masterId> <captureDirection> <messageType>
<b>Parameters</b>	<p>&lt;masterId&gt;            The PTP master identifier. This must be one of the following:                           MASTER1        MASTER2</p> <p>&lt;captureDirection&gt;    The direction the message was captured at the selected port. Tx is a transmitted message at the port, Rx is a received message at the port.                           Tx                Rx                Any</p> <p>&lt;messageType&gt;        The PTP message type. This must be one of the following:                           SYNC            FOLLOWUP                           DELREQ        DELRESP                           PDELREQ      PDELRESP      PDELRESPFUP                           ANNOUNCE    SIGNALING      MANAGEMENT</p>
<b>Pre-requisites</b>	A Paragon-neo 'Two Masters' capture file must be loaded and have been analyzed by the PFV.
<b>Result</b>	<p>Returns the result of the relevant PFV analysis 'averageMessageRate' rule. If TRUE, the message rate passed the rule associated with the &lt;messageType&gt;, &lt;captureDirection&gt; and &lt;masterId&gt;; if FALSE then the rules check has failed.</p> <p>Note that if the specified &lt;messageType&gt;, &lt;captureDirection&gt; and &lt;masterId&gt; has no rules associated with it the result returned will be FALSE.</p>

**PFV PTP TwoMasters MessageIntervalArithmeticMean <masterId> <communicationType>  
<messageType>**

<b>Get</b>							
<b>Description</b>	<p><b>"In support of the IEEE 1588:2022 standard amendment 2 although this command will continue to be supported we suggest you use the equivalent 'TimeTransmitter' command."</b></p> <p>This extended version of the MessageIntervalArithmeticMean command is provided to query the analysis of Paragon-neo 'Two Masters' captures where message captures from two disitinct master devices are present in a single capture file.</p> <p>Returns the arithmetic mean, in seconds to a picosecond resolution, of all inter-message intervals for the specified communication type and message type on the specified master device identifier.</p>						
<b>Command</b>	PFV PTP TwoMasters MessageIntervalArithmeticMean <masterId> <communicationType> <messageType>						
<b>Parameters</b>	<table style="width: 100%; border: none;"> <tr> <td style="width: 30%; vertical-align: top;"><i>&lt;masterId&gt;</i></td> <td>The PTP master identifier. This must be one of the following: MASTER1      MASTER2</td> </tr> <tr> <td style="vertical-align: top;"><i>&lt;communicationType&gt;</i></td> <td>PTP messages will be 'Unicast' or 'Multicast' determined by the value of the 'Unicast' flag in the common message header. UNICAST      MULTICAST</td> </tr> <tr> <td style="vertical-align: top;"><i>&lt;messageType&gt;</i></td> <td>The PTP message type. This must be one of the following: SYNC            ANNOUNCE DELREQ        PDELREQ</td> </tr> </table>	<i>&lt;masterId&gt;</i>	The PTP master identifier. This must be one of the following: MASTER1      MASTER2	<i>&lt;communicationType&gt;</i>	PTP messages will be 'Unicast' or 'Multicast' determined by the value of the 'Unicast' flag in the common message header. UNICAST      MULTICAST	<i>&lt;messageType&gt;</i>	The PTP message type. This must be one of the following: SYNC            ANNOUNCE DELREQ        PDELREQ
<i>&lt;masterId&gt;</i>	The PTP master identifier. This must be one of the following: MASTER1      MASTER2						
<i>&lt;communicationType&gt;</i>	PTP messages will be 'Unicast' or 'Multicast' determined by the value of the 'Unicast' flag in the common message header. UNICAST      MULTICAST						
<i>&lt;messageType&gt;</i>	The PTP message type. This must be one of the following: SYNC            ANNOUNCE DELREQ        PDELREQ						
<b>Pre-requisites</b>	A Paragon-neo 'Two Masters' capture file must be loaded and have been analyzed by the PFV.						
<b>Result</b>	<p>Returns the arithmetic mean of all inter-message intervals measured for the requested &lt;communicationType&gt; and &lt;messageType&gt; combination on the requested &lt;masterId&gt; capture flow.</p> <p>The result returned is expressed in seconds to a maximum of 12 decimal places. If the requested &lt;communicationType&gt; and &lt;messageType&gt; is not present in the capture file for the specified &lt;masterId&gt; "N/A" will be returned.</p>						

**PFV PTP TwoMasters MessageIntervalArithmeticMeanPass <masterId> <communicationType>  
<messageType>**

<b>Get</b>	
<b>Description</b>	<p><b>"In support of the IEEE 1588:2022 standard amendment 2 although this command will continue to be supported we suggest you use the equivalent 'TimeTransmitter' command."</b></p> <p>This extended version of the MessageIntervalArithmeticMeanPass command is provided to query the analysis of Paragon-neo 'Two Masters' captures where message captures from two disitinct master devices are present in a single capture file.</p> <p>Returns a Boolean result indicating the success or failure of a 'meanInterMessageInterval' rule for the specified master identifier, communication type and message type.</p>
<b>Command</b>	PFV PTP TwoMasters MessageIntervalArithmeticMeanPass <masterId> <communicationType> <messageType>
<b>Parameters</b>	<p><i>&lt;masterId&gt;</i>            The PTP master identifier. This must be one of the following:                                  MASTER1        MASTER2</p> <p><i>&lt;communicationType&gt;</i>   PTP messages will be 'Unicast' or 'Multicast' determined by the value of the 'Unicast' flag in the common message header.                                  UNICAST        MULTICAST</p> <p><i>&lt;messageType&gt;</i>            The PTP message type. This must be one of the following:                                  SYNC            ANNOUNCE                                  DELREQ        PDELREQ</p>
<b>Pre-requisites</b>	A Paragon-neo 'Two Masters' capture file must be loaded and have been analyzed by the PFV. An appropriate rules file must also be active.
<b>Result</b>	<p>Returns the result of the relevant PFV analysis 'meanInterMessageInterval' rule. If TRUE, the calculated 'MessageIntervalArithmeticMean' value passed the rule associated with the &lt;masterId&gt;, &lt;communicationType&gt; and &lt;messageType&gt;; if FALSE then the rule check has failed.</p> <p>Note that if the specified &lt;masterId&gt;, &lt;communicationType&gt; and &lt;messageType&gt; has no rule associated with it the result returned will be FALSE.</p>

**PFV PTP TwoMasters MessageIntervalPassPercentage <masterId> <communicationType>  
<messageType>**

<b>Get</b>							
<b>Description</b>	<p><b>"In support of the IEEE 1588:2022 standard amendment 2 although this command will continue to be supported we suggest you use the equivalent 'TimeTransmitter' command."</b></p> <p>This extended version of the MessageIntervalPassPercentage command is provided to query the analysis of Paragon-neo 'Two Masters' captures where message captures from two disitinct master devices are present in a single capture file. Returns the percentage of inter-message intervals measured for the specified communication type and message type that lie within +/- 30% of the corresponding log message interval.</p> <p>This metric is calculated in order to test against inter-message interval expectations as detailed in the 2019 revision of the IEEE 1588 standards.</p>						
<b>Command</b>	PFV PTP TwoMasters MessageIntervalPassPercentage <masterId> <communicationType> <messageType>						
<b>Parameters</b>	<table border="0"> <tr> <td style="vertical-align: top; padding-right: 10px;"><i>&lt;masterId&gt;</i></td> <td>The PTP master identifier. This must be one of the following: MASTER1      MASTER2</td> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;"><i>&lt;communicationType&gt;</i></td> <td>PTP messages will be 'Unicast' or 'Multicast' determined by the value of the 'Unicast' flag in the common message header. UNICAST      MULTICAST</td> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;"><i>&lt;messageType&gt;</i></td> <td>The PTP message type. This must be one of the following: SYNC              ANNOUNCE DELREQ</td> </tr> </table>	<i>&lt;masterId&gt;</i>	The PTP master identifier. This must be one of the following: MASTER1      MASTER2	<i>&lt;communicationType&gt;</i>	PTP messages will be 'Unicast' or 'Multicast' determined by the value of the 'Unicast' flag in the common message header. UNICAST      MULTICAST	<i>&lt;messageType&gt;</i>	The PTP message type. This must be one of the following: SYNC              ANNOUNCE DELREQ
<i>&lt;masterId&gt;</i>	The PTP master identifier. This must be one of the following: MASTER1      MASTER2						
<i>&lt;communicationType&gt;</i>	PTP messages will be 'Unicast' or 'Multicast' determined by the value of the 'Unicast' flag in the common message header. UNICAST      MULTICAST						
<i>&lt;messageType&gt;</i>	The PTP message type. This must be one of the following: SYNC              ANNOUNCE DELREQ						
<b>Pre-requisites</b>	A Paragon-neo 'Two Masters' capture file must be loaded and have been analyzed by the PFV.						
<b>Result</b>	Returns the pass percentage all inter-message intervals measured for the requested <masterId>, <communicationType> and <messageType> combination that lie within +/- 30% of the corresponding log message interval. If the requested <masterId>, <communicationType> and <messageType> is not present in the capture file "N/A" will be returned.						

**PFV PTP TwoMasters MessageIntervalPassPercentagePass <masterId> <communicationType> <messageType>**

<b>Get</b>	
<b>Description</b>	<p><b>"In support of the IEEE 1588:2022 standard amendment 2 although this command will continue to be supported we suggest you use the equivalent 'TimeTransmitter' command."</b></p> <p>This extended version of the MessageIntervalPassPercentagePass command is provided to query the analysis of Paragon-neo 'Two Masters' captures where message captures from two disitinct master devices are present in a single capture file.</p> <p>Returns a Boolean result indicating the success or failure of a 'interMessageIntervalPassPercentage' rule for the specified master identifier, communication type and message type.</p>
<b>Command</b>	<p>PFV PTP TwoMasters MessageIntervalPassPercentagePass &lt;masterId&gt; &lt;communicationType&gt; &lt;messageType&gt;</p>
<b>Parameters</b>	<p><i>&lt;masterId&gt;</i>                      The PTP master identifier. This must be one of the following:  <span style="margin-left: 100px;">MASTER1      MASTER2</span></p> <p><i>&lt;communicationType&gt;</i>      PTP messages will be 'Unicast' or 'Multicast' determined by the value of the 'Unicast' flag in the common message header.  <span style="margin-left: 100px;">UNICAST      MULTICAST</span></p> <p><i>&lt;messageType&gt;</i>                      The PTP message type. This must be one of the following:  <span style="margin-left: 100px;">SYNC              ANNOUNCE</span>  <span style="margin-left: 100px;">DELREQ</span></p>
<b>Pre-requisites</b>	<p>A Paragon-neo 'Two Masters' capture file must be loaded and have been analyzed by the PFV. An appropriate rules file must also be active.</p>
<b>Result</b>	<p>Returns the result of the relevant PFV analysis 'interMessageIntervalPassPercentage' rule. If TRUE, the calculated 'MessageIntervalPassPercentage' value passed the rule associated with the &lt;masterId&gt;, &lt;communicationType&gt; and &lt;messageType&gt;; if FALSE then the rule check has failed.</p> <p>Note that if the specified &lt;masterId&gt;, &lt;communicationType&gt; and &lt;messageType&gt; has no rule associated with it the result returned will be FALSE.</p>

PFV PTP TwoMasters MessageCount <masterId> <captureDrection> <messageType>

Get	
<b>Description</b>	<p><b>"In support of the IEEE 1588:2022 standard amendment 2 although this command will continue to be supported we suggest you use the equivalent 'TimeTransmitter' command."</b></p> <p>This extended version of the MessageCount command is provided to query the analysis of Paragon-neo 'Two Masters' captures where message captures from two disitinct master devices are present in a single capture file.</p> <p>Returns the message count for the specified message type, capture direction and master identifier.</p>
<b>Command</b>	PFV PTP TwoMasters MessageCount <masterId> <captureDirection> <messageType>
<b>Parameters</b>	<p>&lt;masterId&gt;            The PTP master identifier. This must be one of the following:                                   MASTER1        MASTER2</p> <p>&lt;captureDirection&gt;    The direction the message was captured at the measurement port. Tx is a transmitted message at the port, Rx is a received message at the port.                                   Tx                Rx                Any</p> <p>&lt;messageType&gt;        The PTP message type. This must be one of the following:                                   SYNC                FOLLOWUP                                   DELREQ            DELRESP                                   PDELREQ        PDELRESP        PDELRESPFUP                                   ANNOUNCE        SIGNALING        MANAGEMENT</p>
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.
<b>Result</b>	<p>Returns the message count of the requested &lt;messageType&gt; in the requested &lt;captureDirection&gt; for the specified &lt;masterId&gt;.</p> <p>Use &lt;captureDirection&gt; 'Any' to retrieve the total number of &lt;messageType&gt; messages captured.</p> <p>If the requested &lt;messageType&gt; is not present in the capture file 0 will be returned.</p>



**PFV PTP TwoMasters MessageCountPass <masterId> <captureDirection> <messageType>**

<b>Get</b>	
<b>Description</b>	<p><b>"In support of the IEEE 1588:2022 standard amendment 2 although this command will continue to be supported we suggest you use the equivalent 'TimeTransmitter' command."</b></p> <p>This extended version of the MessageCountPass command is provided to query the analysis of Paragon-neo 'Two Masters' captures where message captures from two disitinct master devices are present in a single capture file.</p> <p>Returns a Boolean result indicating the success or failure of a message count rule for the specified message type, capture direction and master identifier.</p>
<b>Command</b>	PFV PTP TwoMasters MessageCountPass <masterId> <captureDirection> <messageType>
<b>Parameters</b>	<p><i>&lt;masterId&gt;</i>            The PTP master identifier. This must be one of the following:                                           MASTER1      MASTER2</p> <p><i>&lt;captureDirection&gt;</i>    The direction the message was captured at the measurement port. Tx is a transmitted message at the port, Rx is a received message at the port.                                           Tx                Rx                Any</p> <p><i>&lt;messageType&gt;</i>        The PTP message type. This must be one of the following:                                           SYNC            FOLLOWUP                                           DELREQ        DELRESP                                           PDELREQ      PDELRESP      PDELRESPFUP                                           ANNOUNCE     SIGNALING      MANAGEMENT</p>
<b>Pre-requisites</b>	A file must be loaded and have been analyzed by the PFV.
<b>Result</b>	<p>Returns the result of the relevant PFV analysis 'messageCount' rule. If TRUE the message count passed the rule associated with the &lt;messageType&gt;, &lt;captureDirection&gt; and &lt;masterId&gt;; if FALSE then the rule check has failed.</p> <p>Note that if the specified &lt;messageType&gt;, &lt;captureDirection&gt; and &lt;masterId&gt; has no rule associated with it the result returned will be FALSE.</p>

---

**PFV Table Count (Deprecated)**

*This command is deprecated and should be replaced with PFV PTP Table Count.  
See PFV PTP Table Count for description.*

---

**PFV PTP Table Count**

Get	
<b>Description</b>	Returns the number of rows in the PTP data table.
<b>Command</b>	PFV PTP Table Count
<b>Result</b>	The number of rows in the table.

---

**PFV ToD Table Count**

Get	
<b>Description</b>	Returns the number of rows in the ToD data table.
<b>Command</b>	PFV ToD Table Count
<b>Result</b>	The number of rows in the table.

---

**PFV Table Data <rowOffset> <rowCount> (Deprecated)**

*This command is deprecated and should be replaced with PFV PTP Table Data.  
See PTP PFV Table Data for description.*

---

**PFV PTP Table Data <rowOffset> <rowCount>**

Get	
<b>Description</b>	Returns the data from specified PTP table rows.
<b>Command</b>	PFV PTP Table Data <rowOffset> <rowCount>
<b>Parameters</b>	<rowOffset> Integer. Zero-based index of the first row to be returned. This must be less than the total number of rows in the table. <rowCount> Integer. The number of rows to return.
<b>Result</b>	Returns the specified rows from the table preceded by the table header in csv format.

---

**PFV ToD Table Data <rowOffset> <rowCount>**

Get	
<b>Description</b>	Returns the data from specified ToD table rows.
<b>Command</b>	PFV ToD Table Data <rowOffset> <rowCount>
<b>Parameters</b>	<rowOffset> Integer. Zero-based index of the first row to be returned. This must be less than the total number of rows in the table. <rowCount> Integer. The number of rows to return.
<b>Result</b>	Returns the specified rows from the table preceded by the table header in csv format.

---

**PFV PTP Table Diagnostic <enable>**

Set	
<b>Description</b>	Toggles the visibility of diagnostic columns in the PTP data table.
<b>Command</b>	PFV PTP Table Diagnostic <enable>
<b>Parameters</b>	<enable> Boolean. TRUE will set the diagnostic columns in the data table visible, FALSE will hide them.
<b>Result</b>	A response type of 'Ok' if the command was successful. If diagnostic information is not available an appropriate message will be returned with an error response type.

---

**PFV PTP Table Export <file>**

Set	
<b>Description</b>	Exports the PTP data table contents to a comma-separated value (.csv) file.
<b>Command</b>	PFV PTP Table Export <file>
<b>Parameters</b>	<file> The full path and name of the exported file (.csv extension is optional, it will be added automatically if omitted).

---

**PFV PTP Statistics Export <file>**

Set	
<b>Description</b>	Exports the PTP statistics table contents to a comma-separated value (.csv) file.
<b>Command</b>	PFV PTP Statistics Export <file>
<b>Parameters</b>	<file> The full path and name of the exported file (.csv extension is optional, it will be added automatically if omitted).

---

**PFV AutoRulesFileSelect <enable>**

Set	
<b>Description</b>	Sets whether PFV will automatically select a Rules File based on PTP profile in the capture file, if available.
<b>Command</b>	PFV AutoRulesFileSelect <enable>
<b>Parameters</b>	<enable> TRUE or FALSE
Get	
<b>Description</b>	Whether PFV will automatically select a Rules File based on PTP profile in the capture file, if available.
<b>Command</b>	PFV AutoRulesFileSelect
<b>Result</b>	TRUE or FALSE corresponding to whether the AutoRulesFileSelect setting is enabled or disabled respectively.

---

**PFV Close**

Set	
<b>Description</b>	Closes the PFV application.
<b>Command</b>	PFV Close

## Command List Index

connect, 20  
disconnect, 20  
PFV AutoRulesFileSelect <enable>, 68  
PFV Close, 68  
PFV FieldErrorCount <fieldName> (Deprecated), 24  
PFV GenerateReport <filename>, 23  
PFV OpenFile, 21  
PFV PassRate (Deprecated), 24  
PFV PassResult (Deprecated), 23  
PFV PTP DirectionalMessageRate <captureDirection> <messageType>, 36  
PFV PTP DirectionalMessageRatePass <captureDirection> <messageType>, 39  
PFV PTP FieldErrorCount <fieldName>, 25  
PFV PTP HSRTransitions <messageType> <fieldName>, 27  
PFV PTP MessageCountPass <captureDrection> <messageType>, 50  
PFV PTP MessageIntervalArithmeticMean <communicationType> <messageType>, 41  
PFV PTP MessageIntervalArithmeticMeanPass <communicationType> <messageType>, 43  
PFV PTP MessageIntervalPassPercentage <communicationType> <messageType>, 45  
PFV PTP MessageIntervalPassPercentagePass <communicationType> <messageType>, 47  
PFV PTP MessageRate <messageType>, 35  
PFV PTP MessageRatePass <messageType>, 38  
PFV PTP PassRate, 24  
PFV PTP PassResult, 23  
PFV PTP PRPTransitions <messageType> <fieldName>, 27  
PFV PTP Rules, 21  
PFV PTP Table Count, 66  
PFV PTP Table Data <rowOffset> <rowCount>, 67  
PFV PTP Table Diagnostic <enable>, 67  
PFV PTP Table Export <file>, 68  
PFV PTP TLVFieldErrorCount <tlvName> <tlvField>, 25  
PFV PTP TLVTransitions <messageType> <tlvName> <tlvField>, 28  
PFV PTP Transitions <messageType> <fieldName>, 26  
PFV PTP TwoMasters DirectionalMessageRate <masterId> <captureDirection> <messageType>, 57  
PFV PTP TwoMasters DirectionalMessageRatePass <masterId> <messageType>, 59  
PFV PTP TwoMasters HSRTransitions <masterId> <messageType> <fieldName>, 54  
PFV PTP TwoMasters MessageCount <masterId> <captureDrection> <messageType>, 64  
PFV PTP TwoMasters MessageCountPass <masterId> <captureDrection> <messageType>, 65  
PFV PTP TwoMasters MessageIntervalArithmeticMean <masterId> <communicationType> <messageType>, 60  
PFV PTP TwoMasters MessageIntervalArithmeticMeanPass <masterId> <communicationType>  
<messageType>, 61  
PFV PTP TwoMasters MessageIntervalPassPercentage <masterId> <communicationType> <messageType>, 62  
PFV PTP TwoMasters MessageIntervalPassPercentagePass <masterId> <communicationType> <messageType>,  
63  
PFV PTP TwoMasters MessageRate <masterId> <messageType>, 56  
PFV PTP TwoMasters MessageRatePass <masterId> <messageType>, 58  
PFV PTP TwoMasters PRPTransitions <masterId> <messageType> <fieldName>, 55  
PFV PTP TwoMasters TLVTransitions <messageType> <tlvName> <tlvField>, 53  
PFV PTP TwoMasters Transitions <masterId> <messageType> <fieldName>, 52  
PFV PTP TwoTimeTransmitters DirectionalMessageRate <masterId> <captureDirection> <messageType>, 37  
PFV PTP TwoTimeTransmitters DirectionalMessageRatePass <masterId> <messageType>, 40  
PFV PTP TwoTimeTransmitters HSRTransitions <masterId> <messageType> <fieldName>, 33  
PFV PTP TwoTimeTransmitters MessageCount <masterId> <captureDrection> <messageType>, 49  
PFV PTP TwoTimeTransmitters MessageCountPass <masterId> <captureDrection> <messageType>, 51  
PFV PTP TwoTimeTransmitters MessageIntervalArithmeticMean <masterId> <communicationType>  
<messageType>, 42

PFV PTP TwoTimeTransmitters MessageIntervalArithmeticMeanPass <masterId> <communicationType>  
<messageType>, 44

PFV PTP TwoTimeTransmitters MessageIntervalPassPercentage <masterId> <communicationType>  
<messageType>, 46

PFV PTP TwoTimeTransmitters MessageIntervalPassPercentagePass <masterId> <communicationType>  
<messageType>, 48

PFV PTP TwoTimeTransmitters MessageRate <masterId> <messageType>, 35

PFV PTP TwoTimeTransmitters MessageRatePass <masterId> <messageType>, 38

PFV PTP TwoTimeTransmitters PRPTransitions <masterId> <messageType> <fieldName>, 34

PFV PTP TwoTimeTransmitters TLVTransitions <messageType> <tlvName> <tlvField>, 32

PFV PTP TwoTimeTransmitters Transitions <masterId> <messageType> <fieldName>, 31

PFV Rules (Deprecated), 21

PFV Show, 22

PFV Table Count (Deprecated), 66

PFV Table Data <rowOffset> <rowCount> (Deprecated), 66

PFV ToD FieldErrorCount <messageType> <fieldName>, 25

PFV ToD G8271Transitions <messageType> <fieldName>, 29

PFV ToD PassRate, 24

PFV ToD PassResult, 24

PFV ToD Rules, 22

PFV ToD Table Count, 66

PFV ToD Table Data <rowOffset> <rowCount>, 67

PFV ToD Transitions <fieldName>, 30

PFV Transitions <messageType> <fieldName>, 25



Calnex Solutions plc  
Oracle Campus  
Linlithgow EH49 7LR  
United Kingdom

t: +44 (0) 1506 671 416  
e: [info@calnexsol.com](mailto:info@calnexsol.com)

[calnexsol.com](https://calnexsol.com)

© Calnex Solutions, 2024

This information is subject to  
change without notice.

Apr 2024