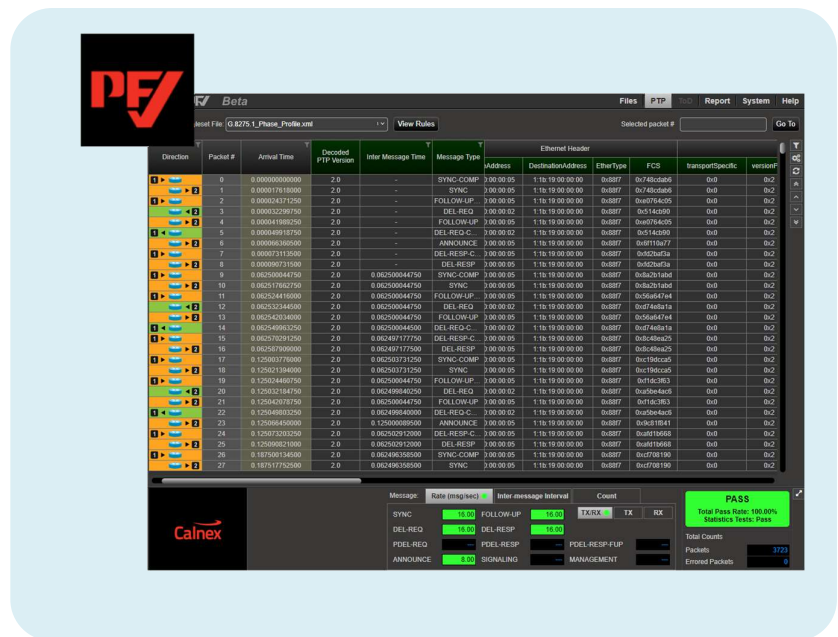


Calnex PFV – Packet Field Verifier



Message	Rate (msg/sec)	Inter message Interval	Count	TX/RX	TX	RX
SYNC	16.00	FOLLOW-UP	16.00	TX/RX	TX	RX
DEL-REQ	16.00	DEL-RESP	16.00			
PDEL-REQ		PDEL-RESP		PDEL-RESP-FUP		
ANNOUNCE	8.00	SIGNALING		MANAGEMENT		

PASS
Total Pass Rate: 100.00%
Statistics Type: Pass
Total Counts: 1773
Packets: 1773
Errored Packets: 0

GETTING STARTED GUIDE

Version 1.1

Table of Contents

1	Introduction	4
2	Using PFV.....	5
2.1	<i>Launching the Application.....</i>	5
	From Paragon-neo/Paragon-100G	5
	From Paragon-X.....	5
	Stand-alone operation	6
2.2	<i>Loading a File</i>	6
	Calnex Sentinel/Sentry Operation.....	6
2.3	<i>Rules Selection</i>	8
2.4	<i>The Main Display.....</i>	9
2.5	<i>Column Organization</i>	12
2.6	<i>Report Generation.....</i>	14
2.7	<i>Go To Packet</i>	15
2.8	<i>Show Filters.....</i>	16
2.9	<i>Loading a Licence File.....</i>	18
3	Customising the Rules File	19
3.1	<i>Accessing Paragon-neo/Paragon-100G Filesystem</i>	19
3.2	<i>File Locations.....</i>	20
3.3	<i>XML Syntax – Top-Level Structure.....</i>	21
3.4	<i>PTP Elements (testType = PTP).....</i>	22
	XML Syntax - ptpGeneral	22
	XML Syntax - ptpHeader	23
	XML Syntax – announceMessageBody.....	24
	XML Syntax – announceMessageCMCC5GTLV.....	24
	XML Syntax – managementMessageBody	25
	XML Syntax – ethernetHeader	25
	XML Syntax – ipv4Header	26
	XML Syntax – ipv6Header	26
	XML Syntax – tlvData.....	27
3.5	<i>ToD Elements (testType = ToD)</i>	28
	XML Syntax - ccsaTimeInformationMessage	28
	XML Syntax – ccsaTimeEventMessage	28
	XML Syntax – g8271TimeEventMessage	29
	XML Syntax – g8271TimeAnnounceMessage.....	29
	XML Syntax – g8271GnssMessage	30
3.6	<i>Common Elements</i>	31
	XML Syntax - field.....	31
	XML Syntax – tests	32
	XML Syntax – conditions	33
3.7	<i>Enumerated Values.....</i>	34
	Message Types	34
	Comparison Operators.....	34
	Logical Operators	34
	Automatic Tests	34
	PTP Header Fields.....	35
	Announce Message Body Fields.....	35

Announce Message TLV Fields	36
Management Message Body Fields	36
Ethernet Header Fields	36
IPv4 Header Fields.....	37
IPv6 Header Fields.....	37
TLV Data Fields	38
CCSA Time Information Message Fields.....	43
CCSA Time Event Message Fields.....	44
G.8271 Time Event Message Fields.....	45
G.8271 Time Announce Message Fields	45
G.8271 GNSS Status Message Fields	46
<i>3.8 Editing a Rules File and the XML Schema.....</i>	<i>47</i>

1 Introduction

The Calnex Packet Field Verifier (PFV) allows you to verify that the PTP and ToD (CCSA) messages being generated by the device-under-test (DUT) conform to applicable standards, or to a user-defined set of rules.

The PFV application is available as an option with Paragon-X, Paragon-neo and Paragon-100G. In addition, it is available as a stand-alone application which can also be used with the Calnex Sentinel/Sentry test instruments.

The PFV application takes as input a capture (.cpd file captured using Paragon-X, .CDF file captured using Paragon-neo, or a .pcap) and verifies the fields of captured messages against a selected set of rules, with clear pass/fail indications.

Calnex provides rules for the following standards: IEEE1588-2008, IEEE1588-2008 (Annex J), IEEE1588-2019 (Annex_I_peer-to-peer), IEEE1588-2019 (Annex_I_end-to-end), IEEE1588-2019, IEEE802.1AS (gPTP), IEEE802.1AS-rev (gPTP), IEEE-AES67(end-to-end), IEEE-AES67 (peer-to-peer), IEEE-AES67_SMPTE (end-to-end), IEEE-AES67_SMPTE (pee-to-peer), IEEE-C37.238-2011, IEEE-C37.238-2017, ITU-T G.8265.1, ITU-T G.8275.1, G.8275.2, IEC-61850-9-3, SMPTE-ST-2059 (peer-to-peer), SMPTE-ST-2059(end-to-end) and (for ToD) CCSA and ITU-T G.8271.

Rules are defined using xml. This provides an easy way for you to supply your own custom rules that can be applied in addition to the pre-defined rules provided by Calnex. All PTP header fields and key Announce message body fields can be verified. In addition, Paragon-neo capture file content allows the verification of Management message body fields. The xml definitions provide an extremely flexible mechanism for checking the contents of PTP and ToD message fields.

PFV highlights any messages that do not match the rules, provides an overall pass/fail result, and displays statistics relating to the messages in the capture.

The screenshot displays the PFV application interface. At the top, there are menu options: Files, PTP, ToD, Report, System, Help. Below the menu, the 'Ruleset File' is set to '1588-2019(Annex_I_end-to-end).xml'. A table of captured packets is shown with columns for Direction, Packet #, Arrival Time, Decoded PTP Version, Inter Message Time, Message Type (1920), SourceAddress, DestinationAddress (1920), EtherType, FCS, messageLength, and doc. The table contains 25 rows of data, with some rows highlighted in red to indicate failures. At the bottom, a summary statistics panel shows the following data:

Message	Rate (msg/sec)	Inter-message Interval	Count
SYNC	128.00		128.00
FOLLOW-UP			128.00
DEL-REQ	0.00		0.00
DEL-RESP			0.00
PDEL-REQ	32.00		32.00
PDEL-RESP			32.00
PDEL-RESP-FUP			32.00
ANNOUNCE	0.00		0.00
SIGNALING			---
MANAGEMENT			---

A red 'FAIL' banner is present in the bottom right corner, indicating a 'Total Pass Rate: 84.20%' and 'Statistics Tests: Failures'. Below the banner, the total counts are shown: Packets: 7038, Errored Packets: 1920.

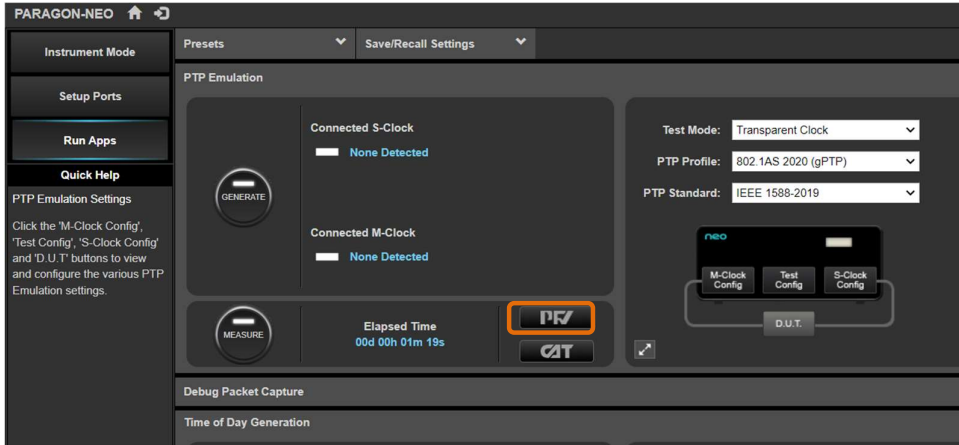
In addition, PFV can generate a report (in PDF or html format) that documents the overall pass/fail status along with details of any failures.

2 Using PFV

2.1 Launching the Application

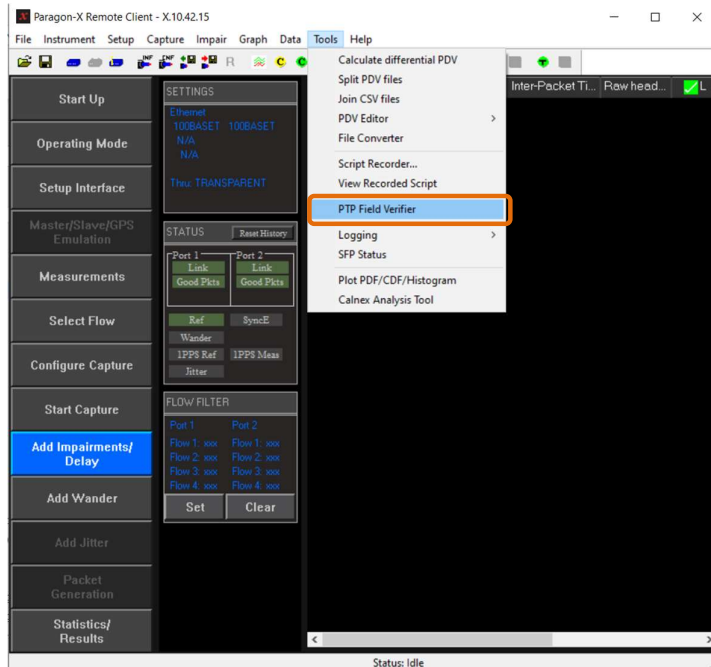
From Paragon-neo/Paragon-100G

PFV can be launched from Paragon-neo/Paragon-100G using the **PFV** button in the **PTP Emulation** application:



From Paragon-X

PFV can be launched from the Paragon-X application (**Tools > PTP Field Verifier**) as shown below:



Stand-alone operation

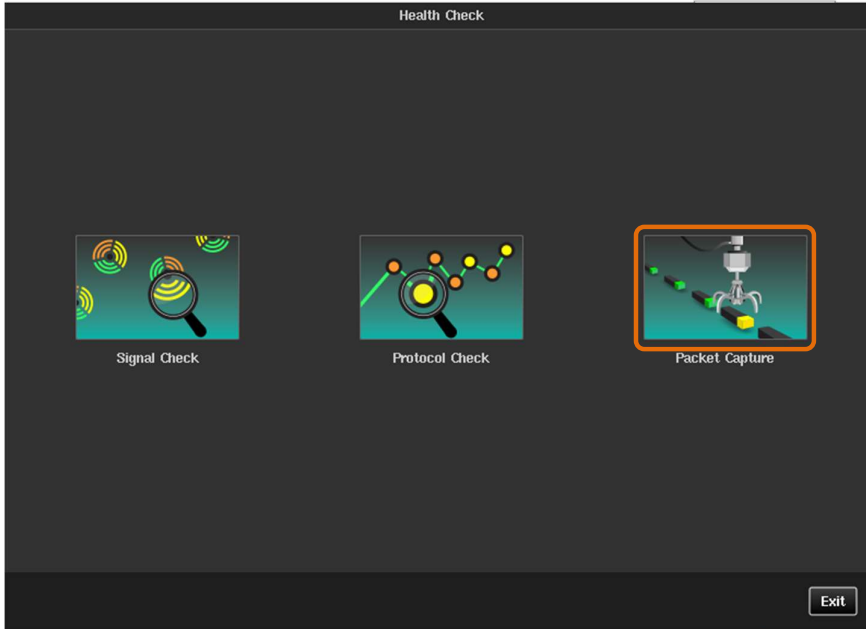
When installed either stand-alone or as part of the Paragon-X software bundle, the **Calnex PFV** can be launched from the **Start** menu:



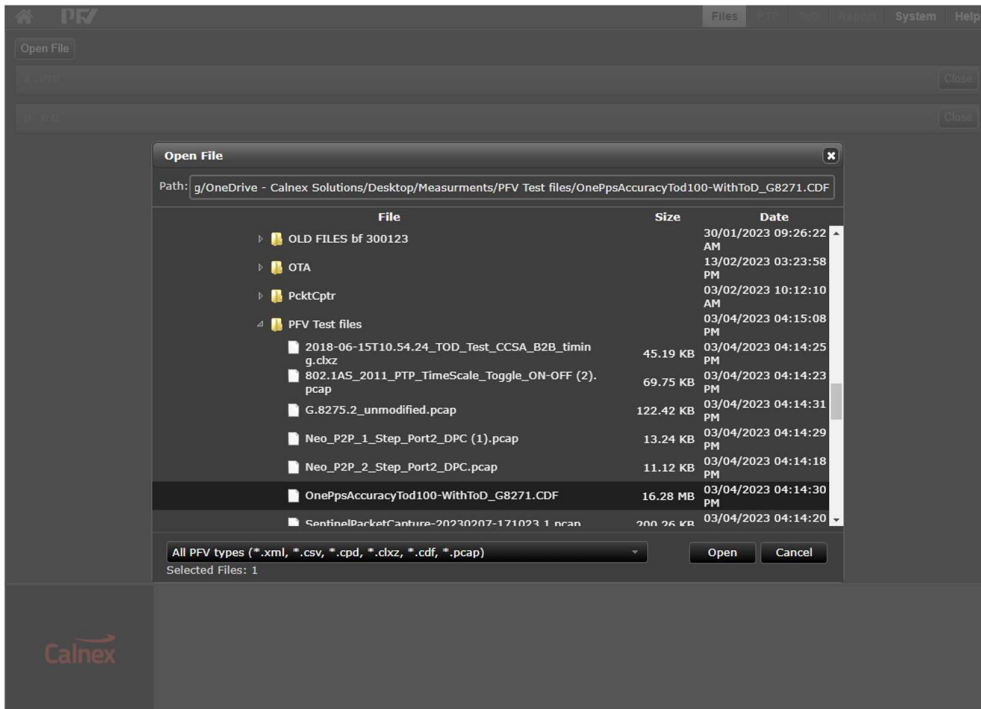
2.2 Loading a File

Calnex Sentinel/Sentry Operation

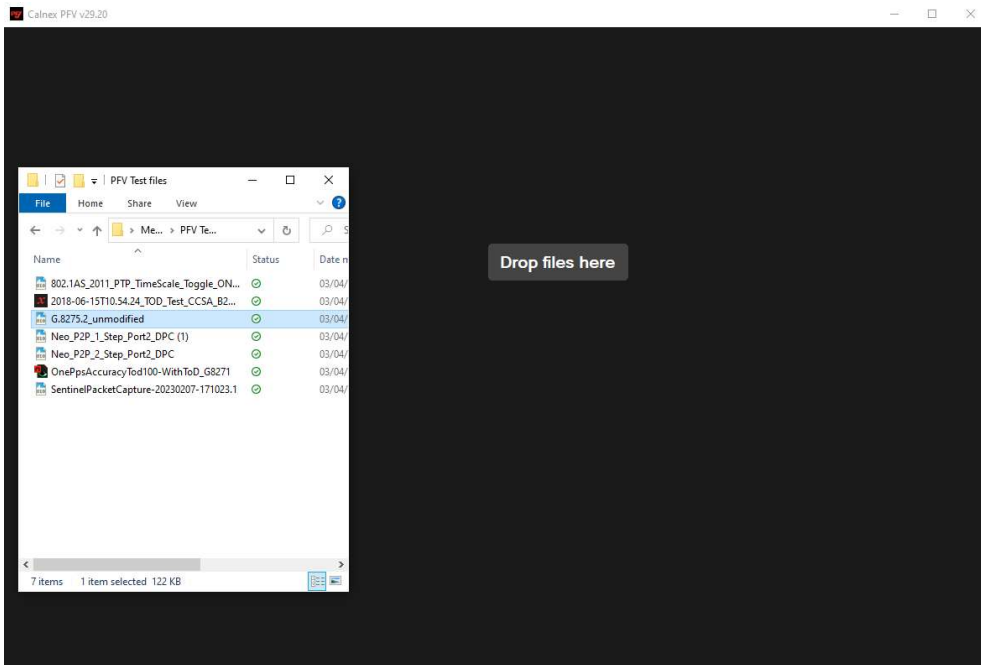
To analyze the Calnex Sentinel/Sentry PTP stream, capture the PTP packets (in .pcap format) using the 'Packet Capture' option under the 'Health Check' menu. The captured file(s) can be taken out of the Calnex Sentinel/Sentry using a USB drive or over FTP and then analyzed in the stand-alone **Calnex PFV** utility installed in a PC.



When launched from the Paragon-X application or Paragon-neo/Paragon-100G instrument web application, PFV will automatically load any current PTP or CCSA capture file. If there is no existing capture or PFV has been launched directly, then a capture file (.cpd, .CDF or .pcap) can be loaded using the **Files** menu:

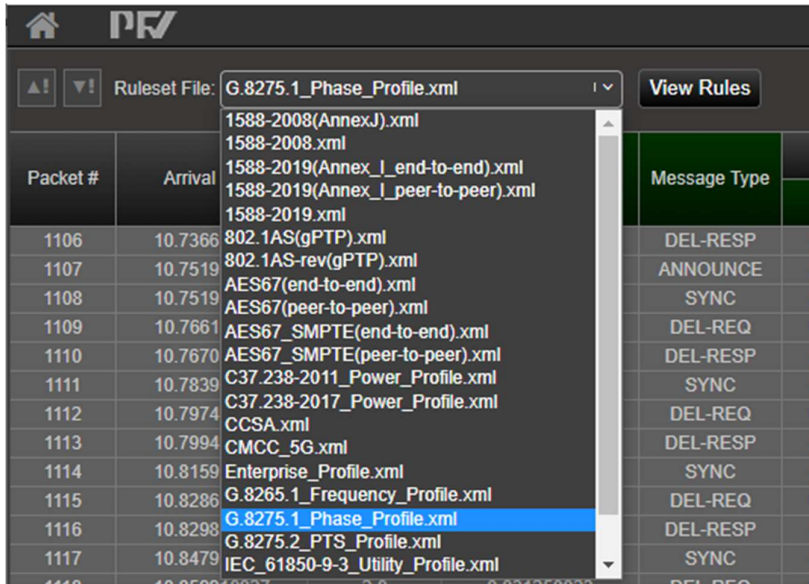


Alternatively, if using the stand-alone version, a file can be loaded using drag and drop:



2.3 Rules Selection

Rules files can be easily selected from the pull-down menu. Changing the rules file will cause the capture to be re-analyzed (this may take some time for large captures).



Note: Checking message contents against rules requires either a PC-based licence or that the capture file has been created from a Sentinel, Sentry, Paragon-X, Paragon-neo or Paragon-100G instrument with the appropriate options installed.

2.4 The Main Display

The main display of PFV is shown below:

Labels and annotations in the screenshot:

- Go to previous/next
- Rules selection
- View selected
- Go to packet
- View PTP or ToD
- Generate Report
- Go to packet arrival
- Show Filters
- Column Organizer
- Reset to default
- Scroll page/line
- Expand statistics summary panel

Message statistics summary panel

If any field within a message does not comply with the selected rules, then the affected cell is highlighted in red and the entire row is highlighted in grey. In addition, the column header will be highlighted in red if there are any errors in the associated field in any message. Details regarding the failure can be seen by hovering over the field:

Message statistics summary panel (FAIL):

- Total Pass Rate: 60.00%
- Statistics Tests: Fail

Arrival Time	Decoded PTP Version	Inter Message Time	Message Type	messageLength	domainNumber	minorSdoid	flagField0 (0)	flagField1	messageTypeSpecific
0.00000000	2.1	-	DEL-REQ	0x2c	0x0	0x0	0x24	0x0	0x0
0.00019000	2.1	-	SYNC	0x2c	0x0	0x0	0x6	0x0	0x0
0.00019000	2.1	-	ANNOUNCE	0x40	0x0	0x0	0x4	0x8	0x0
0.09212500	2.1	0.09212500	DEL-REQ	0x2c	0x0	0x0	0x24	0x0	0x0
0.09218700	2.1	0.09201800	SYNC	0x2c	0x0	0x0	0x6	0x0	0x0
0.09223900	2.1	0.09204000	ANNOUNCE	0x40	0x0	0x0	0x4	0x8	0x0
0.10333400	2.1	-	DEL-REQ	0x2c	0x0	0x0	0x24	0x0	0x0
0.10349600	2.1	-	SYNC	0x2c	0x0	0x0	0x6	0x0	0x0
0.10352600	2.1	-	ANNOUNCE	0x40	0x0	0x0	0x4	0x8	0x0
0.58483700	2.1	0.47271200	DEL-REQ	0x2c	0x0	0x0	0x24	0x0	0x0
0.58621100	2.1	0.47402400	SYNC	0x2c	0x0	0x0	0x6	0x0	0x0
0.58628100	2.1	0.47404200	ANNOUNCE	0x40	0x0	0x0	0x4	0x8	0x0
1.48779100	2.1	1.38445700	DEL-REQ	0x2c	0x0	0x0	0x24	0x0	0x0
1.48786700	2.1	1.38437100	SYNC	0x2c	0x0	0x0	0x6	0x0	0x0
1.48790700	2.1	1.38438100	ANNOUNCE	0x40	0x0	0x0	0x4	0x8	0x0
2.28012000	2.1	1.85517800	DEL-REQ	0x2c	0x0	0x0	0x24	0x0	0x0
2.28124800	2.1	1.50347300	DEL-REQ	0x2c	0x0	0x0	0x24	0x0	0x0
2.29112200	2.1	1.50345500	SYNC	0x2c	0x0	0x0	0x6	0x0	0x0
2.29113800	2.1	1.50347300	ANNOUNCE	0x40	0x0	0x0	0x4	0x8	0x0
4.65826200	2.1	1.66699800	DEL-REQ	0x2c	0x0	0x0	0x24	0x0	0x0

Tooltip details for 'ReservedOct08r3: False':

- AlternateMasterFlag: False
- TwoStepFlag: False
- UnicastFlag: True
- ReservedOct08r3: False
- ReservedOct08r4: False
- ProfileSpec1: True (ERROR)
- ProfileSpec2: False
- ReservedOct08r7: False

In the Message statistics summary panel, average message rates for each message type in the capture are displayed. These rates are compared with the rates defined in the rules file to produce a Pass/fail result.

Additional statistics-based tests have been added in support of 1588 2019 testing and so a lot of information is displayed within the statistics summary panel. Color highlighting has been employed to indicate which category of statistic tests are relevant to the applied rules file.

Clicking on the 'Expand statistics summary panel' icon will show all message statistics in a single table view:

The screenshot shows the PTP software interface with a detailed table of message statistics. The table includes columns for Direction, Packet #, Arrival Time, Decoded FTP Version, Inter Message Time, Message Type, SourceAddress, DestinationAddress, Ether Type, FCS, and transportS. Below the table is a summary panel with the following data:

Message Type	TXRX	TX	RX	Inter-message Interval - Arithmetic Mean	Inter-message bits	Overall
SYNC	18.00	---	18.00	0.062500024002	100.0	PASS
FOLLOW-UP	---	---	---	---	---	Total Pass Rate: 100.00%
DEL-REQ	18.00	18.00	---	0.062500024004	---	Statistics Tests: Pass
DEL-RESP	18.00	---	18.00	---	---	
PDEL-REQ	---	---	---	---	---	
PDEL-RESP	---	---	---	---	---	
PDEL-RESP-FUP	---	---	---	---	---	
ANNOUNCE	8.00	---	8.00	0.125000047998	100.0	Total Packets: 2068
SIGNALING	---	---	---	---	---	Errored Packets: 0
MANAGEMENT	---	---	---	---	---	

Clicking on the collapse icon will return to the summary view.

The statistics panel shows the total number of packets in the capture and the number of errored packets.

The percentage number of passing packets, the collective result of all applied statistics tests, and an overall pass/fail is clearly indicated.

The main display will show different columns depending on the capture type loaded. The examples below show Paragon-X PTP captures. A CCSA Time Of Day (1PPS) capture from Paragon-X shows the CCSA message types and corresponding fields:

The screenshot shows the PTP software interface with a detailed table of CCSA Information Message statistics. The table includes columns for Sample #, Length, UTC Time, Time of Week, Reserved0, Week, Leap, PPS Status (Raw), PPS Status, TAcc, Reserved1, Reserved2, Reserved3, and FCS. Below the table is a summary panel with the following data:

Message Type	Length	UTC Time	Time of Week	Reserved0	Week	Leap	PPS Status (Raw)	PPS Status	TAcc	Reserved1	Reserved2	Reserved3	FCS
788	0x10	2018-06-15 10:07:43	468480	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x0
789	0x10	2018-06-15 10:07:44	468481	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x1d
800	0x10	2018-06-15 10:07:45	468482	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x8d
801	0x10	2018-06-15 10:07:45	468483	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x2a
802	0x10	2018-06-15 10:07:47	468484	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x74
803	0x10	2018-06-15 10:07:48	468485	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x1d
804	0x10	2018-06-15 10:07:49	468486	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x23
805	0x10	2018-06-15 10:07:50	468487	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x84
806	0x10	2018-06-15 10:07:51	468488	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x9f
807	0x10	2018-06-15 10:07:52	468489	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x38
808	0x10	2018-06-15 10:07:53	468490	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0xc3
809	0x10	2018-06-15 10:07:54	468491	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x8f
810	0x10	2018-06-15 10:07:55	468492	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x31
811	0x10	2018-06-15 10:07:56	468493	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x96
812	0x10	2018-06-15 10:07:57	468494	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x66
813	0x10	2018-06-15 10:07:58	468495	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0xc1
814	0x10	2018-06-15 10:07:59	468496	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x59
815	0x10	2018-06-15 10:08:00	468497	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x17
816	0x10	2018-06-15 10:08:01	468498	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x7
817	0x10	2018-06-15 10:08:02	468499	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0xa0
818	0x10	2018-06-15 10:08:03	468500	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x6e
819	0x10	2018-06-15 10:08:04	468501	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x59
820	0x10	2018-06-15 10:08:05	468502	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0xa9
821	0x10	2018-06-15 10:08:06	468503	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x9e
822	0x10	2018-06-15 10:08:07	468504	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x15
823	0x10	2018-06-15 10:08:08	468505	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0xb2
824	0x10	2018-06-15 10:08:09	468506	0x0	2005	17	0x0	Normal	0	0x0	0x0	0x0	0x42

Summary Panel:

- PASS**
- Total Pass Rate: 100.00%
- Total Counts:
 - Packets: 836
 - Errored Packets: 0

Note when analyzing a '.pcap' file containing PTP data, PFV fully decodes the packet data similar to how Calnex '.cpd' and '.CDF' format files are decoded. However packet capture information is not available from '.pcap' files.

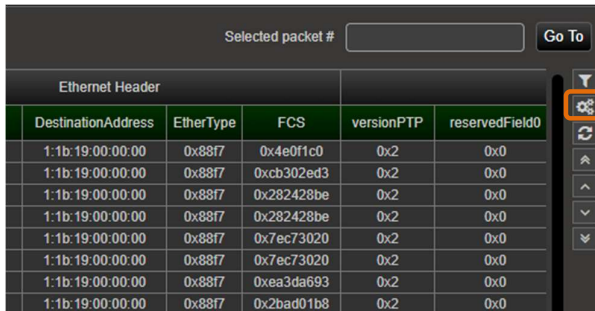
Packet #	Arrival Time	Decoded PTP Version	Inter Message Time	Message Type	PortIdentity	sequenceld	controlField (6401)	logMessageInterval
7979	99.718750000	2.0	0.062500000	DEL-RESP	000000010001	1695	0x0	-4
7980	99.750000000	2.0	0.062500000	ANNOUNCE	000000010001	1596	0x0	-4
7981	99.750000000	2.0	0.062500000	SYNC	000000010001	1696	0x0	-4
7982	99.750000000	2.0	0.062500000	FOLLOW-UP	000000010001	1696	0x0	-4
7983	99.781250000	2.0	0.062500000	DEL-REQ	000000020001	1696	0x0	127
7984	99.781250000	2.0	0.062500000	DEL-RESP	000000010001	1696	0x0	-4
7985	99.812500000	2.0	0.062500000	ANNOUNCE	000000010001	1597	0x0	-4
7986	99.812500000	2.0	0.062500000	SYNC	000000010001	1697	0x0	-4
7987	99.812500000	2.0	0.062500000	FOLLOW-UP	000000010001	1697	0x0	-4
7988	99.843750000	2.0	0.062500000	DEL-REQ	000000020001	1697	0x0	127
7989	99.843750000	2.0	0.062500000	DEL-RESP	000000010001	1697	0x0	-4
7990	99.875000000	2.0	0.062500000	ANNOUNCE	000000010001	1598	0x0	-4
7991	99.875000000	2.0	0.062500000	SYNC	000000010001	1698	0x0	-4
7992	99.875000000	2.0	0.062500000	FOLLOW-UP	000000010001	1698	0x0	-4

Capture port and direction information is available in both '.cpd' and '.CDF' formats.

Direction	Packet #	Arrival Time	Decoded PTP Version	Inter Message Time	Message Type	PortIdentity	sequenceld	controlField (6401)	logMessageInterval
→	7979	99.718750000	2.0	0.062500000	DEL-RESP	000000010001	1695	0x0	-4
←	7980	99.750000000	2.0	0.062500000	ANNOUNCE	000000010001	1596	0x0	-4
→	7981	99.750000000	2.0	0.062500000	SYNC	000000010001	1696	0x0	-4
→	7982	99.750000000	2.0	0.062500000	FOLLOW-UP	000000010001	1696	0x0	-4
←	7983	99.781250000	2.0	0.062500000	DEL-REQ	000000020001	1696	0x0	127
→	7984	99.781250000	2.0	0.062500000	DEL-RESP	000000010001	1696	0x0	-4
←	7985	99.812500000	2.0	0.062500000	ANNOUNCE	000000010001	1597	0x0	-4
→	7986	99.812500000	2.0	0.062500000	SYNC	000000010001	1697	0x0	-4
→	7987	99.812500000	2.0	0.062500000	FOLLOW-UP	000000010001	1697	0x0	-4
←	7988	99.843750000	2.0	0.062500000	DEL-REQ	000000020001	1697	0x0	127
→	7989	99.843750000	2.0	0.062500000	DEL-RESP	000000010001	1697	0x0	-4
←	7990	99.875000000	2.0	0.062500000	ANNOUNCE	000000010001	1598	0x0	-4
→	7991	99.875000000	2.0	0.062500000	SYNC	000000010001	1698	0x0	-4

2.5 Column Organization

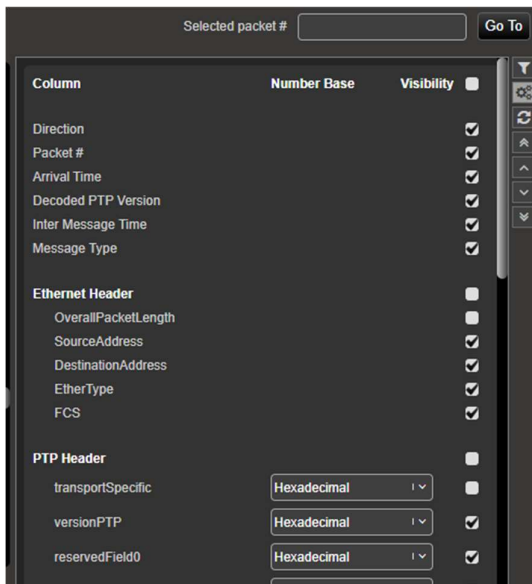
By default, PFV displays columns for fields that are tested by the selected rules file. Columns can be added or removed using the Column Organizer. This is also used to change the format of the column value (for example, to show decimal rather than hexadecimal values). The Column Organizer is accessed from the right-hand button menu:



The screenshot shows a table with the following data:

DestinationAddress	EtherType	FCS	versionPTP	reservedField0
1:1b:19:00:00:00	0x88f7	0x4e0f1c0	0x2	0x0
1:1b:19:00:00:00	0x88f7	0xcb302ed3	0x2	0x0
1:1b:19:00:00:00	0x88f7	0x282428be	0x2	0x0
1:1b:19:00:00:00	0x88f7	0x282428be	0x2	0x0
1:1b:19:00:00:00	0x88f7	0x7ec73020	0x2	0x0
1:1b:19:00:00:00	0x88f7	0x7ec73020	0x2	0x0
1:1b:19:00:00:00	0x88f7	0xea3da693	0x2	0x0
1:1b:19:00:00:00	0x88f7	0x2bad01b8	0x2	0x0

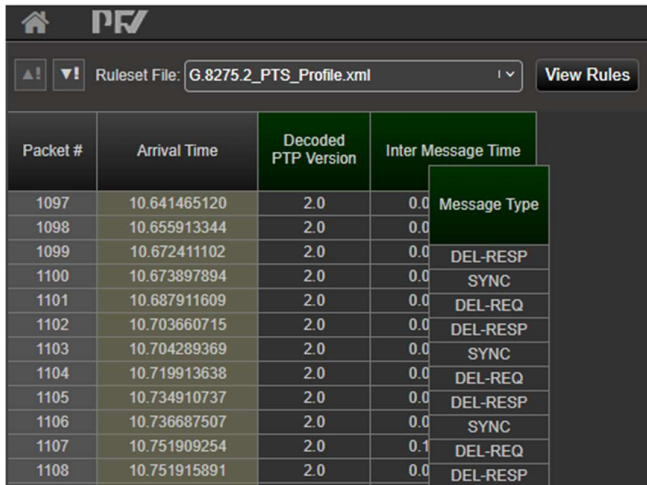
Only the columns that are available in the capture are shown:



The screenshot shows the Column Organizer interface with the following columns and settings:

Column	Number Base	Visibility
Direction		<input checked="" type="checkbox"/>
Packet #		<input checked="" type="checkbox"/>
Arrival Time		<input checked="" type="checkbox"/>
Decoded PTP Version		<input checked="" type="checkbox"/>
Inter Message Time		<input checked="" type="checkbox"/>
Message Type		<input checked="" type="checkbox"/>
Ethernet Header		<input type="checkbox"/>
OverallPacketLength		<input type="checkbox"/>
SourceAddress		<input checked="" type="checkbox"/>
DestinationAddress		<input checked="" type="checkbox"/>
EtherType		<input checked="" type="checkbox"/>
FCS		<input checked="" type="checkbox"/>
PTP Header		<input type="checkbox"/>
transportSpecific	Hexadecimal	<input type="checkbox"/>
versionPTP	Hexadecimal	<input checked="" type="checkbox"/>
reservedField0	Hexadecimal	<input checked="" type="checkbox"/>

Columns can be resized by dragging the right-hand edge of the column. They can also be re-ordered using drag-and-drop on the column headers.



Packet #	Arrival Time	Decoded PTP Version	Inter Message Time	Message Type
1097	10.641465120	2.0	0.0	
1098	10.655913344	2.0	0.0	
1099	10.672411102	2.0	0.0	DEL-RESP
1100	10.673897894	2.0	0.0	SYNC
1101	10.687911609	2.0	0.0	DEL-REQ
1102	10.703660715	2.0	0.0	DEL-RESP
1103	10.704289369	2.0	0.0	SYNC
1104	10.719913638	2.0	0.0	DEL-REQ
1105	10.734910737	2.0	0.0	DEL-RESP
1106	10.736687507	2.0	0.0	SYNC
1107	10.751909254	2.0	0.1	DEL-REQ
1108	10.751915891	2.0	0.0	DEL-RESP

2.6 Report Generation

To generate a report, select **Report** from the top-right menu bar:



The **Report** page allows several report fields to be manually entered prior to generating the report. This page also previews the contents of the automatically generated contents of the report. To create the report, click the **Save Report** button.

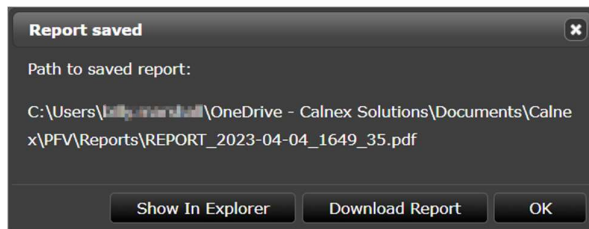
A screenshot of the 'Report' page in the software. The page has a dark theme. At the top, there is a 'Report File Name' field with the value 'REPORT_2023-04-04_1005_39', a '.pdf (A4)' dropdown, and a 'Save Report' button. Below this is a 'General Information' section with a 'Company Logo' preview area showing the PFV logo and a 'Clear Load Image' button. The main form contains several input fields: 'Report Title', 'Report Description', 'Company', 'User Name', 'Network Operator', 'Test Location', 'Report Date' (pre-filled with '2023-04-04 09:55:42 AM'), and a 'Notes' text area. Below this is a 'PTP Test Information' section with fields for 'Beginning of PTP Test' (pre-filled with '2021-07-22 09:57:29 PM'), 'End of PTP Test' (pre-filled with '2021-07-22 09:57:39 PM'), 'PTP Instrument Type' (pre-filled with 'Data Source Unknown'), and 'PTP Instrument Serial Number'. A 'Save Report' button is at the bottom right of the form.

A logo can be selected. This will be printed on each page of the report.

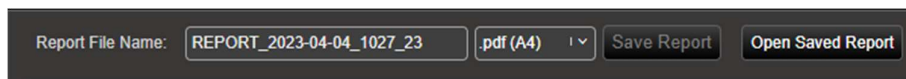
Custom information can be entered into a number of report fields.

You can put notes here.

Some report fields will be auto-populated if that information is available in the capture file.



The report is automatically saved in the PFV Reports folder. This folder can be opened by clicking **Show in Explorer** or can be saved to a different location by clicking **Download Report**. Once the report has been created, it can then be opened by clicking **Open Save Report**.



The contents of the rules file being used is included in the report.

2.7 Go To Packet

PFV allows rapid navigation to records-of-interest by use of the 'Selected packet #' and 'Arrival Time' 'Go To' feature.

With reference to the relevant 'Packet #' or 'Arrival Time' columns in the analysis table you can enter an approximate value of interest and press 'Go To'. PFV will select the record closest to the value entered. For example:

Direction	Packet #	Arrival Time	Decoded PTP Version	Inter Message Time	Message Type (1280)	sequenceId	controlField (1280)	logMessageInterval (640)	PTP Body F
P	3807	19.926125000000	2.1	0.007812500000	SYNC	2587	0x0	-7	origTstamp= 1970 1 00 1
P	3808	19.935837500000	2.1	0.007812500000	SYNC	2588	0x0	-7	origTstamp= 1970 1 00 1
P	3809	19.943750000000	2.1	0.007812500000	SYNC	2589	0x0	-7	origTstamp= 1970 1 00 1
P	3810	19.943750000000	2.1	0.031250000000	PDEL-REQ	835	0x0	-5	
P	3811	19.943750023600	2.1	0.031248590000	PDEL-RESP	835	0x0	-5	reqRcptTstamp= 1970 1 00 00 20 843750236
P	3812	19.951562500000	2.1	0.007812500000	SYNC	2590	0x0	-7	origTstamp= 1970 1 00 1
P	3813	19.959375000000	2.1	0.007812500000	SYNC	2591	0x0	-7	origTstamp= 1970 1 00 1
P	3814	19.967187500000	2.1	0.007812500000	SYNC	2592	0x0	-7	origTstamp= 1970 1 00 1
P	3815	19.975000000000	2.1	0.007812500000	SYNC	2593	0x0	-7	origTstamp= 1970 1 00 1
P	3816	19.975000000000	2.1	0.031250000000	PDEL-REQ	836	0x0	-5	
P	3817	19.975001090750	2.1	0.031250584750	PDEL-RESP	836	0x0	-5	reqRcptTstamp= 1970 1 00 00 20 875001090
P	3818	19.982812500000	2.1	0.007812500000	SYNC	2594	0x0	-7	origTstamp= 1970 1 00 1
P	3819	19.989625000000	2.1	0.007812500000	SYNC	2595	0x0	-7	origTstamp= 1970 1 00 1
P	3820	19.998437500000	2.1	0.007812500000	SYNC	2596	0x0	-7	origTstamp= 1970 1 00 1
P	3821	19.998250000000	2.1	0.007812500000	SYNC	2597	0x0	-7	origTstamp= 1970 1 00 1
P	3822	19.998250000000	2.1	0.031250000000	PDEL-REQ	837	0x0	-5	
P	3823	19.998252481750	2.1	0.031251391000	PDEL-RESP	837	0x0	-5	reqRcptTstamp= 1970 1 00 00 20 998252481
P	3824	19.914062500000	2.1	0.007812500000	SYNC	2598	0x0	-7	origTstamp= 1970 1 00 1
P	3825	19.921875000000	2.1	0.007812500000	SYNC	2599	0x0	-7	origTstamp= 1970 1 00 1
P	3826	19.929687500000	2.1	0.007812500000	SYNC	2600	0x0	-7	origTstamp= 1970 1 00 1
P	3827	19.937500000000	2.1	0.007812500000	SYNC	2601	0x0	-7	origTstamp= 1970 1 00 1
P	3828	19.937500000000	2.1	0.031250000000	PDEL-REQ	838	0x0	-5	
P	3829	19.937500010750	2.1	0.031247529000	PDEL-RESP	838	0x0	-5	reqRcptTstamp= 1970 1 00 00 20 937500010
P	3830	19.945312500000	2.1	0.007812500000	SYNC	2602	0x0	-7	origTstamp= 1970 1 00 1
P	3831	19.953125000000	2.1	0.007812500000	SYNC	2603	0x0	-7	origTstamp= 1970 1 00 1
P	3832	19.960937500000	2.1	0.007812500000	SYNC	2604	0x0	-7	origTstamp= 1970 1 00 1
P	3833	19.968750000000	2.1	0.007812500000	SYNC	2605	0x0	-7	origTstamp= 1970 1 00 1
P	3834	19.968750000000	2.1	0.031250000000	PDEL-REQ	839	0x0	-5	
P	3835	19.968750445250	2.1	0.031250434500	PDEL-RESP	839	0x0	-5	reqRcptTstamp= 1970 1 00 00 20 968750445
P	3836	19.976562500000	2.1	0.007812500000	SYNC	2606	0x0	-7	origTstamp= 1970 1 00 1
P	3837	19.984375000000	2.1	0.007812500000	SYNC	2607	0x0	-7	origTstamp= 1970 1 00 1
P	3838	19.992187500000	2.1	0.007812500000	SYNC	2608	0x0	-7	origTstamp= 1970 1 00 1

Pressing the 'Arrival Time' 'Go To' button above will center the selected record closest to 12 (exactly 12 in this case) in the display:

Direction	Packet #	Arrival Time	Decoded PTP Version	Inter Message Time	Message Type (1280)	sequenceId	controlField (1280)	logMessageInterval (640)	PTP Body F
P	2287	11.906950374000	2.1	0.031250031250	PDEL-RESP	581	0x0	-5	reqRcptTstamp= 1970 1 00 00 12 906950374
P	2288	11.914662500000	2.1	0.007812500000	SYNC	1574	0x0	-7	origTstamp= 1970 1 00 1
P	2289	11.921875000000	2.1	0.007812500000	SYNC	1575	0x0	-7	origTstamp= 1970 1 00 1
P	2290	11.929687500000	2.1	0.007812500000	SYNC	1576	0x0	-7	origTstamp= 1970 1 00 1
P	2291	11.937500000000	2.1	0.007812500000	SYNC	1577	0x0	-7	origTstamp= 1970 1 00 1
P	2292	11.937500000000	2.1	0.031250000000	PDEL-REQ	582	0x0	-5	
P	2293	11.937500232250	2.1	0.031251888250	PDEL-RESP	582	0x0	-5	reqRcptTstamp= 1970 1 00 00 12 937500232
P	2294	11.945312500000	2.1	0.007812500000	SYNC	1578	0x0	-7	origTstamp= 1970 1 00 1
P	2295	11.953125000000	2.1	0.007812500000	SYNC	1579	0x0	-7	origTstamp= 1970 1 00 1
P	2296	11.960937500000	2.1	0.007812500000	SYNC	1580	0x0	-7	origTstamp= 1970 1 00 1
P	2297	11.968750000000	2.1	0.007812500000	SYNC	1581	0x0	-7	origTstamp= 1970 1 00 1
P	2298	11.968750000000	2.1	0.031250000000	PDEL-REQ	583	0x0	-5	
P	2299	11.968750482750	2.1	0.031248250500	PDEL-RESP	583	0x0	-5	reqRcptTstamp= 1970 1 00 00 12 968750482
P	2300	11.976562500000	2.1	0.007812500000	SYNC	1582	0x0	-7	origTstamp= 1970 1 00 1
P	2301	11.984375000000	2.1	0.007812500000	SYNC	1583	0x0	-7	origTstamp= 1970 1 00 1
P	2302	11.992187500000	2.1	0.007812500000	SYNC	1584	0x0	-7	origTstamp= 1970 1 00 1
P	2303	12.000000000000	2.1	0.007812500000	SYNC	1585	0x0	-7	origTstamp= 1970 1 00 1
P	2304	12.000000000000	2.1	0.031250000000	PDEL-REQ	584	0x0	-5	
P	2305	12.000000403000	2.1	0.031249920250	PDEL-RESP	584	0x0	-5	reqRcptTstamp= 1970 1 00 00 13 000000403
P	2306	12.007812500000	2.1	0.007812500000	SYNC	1586	0x0	-7	origTstamp= 1970 1 00 1
P	2307	12.015625000000	2.1	0.007812500000	SYNC	1587	0x0	-7	origTstamp= 1970 1 00 1
P	2308	12.023437500000	2.1	0.007812500000	SYNC	1588	0x0	-7	origTstamp= 1970 1 00 1
P	2309	12.031250000000	2.1	0.007812500000	SYNC	1589	0x0	-7	origTstamp= 1970 1 00 1
P	2310	12.031250000000	2.1	0.031250000000	PDEL-REQ	585	0x0	-5	
P	2311	12.031251871500	2.1	0.031251468500	PDEL-RESP	585	0x0	-5	reqRcptTstamp= 1970 1 00 00 13 031251871
P	2312	12.039062500000	2.1	0.007812500000	SYNC	1590	0x0	-7	origTstamp= 1970 1 00 1
P	2313	12.046875000000	2.1	0.007812500000	SYNC	1591	0x0	-7	origTstamp= 1970 1 00 1
P	2314	12.054687500000	2.1	0.007812500000	SYNC	1592	0x0	-7	origTstamp= 1970 1 00 1
P	2315	12.062500000000	2.1	0.007812500000	SYNC	1593	0x0	-7	origTstamp= 1970 1 00 1
P	2316	12.062500000000	2.1	0.031250000000	PDEL-REQ	586	0x0	-5	
P	2317	12.062500228750	2.1	0.031250337250	PDEL-RESP	586	0x0	-5	reqRcptTstamp= 1970 1 00 00 13 062500228
P	2318	12.070312500000	2.1	0.007812500000	SYNC	1594	0x0	-7	origTstamp= 1970 1 00 1

2.8 Show Filters

The PFV filters feature allows you to focus on a specific subset of the current PFV analysis which can be useful to identify specific problems and diagnose issues with the devices participating in the exchange of data captured.

Access to the applicable filters is toggled by clicking on the 'Show Filters' icon:

The screenshot shows the PFV interface with a list of packets and a filters panel. The filters panel is open, showing various filter criteria. The 'Show Filters' icon is highlighted in the top right corner of the interface.

Direction	Packet #	Arrival Time	Decoded PTP Version	Inter Message Time	Message Type (1280)	sequenceld	controlField (1280)	logMessage (640)
2	9	0.046875000000	2.1	0.007812500000	SYNC	55	0x0	-7
2	10	0.054887500000	2.1	0.007812500000	SYNC	56	0x0	-7
2	11	0.062500000000	2.1	0.007812500000	SYNC	57	0x0	-7
2	12	0.062500000000	2.1	0.031250000000	PDEL-REQ	202	0x0	-5
2	13	0.062502039250	2.1	0.031250738500	PDEL-RESP	202	0x0	-5
2	14	0.070312500000	2.1	0.007812500000	SYNC	58	0x0	-7
2	15	0.078125000000	2.1	0.007812500000	SYNC	59	0x0	-7
2	16	0.085937500000	2.1	0.007812500000	SYNC	60	0x0	-7
2	17	0.093750000000	2.1	0.007812500000	SYNC	61	0x0	-7
2	18	0.093750000000	2.1	0.031250000000	PDEL-REQ	203	0x0	-5
2	19	0.093750759250	2.1	0.031248720000	PDEL-RESP	203	0x0	-5
2	20	0.101562500000	2.1	0.007812500000	SYNC	62	0x0	-7
2	21	0.109375000000	2.1	0.007812500000	SYNC	63	0x0	-7
2	22	0.117187500000	2.1	0.007812500000	SYNC	64	0x0	-7
2	23	0.125000000000	2.1	0.007812500000	SYNC	65	0x0	-7
2	24	0.125000000000	2.1	0.031250000000	PDEL-REQ	204	0x0	-5
2	25	0.125001606500	2.1	0.031250847250	PDEL-RESP	204	0x0	-5
2	26	0.132812500000	2.1	0.007812500000	SYNC	66	0x0	-7
2	27	0.140625000000	2.1	0.007812500000	SYNC	67	0x0	-7
2	28	0.148437500000	2.1	0.007812500000	SYNC	68	0x0	-7
2	29	0.156250000000	2.1	0.007812500000	SYNC	69	0x0	-7
2	30	0.156250000000	2.1	0.031250000000	PDEL-REQ	205	0x0	-5
2	31	0.156251126500	2.1	0.031249520000	PDEL-RESP	205	0x0	-5

Due to the complexity of the PFV rules checking engine, filters are restricted to targeted fields based on capture file content. The 'Show Filters' panel will display all available filters for the current analysis.

Consider the following filter settings:

The screenshot shows the PFV interface with the filters panel open. The filters are set to: Arrival Time (From: 10, To: 11), Message Type (PDEL-RESP), and Has Error (checked).

Here PFV restricts the records shown to those with an arrival time between 10 and 11 seconds, of message type PDEL-RESP that are marked as in error (the 'Has Error' flag specifies this).

Applying the above filter gives the following results:

Direction	Packet #	Arrival Time	Decoded PTP Version	Inter Message Time	Message Type	sequenceld	controlField (1280)	logMessageInterval (640)	PTP Body
←	1921	10.000002488250	2.1	0.031252309250	PDEL-RESP	520	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.000002488
←	1927	10.031251063000	2.1	0.031248574750	PDEL-RESP	521	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.031251063
←	1933	10.062500408250	2.1	0.031249345250	PDEL-RESP	522	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.062500408
←	1939	10.093752469500	2.1	0.031252061250	PDEL-RESP	523	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.093752469
←	1945	10.125002155750	2.1	0.031249686250	PDEL-RESP	524	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.125002155
←	1951	10.156250335000	2.1	0.031248179250	PDEL-RESP	525	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.156250335
←	1957	10.187501963250	2.1	0.031251628250	PDEL-RESP	526	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.187501963
←	1963	10.218750548500	2.1	0.031248585250	PDEL-RESP	527	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.218750548
←	1969	10.250002473250	2.1	0.031251924750	PDEL-RESP	528	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.250002473
←	1975	10.281252451250	2.1	0.031249978000	PDEL-RESP	529	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.281252451
←	1981	10.312500739000	2.1	0.031248287750	PDEL-RESP	530	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.312500739
←	1987	10.343752308750	2.1	0.031251569750	PDEL-RESP	531	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.343752308
←	1993	10.375002455250	2.1	0.031250148500	PDEL-RESP	532	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.375002455
←	1999	10.406250090000	2.1	0.031247553750	PDEL-RESP	533	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.406250090
←	2005	10.437500162500	2.1	0.031250163500	PDEL-RESP	534	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.437500162
←	2011	10.468750621750	2.1	0.031250459250	PDEL-RESP	535	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.468750621
←	2017	10.500001928500	2.1	0.031261306750	PDEL-RESP	536	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.500001928
←	2023	10.531252440250	2.1	0.031260520750	PDEL-RESP	537	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.531252440
←	2029	10.562502491900	2.1	0.031250042250	PDEL-RESP	538	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.562502491
←	2035	10.593752058500	2.1	0.031249567000	PDEL-RESP	539	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.593752058
←	2041	10.625001543000	2.1	0.031249484500	PDEL-RESP	540	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.625001543
←	2047	10.656252022000	2.1	0.031250479000	PDEL-RESP	541	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.656252022
←	2053	10.687500890000	2.1	0.031248868000	PDEL-RESP	542	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.687500890
←	2059	10.718750911250	2.1	0.031250021250	PDEL-RESP	543	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.718750911
←	2065	10.750001703250	2.1	0.031250792000	PDEL-RESP	544	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.750001703
←	2071	10.781252272250	2.1	0.031250959000	PDEL-RESP	545	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.781252272
←	2077	10.812501359000	2.1	0.031249086750	PDEL-RESP	546	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.812501359
←	2083	10.843752312250	2.1	0.031250953250	PDEL-RESP	547	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.843752312
←	2089	10.875002127250	2.1	0.031249815000	PDEL-RESP	548	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.875002127
←	2095	10.906252433000	2.1	0.031250305750	PDEL-RESP	549	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.906252433
←	2101	10.937501462750	2.1	0.031249029750	PDEL-RESP	550	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.937501462
←	2107	10.968750764750	2.1	0.031249302000	PDEL-RESP	551	0x0	-5	reqRcptTstamp= 1970 1 00.00.11.968750764

The results focus on the errored messages of the chosen type over the defined period which may help to highlight other errors particular to this filtered dataset. This can be useful to identify subtle inconsistencies in how the relevant standards have been interpreted and applied.

The current active filter configuration can be cleared from the 'Show Filters' panel to restore the analysis to its original state:

Filters

Arrival Time Has Error

From: To:

Inter Message Time Has Error

From: To:

Message Type Has Error

SYNC PDEL-REQ PDEL-RESP

Sequenceld Has Error

From: To:

Direction Has Error DomainNumber Has Error

Column header error counts reflect total errors and remain unchanged by filters.

Apply

2.9 Loading a Licence File

The stand-alone PFV application (that is, installed on the user's PC) can be licensed. This license allows rules checking to be performed on:

- Any PTP or CCSA ToD capture from Paragon-X (regardless of options installed on the instrument).
- Any PTP capture from Paragon-neo/Paragon-100G (regardless of options installed on the instrument).
- Any .pcap containing PTP data with Ethernet or IPv4 encapsulation.
- PFV does not require any licence to analyze the .pcap file captured with Sentinel or Sentry.

Licences can be loaded from the **System** page under **Options**.

The screenshot shows the PFV application interface. The top navigation bar includes 'Files', 'PTP', 'ToD', 'Report', 'System', and 'Help'. The left sidebar has 'Options', 'Status', and 'Preferences'. The main content area is titled 'PFV Keys' and 'Options'. The 'PFV Keys' section displays two keys: Key 1: 98E743821547 and Key 2: 4C1D96E42843. The 'Options' section includes a table with columns for Option, Description, Expiry Date, and Key. The table lists two options: PFV-PTP and PFV-ToD, both with permanent expiry dates and the same key (4C1D96E42843). A 'Load licence file' button is also visible.

Option	Description	Expiry Date	Key
PFV-PTP	PTP Profile Conformance Testing for PFV	Permanent	4C1D96E42843
PFV-ToD	ToD Conformance Testing for PFV	Permanent	4C1D96E42843

PFV Displays one or more keys that are PC specific, please provide these keys when ordering a standalone PFV licence. Calnex will supply the licence that locks standalone PFV to the specific PC.

To load a licence, click **Load new licence file** and select the appropriate licence file.

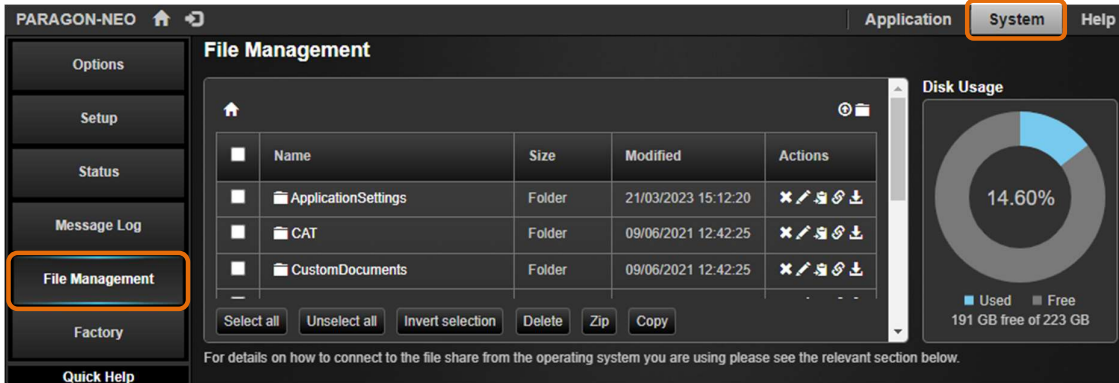
3 Customising the Rules File

Note: Some knowledge of XML and XML terminology is assumed.

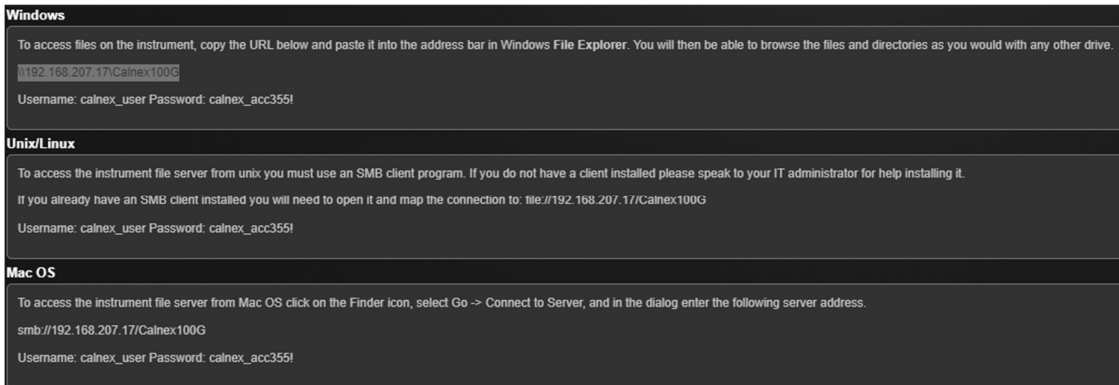
3.1 Accessing Paragon-neo/Paragon-100G Filesystem

The Paragon-neo/Paragon-100G filesystem can be accessed using Windows explorer, Unix/Linux or Mac OS.

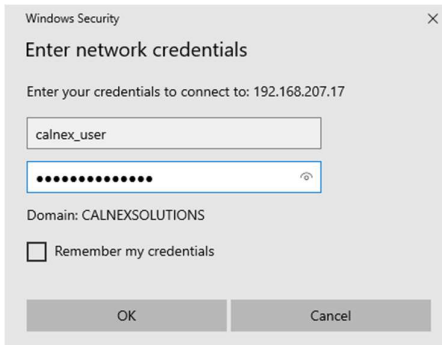
Open the Paragon-neo/Paragon-100G GUI in the web browser, select **System** in the top-right menu bar and then **File Management** in the left side menu.



In the Windows section, copy the URL and paste it into the address bar in Windows Explorer.



Windows Security will ask you for your username and password.



The Windows Explorer will show you the files after successful authentication.

Name	Date modified	Type	Size
ApplicationSettings	21/03/2023 03:12 PM	File folder	
CAT	09/06/2021 01:42 PM	File folder	
CustomDocuments	09/06/2021 01:42 PM	File folder	
Filters	09/06/2021 01:42 PM	File folder	
Logs	18/10/2022 12:44 PM	File folder	
Patterns	09/03/2023 03:20 PM	File folder	
PFV	09/06/2021 01:42 PM	File folder	
RemoteControl	21/03/2023 03:12 PM	File folder	
Sessions	04/04/2023 10:47 AM	File folder	
UserSettings	03/04/2023 07:10 PM	File folder	
Paragon-Neo_06.00.22.tar	04/06/2021 12:51 PM	TAR File	915,930 KB

3.2 File Locations

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\

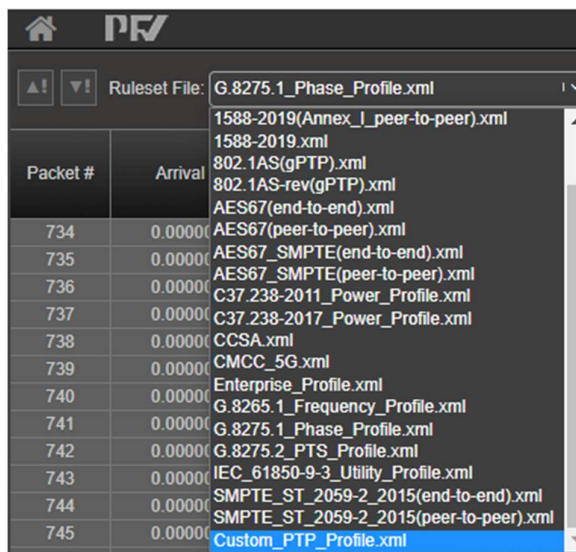
For the PC-based PFV, user-defined rules files should be stored in:
C:\Users\<User>\Documents\Calnex\PFV\Rules\

For Paragon-100G, user-defined rules files should be stored in:
\\<P100G IP>\Calnex100G\PFV\Rules

For Paragon-Neo, user-defined rules files should be stored in:
\\<P-Neo IP>\calnex100g\PFV\Rules

The application will populate the rules pull-down from all xml files in these folders.

The easiest way to create a custom rules file is to copy one of the pre-defined files (e.g., 1588-2008.xml) to the user-defined folder and re-name it. The new user-defined file can then be edited.



3.3 XML Syntax – Top-Level Structure

The rules are defined within the `testSpecification` element. Only one of these elements can be present in the file.

```
<?xml version="1.0" encoding="utf-8"?>
<testSpecification schemaVersion="0.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.w3.org/2001/XMLSchema-instance" >
  <fileInformation fileVersion="2.0" name="G.8275.1 Phase Profile" />
  <referenceSource documentNumber="G.8275.1">ITU-T G.8275-1/Y.1369.1 (2020)/Amd.2 (06/2021)</referenceSource>
  <referenceSource documentNumber="1588-2008">IEEE Std 1588-2008</referenceSource>
  <referenceSource documentNumber="1588-2019">IEEE Std 1588-2019</referenceSource>
  <ptpGeneral>
  <ptpHeader>
  <announceMessageBody>
  <ethernetHeader>
    <field description="Ethernet Header, Destination Address (E2E Multicast)" referenceDocumentNumber="G.8275.1" ref
    <fieldName>ethDestinationAddress</fieldName>
    <enabled>true</enabled>
    <conditions description="Only applies to E2E messages when multicast">
      <AND>
        <OR>
          <EQUAL>
```

The `testType` attribute is used to specify whether the rules apply to ToD or PTP. The valid `testTypes` are:

- PTP: the default
- ToD CCSA: For CCSA format ToD

The `testType` attribute does not need to be specified for PTP files since PTP is the default. For ToD, ToD CCSA must be specified.

```
<?xml version="1.0" encoding="utf-8"?>
<testSpecification testType="ToD CCSA" schemaVersion="0.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
  <fileInformation fileVersion="0.1" name="CCSA ToD PFV Rules" />
  <referenceSource documentNumber="CCSA SG15-C1324">CCSA ToD Contribution</referenceSource>
  <ccsaTimeInformationMessage>
    <field description="CCSA Time Information Message length is 16" referenceDocumentNumber="CCSA SG15-C1324" ref
    <fieldName>Length</fieldName>
    <enabled>true</enabled>
    <tests description="CCSA Time Information message length = 16">
      <EQUAL>
        <value>0x10</value>
      </EQUAL>
    </tests>
  </field>
```

The `schemaVersion` attribute to this element specifies the version of the schema associated with this rules file. This can be used to ensure compatibility with the version of the PFV application. The `noNamespaceSchemaLocation` attribute specifies the location of the schema file (see "Editing a Rules File and the XML Schema" below). **These attributes should not be changed.**

The `fileInformation` element is intended to be modifiable and contains two attributes: `fileVersion` can be used to provide version control for the content of this rules file; `name` provides a user-friendly string to identify the file. Only one `fileInformation` element can be present in the file.

The `referenceSource` element provides a means of listing the documents from which the rules have been created. The `documentNumber` attribute specifies the unique identifier for the document; the `name` attribute can be used to specify the document name. There can be as many `referenceSource` elements in the file as required.

The remaining elements (`ptpGeneral`, `ptpHeader`, `announceMessageBody`, `managementMessageBody`, `ethernetHeader`, `ipv4Header`, `ipv6Header`, `tlvData` and `todMessage`) are used to define the verification tests that will be performed by the PFV application.

The `todMessage` element is only relevant when `testType` is ToD CCSA.

3.4 PTP Elements (`testType = PTP`)

The `ptpGeneral`, `ptpHeader`, `announceMessageBody`, `managementMessageBody`, `ethernetHeader`, `ipv4Header`, `ipv6Header` and `tlvData` are used to define the test that will be performed for PTP messages. In some cases, the capture file loaded into PFV will not contain all the fields that can be tested by the rules file. In this case, these rules will be ignored.

- `ptpGeneral`: Defines tests that apply to the capture as a whole and which are not specific to any single message.
- `ptpHeader`: Defines the tests applied to the PTP header of each PTP message in the capture.
- `announceMessageBody`: Defines the tests applied to the PTP Message Body of each ANNOUNCE message in the capture.
- `managementMessageBody`: Defines the tests applied to the PTP Message Body of each MANAGEMENT message in the capture.
- `ethernetHeader`: Defines the tests applied to the ethernet header of each message in the capture.
- `ipv4Header`: Defines the tests applied to the IPv4 header of each message in the capture.
- `tlvData`: Defines the tests applied to TLV data decoded from each message in the capture carrying the relevant data.

The content of these elements is described in detail in the sections below.

XML Syntax - `ptpGeneral`

This section contains a definition of the tests to be applied to the complete capture and not to any single message. Currently, `averageMessageRate` is the only general test supported. There can be as many `averageMessageRate` elements as required.

Note: The average message rate is calculated by dividing the number of messages in the capture by the capture length (in seconds).

```
<ptpGeneral>
  <averageMessageRate description="SYNC message rate 16 per second" referenceDocumentNumber="G.8275.1" referenceSectionNumber="6.2.8" <enabled>true</enabled>
    <messageType>SYNC</messageType>
    <tests>
      </averageMessageRate>
      <averageMessageRate description="FUP message rate (sames as SYNC) 16 per second" referenceDocumentNumber="G.8275.1" referenceSectionNumber="6.2.8" <enabled>true</enabled>
      <averageMessageRate description="DEL-REQ message rate less or equal 16 per second. An 'upper' rate tolerance. Message Rate calculation. A lower tolerance of 30% (11.2pps) and an additional 10% of that (resulting in 10.8 pps) shall be used. A higher tolerance of 10% shall be used. A lower tolerance of 30% (11.2pps) and an additional 10% of that (resulting in 10.8 pps) shall be used." referenceDocumentNumber="G.8275.1" referenceSectionNumber="6.2.8">
      <averageMessageRate description="DEL-RESP message rate (same as DEL-REQ). From 1588 2008 & 2019: The Delay Tolerance shall be less or equal to 16 pps." referenceDocumentNumber="G.8275.1" referenceSectionNumber="6.2.8">
      <averageMessageRate description="ANNOUNCE message rate 8 per second" referenceDocumentNumber="G.8275.1" referenceSectionNumber="6.2.8" <enabled>true</enabled>
      <!-- IMT Check. Condition set to only run on PTP v2.0 flows, as PTP v2.1 flows use IEEE-2019 checks-->
      <field description="Automated test for checking SYNC and ANNOUNCE IMTs, IMT should match the logMessageInterval" referenceDocumentNumber="G.8275.1" referenceSectionNumber="6.2.8">
      <!-- MAX IMT Check'. Condition set to only run on PTP v2.0 flows, as PTP v2.1 uses automatic max IMT rule to calculate the maximum inter-message interval-->
      <field description="Inter Message Time for DEL-REQ shall be less or equal to 0.125s" referenceDocumentNumber="G.8275.1" referenceSectionNumber="6.2.8">
      <!-- PTP v2.1 mean message interval tests, condition to only run on v2.1 or greater flows-->
      <meanInterMessageInterval description="Tests the arithmetic mean message interval against 1588 2019 defined limit" referenceDocumentNumber="G.8275.1" referenceSectionNumber="6.2.8">
      <meanInterMessageInterval description="Tests the arithmetic mean message interval against 1588 2019 defined limit" referenceDocumentNumber="G.8275.1" referenceSectionNumber="6.2.8">
      <!-- PTP v2.1 inter message interval pass percentage tests, condition to only run on v2.1 or greater flows-->
      <interMessageIntervalPassPercentage description="Checks that a required minimum percentage of inter-message interval tests pass" referenceDocumentNumber="G.8275.1" referenceSectionNumber="6.2.8">
      <interMessageIntervalPassPercentage description="Checks that a required minimum percentage of inter-message interval tests pass" referenceDocumentNumber="G.8275.1" referenceSectionNumber="6.2.8">
      <!-- maximumInterMessageTimeCheck SYNC, ANNOUNCE, DEL-REQ. Condition to only run on v2.1 or greater flows-->
      <field description="Automated test for checking that SYNC, ANNOUNCE and DEL-REQ IMTs should not exceed twice the maximum inter-message interval" referenceDocumentNumber="G.8275.1" referenceSectionNumber="6.2.8">
    </ptpGeneral>
</ptpHeader>
```

The `averageMessageRate` element contains the following attributes:

- `description`: A string that describes the test.
- `referenceDocumentNumber`: The document number on which the test is based. This is intended to refer to the `referenceSource` element in the `testSpecification` header.
- `referenceSectionNumber`: The section within the reference source document that has been used as the basis for the test.

The `enabled` element contains a boolean value (`true`, `false`, `0` or `1`) that determines whether this test will be executed. This allows the test to be defined in the rules file but disabled if required.

The `messageType` element specifies the messages to which this test applies. The messages supported are defined in "Message Types" below.

The `tests` element defines the test to be applied to the given `messageType`. A test is considered a pass if the test result is true.

```
<averageMessageRate description="SYNC message rates between 1 every 2 secs and 128 per
second" referenceDocumentNumber="1588-2008" referenceSectionNumber="A.9.4.2">
  <enabled>true</enabled>
  <messageType>SYNC</messageType>
  <tests>
    <AND>
      <GREATER>
        <value>0.5</value>
        <tolerance>30</tolerance>
      </GREATER>
      <LESS>
        <value>128</value>
        <tolerance>30</tolerance>
      </LESS>
    </AND>
  </tests>
</averageMessageRate>
```

In the example above, the test will produce a pass result if the average SYNC message rate is greater than 0.5 +/-30% and less than 128 +/-30%.

Within a `tests` element, there must be at least one comparison operator (such as `GREATER` or `EQUAL`). Comparison operators can be contained within logical boolean operators (`AND` or `OR`). The list of supported comparison and logical operators is detailed in "Comparison Operators" and "Logical Operators" below.

Within an `averageMessageRate` comparison element, there are two elements to specify the `value` to be tested and the `tolerance` for the test. In other words, the test will compare the average message rate against `value` +/- `tolerance`.

The `value` is specified in messages per second, `tolerance` in percent.

XML Syntax - `ptpHeader`

This section contains rules that verify the content of the PTP header fields in individual messages. The `ptpHeader` element contains a number of `field` elements; each `field` defines a test to be applied to each individual message in the capture. The `field` element allows tests to be applied to specific messages or to be applied only when other conditions are satisfied. There can be only one `ptpHeader` in a rules file but there can be as many `field` elements as required.

```
<ptpHeader>
  <!--Decoded PTP version test (2.0 pass only)-->
  <field>
    <field description="PTP Header, message type" referenceDocumentNumber="1588-2008" referenceSectionNumber="13.3.2.2">
    <field description="PTP Header, transportSpecific. Disabled - could be either" referenceDocumentNumber="1588-2008" referenceSectionNumber="13.3.2.2">
    <field description="PTP Header, versionPTP. Spec allows 1 or 2. For the purposes of testing, look for 2" referenceDocumentNumber="1588-2008" referenceSectionNumber="13.3.2.2">
    <field description="PTP Header, messageLength. For the purposes of testing, look for minimum possible" referenceDocumentNumber="1588-2008" referenceSectionNumber="7.1">
    <field description="PTP Header, domainNumber" referenceDocumentNumber="1588-2008" referenceSectionNumber="7.1">
    <field description="PTP Header, sourcePortIdentity" referenceDocumentNumber="1588-2008" referenceSectionNumber="7.1">
    <field description="PTP Header, sequenceId" referenceDocumentNumber="1588-2008" referenceSectionNumber="13.3.2.9">
    <field description="PTP Header, controlfield" referenceDocumentNumber="1588-2008" referenceSectionNumber="13.3.2.2">
    <field description="PTP Header, logMessageInterval - check for 0x7F" referenceDocumentNumber="1588-2008" referenceSectionNumber="13.3.2.2">
    <field description="PTP Header, logMessageInterval - check for 0x7F (unicast)" referenceDocumentNumber="1588-2008" referenceSectionNumber="13.3.2.2">
    <field description="PTP Header, logMessageInterval - check for NOT 0x7F (Announce)" referenceDocumentNumber="1588-2008" referenceSectionNumber="13.3.2.2">
    <field description="PTP Header, logMessageInterval - check for NOT 0x7F (Sync, Follow-Up and Delay-Resp when multicast)" referenceDocumentNumber="1588-2008" referenceSectionNumber="13.3.2.2">
    <!-- Reserved Fields (other than flags) -->
    <field description="PTP Header, reservedfield0 (octet 1, bits 4-7)" referenceDocumentNumber="1588-2008" referenceSectionNumber="13.3.2.2">
    <field description="PTP Header, reservedfield1 (octet 5)" referenceDocumentNumber="1588-2008" referenceSectionNumber="13.3.2.2">
    <field description="PTP Header, reservedfield2 (octet 16)" referenceDocumentNumber="1588-2008" referenceSectionNumber="13.3.2.2">
    <!-- Flag Fields - Octet 0 -->
    <field description="PTP Header, flagfield, octet 0: alternateMasterFlag (Announce, Sync, Fup, Del-Resp). Test disabled since could be either" referenceDocumentNumber="1588-2008" referenceSectionNumber="13.3.2.2">
    <field description="PTP Header, flagfield, octet 0: alternateMasterFlag (Not Announce, Sync, Fup, Del-Resp). Not disabled since could be either" referenceDocumentNumber="1588-2008" referenceSectionNumber="13.3.2.2">
    <field description="PTP Header, flagfield, octet 0: twoStepFlag. Dependent on the operation of the device. Disabled since could be either" referenceDocumentNumber="1588-2008" referenceSectionNumber="13.3.2.2">
    <field description="PTP Header, flagfield, octet 0: twoStepFlag. Should be true only in SYNC and PDEL-RESP messages" referenceDocumentNumber="1588-2008" referenceSectionNumber="13.3.2.2">
    <field description="PTP Header, flagfield, octet 0: unicastFlag. Disabled since could be either" referenceDocumentNumber="1588-2008" referenceSectionNumber="13.3.2.2">
    <fieldName>unicastFlag</fieldName>
    <enabled>>false</enabled>
```

Note: Tests and conditions can only refer to the current message e.g., each test applies only to a single message type.

The content of the `field` element is described in detail in "XML Syntax - `field`" below.

XML Syntax – announceMessageBody

This section contains rules that verify the content of the ANNOUNCE message body fields in individual ANNOUNCE messages. The `announceMessageBody` element contains a number of `field` elements; each `field` defines a test to be applied to each individual ANNOUNCE message in the capture. The `field` element allows tests to be applied only when other conditions are satisfied. There can be only one `announceMessageBody` in a rules file but there can be as many `field` elements as required.

```
<announceMessageBody>
  <field description="Announce Message Body, currentUtcOffset" referenceDocumentNumber="1588-2008" referenceSectionNumber="5.3.4">
  <field description="Announce Message Body, grandmasterPriority1. Disabled - could be any value 0-255" referenceDocumentNumber="1588-2008" referenceSectionNumber="5.3.4">
  <field description="Announce Message Body, grandmasterClockQuality.ClockClass" referenceDocumentNumber="1588-2008" referenceSectionNumber="5.3.4">
  <field description="Announce Message Body, grandmasterClockQuality.ClockAccuracy" referenceDocumentNumber="1588-2008" referenceSectionNumber="5.3.4">
  <field description="Announce Message Body, grandmasterClockQuality.offsetScaledLogVariance" referenceDocumentNumber="1588-2008" referenceSectionNumber="5.3.4">
  <field description="Announce Message Body, grandmasterPriority2. Disabled - could be any value 0-255" referenceDocumentNumber="1588-2008" referenceSectionNumber="5.3.4">
  <field description="Announce Message Body, grandmasterTimeSource" referenceDocumentNumber="1588-2008" referenceSectionNumber="5.3.4">
</announceMessageBody>
```

Note: Tests and conditions can only refer to the current ANNOUNCE message, that is, each test applies only to a single message.

The content of the `field` element is described in detail in “XML Syntax - `field`” below.

XML Syntax – announceMessageCMCC5GTLV

This section contains rules that verify the content of the CMCC 5G ANNOUNCE message TLV fields in individual ANNOUNCE messages. The `announceMessageBody` element contains a number of `field` elements; each `field` defines a test to be applied to each individual ANNOUNCE message in the capture. The `field` element allows tests to be applied only when other conditions are satisfied. There can be only one `announceMessageBody` in a rules file but there can be as many `field` elements as required.

```
<field description="Announce Message Body, TLV Type field" referenceDocumentNumber="China Mobile Specification for Ultra Precision Time Synchronization" referenceSectionNumber="5.3.4">
<field description="Announce Message Body, TLV Length field" referenceDocumentNumber="China Mobile Specification for Ultra Precision Time Synchronization" referenceSectionNumber="5.3.4">
<field description="Announce Message Body, TLV Organisation ID field" referenceDocumentNumber="China Mobile Specification for Ultra Precision Time Synchronization" referenceSectionNumber="5.3.4">
<field description="Announce Message Body, TLV Organisation Subtype field" referenceDocumentNumber="China Mobile Specification for Ultra Precision Time Synchronization" referenceSectionNumber="5.3.4">
<field description="Announce Message Body, CCSA High Precision Accuracy Level 1 field" referenceDocumentNumber="China Mobile Specification for Ultra Precision Time Synchronization" referenceSectionNumber="5.3.4">
<field description="Announce Message Body, CCSA High Precision Steps Removed Level 1 field" referenceDocumentNumber="China Mobile Specification for Ultra Precision Time Synchronization" referenceSectionNumber="5.3.4">
<field description="Announce Message Body, CCSA High Precision Accuracy Level 2 field" referenceDocumentNumber="China Mobile Specification for Ultra Precision Time Synchronization" referenceSectionNumber="5.3.4">
<field description="Announce Message Body, CCSA High Precision Steps Removed Level 2 field" referenceDocumentNumber="China Mobile Specification for Ultra Precision Time Synchronization" referenceSectionNumber="5.3.4">
<field description="Announce Message Body, CCSA High Precision Accuracy Level 3 field" referenceDocumentNumber="China Mobile Specification for Ultra Precision Time Synchronization" referenceSectionNumber="5.3.4">
<field description="Announce Message Body, CCSA High Precision Steps Removed Level 3 field" referenceDocumentNumber="China Mobile Specification for Ultra Precision Time Synchronization" referenceSectionNumber="5.3.4">
</announceMessageBody>
```

Note: Tests and conditions can only refer to the current ANNOUNCE message, that is, each test applies only to a single message.

The content of the `field` element is described in detail in “XML Syntax - `field`” below.

XML Syntax – managementMessageBody

This section contains rules that verify the content of the MANAGEMENT message body fields in individual MANAGEMENT messages. The `managementMessageBody` element contains a number of `field` elements; each `field` defines a test to be applied to each individual MANAGEMENT message in the capture. The `field` element allows tests to be applied only when other conditions are satisfied. There can be only one `managementMessageBody` in a rules file but there can be as many `field` elements as required.

```
<managementMessageBody>
  <field referenceDocumentNumber="ST 2059-2:2015" referenceSectionNumber="5.13">
    <fieldName>MgtTargetPortIdentity</fieldName>
    <enabled>true</enabled>
    <conditions description="The following condition should identify a management message carrying an SMPTE SM TLV">
      <tests failMessage="MANAGEMENT COMMAND message carrying SMPTE SM TLV should have a target port identity of 0xFFFF"
    </tests failMessage="MANAGEMENT COMMAND message carrying SMPTE SM TLV should have a target port identity of 0xFFFF"
    </conditions>
  </field>
  <field referenceDocumentNumber="ST 2059-2:2015" referenceSectionNumber="5.13">
    <fieldName>MgtStartingBoundaryHops</fieldName>
    <enabled>true</enabled>
    <conditions>
      <tests failMessage="MANAGEMENT COMMAND message carrying SMPTE SM TLV should have a maximum startingBoundaryHops va
    </tests failMessage="MANAGEMENT COMMAND message carrying SMPTE SM TLV should have a maximum startingBoundaryHops va
    </conditions>
  </field>
  <field referenceDocumentNumber="ST 2059-2:2015" referenceSectionNumber="5.13">
    <fieldName>MgtReserved0</fieldName>
    <enabled>true</enabled>
    <conditions description="The following condition should identify a management message carrying an SMPTE SM TLV">
      <tests failMessage="MANAGEMENT COMMAND message carrying SMPTE SM TLV Reserved field should be 0 (bits 4-7, offset
    </tests failMessage="MANAGEMENT COMMAND message carrying SMPTE SM TLV Reserved field should be 0 (bits 4-7, offset
    </conditions>
  </field>
  <field referenceDocumentNumber="ST 2059-2:2015" referenceSectionNumber="5.13">
    <fieldName>MgtReserved1</fieldName>
    <enabled>true</enabled>
    <conditions description="The following condition should identify a management message carrying an SMPTE SM TLV">
      <tests failMessage="MANAGEMENT COMMAND message carrying SMPTE SM TLV Reserved field should be 0 (bits 0-7, offset
    </tests failMessage="MANAGEMENT COMMAND message carrying SMPTE SM TLV Reserved field should be 0 (bits 0-7, offset
    </conditions>
  </field>
</managementMessageBody>
```

Note: Tests and conditions can only refer to the current MANAGEMENT message, that is, each test applies only to a single message.

The content of the `field` element is described in detail in "XML Syntax - `field`" below.

XML Syntax – ethernetHeader

This section contains rules that verify the content of the Ethernet header in individual packets. Note that these rules only apply when the capture contains Ethernet headers (these are not present in Paragon-X timing capture files).

The `ethernetHeader` element contains a number of `field` elements; each `field` defines a test to be applied to each message in the capture. The `field` element allows tests to be applied only when other conditions are satisfied. There can be only one `ethernetHeader` in a rules file but there can be as many `field` elements as required.

```
<ethernetHeader>
  <field description="Ethernet Header, Destination Address (E2E Multicast)" referenceDocumentNumber="G.8275.1" referenceSectionNumber="A.3.3">
  <field description="Ethernet Header, Source Address" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="Ethernet Header, EtherType" referenceDocumentNumber="G.8275.1" referenceSectionNumber="A.3.3">
  <field description="Ethernet Header, VLAN1 PID. VLANs are not allowed for GM, BC, SC. Allowance for TC is in Appendix I however,
    "G.8275.1" referenceSectionNumber="6.2.7">
  <field description="Ethernet Header, VLAN1 PCP. VLANs are not allowed for GM, BC, SC. Allowance for TC is in Appendix I however,
    "G.8275.1" referenceSectionNumber="6.2.7">
  <field description="Ethernet Header, VLAN1 DEI. VLANs are not allowed for GM, BC, SC. Allowance for TC is in Appendix I however,
    "G.8275.1" referenceSectionNumber="6.2.7">
  <field description="Ethernet Header, VLAN1 VID. VLANs are not allowed for GM, BC, SC. Allowance for TC is in Appendix I however,
    "G.8275.1" referenceSectionNumber="6.2.7">
  <field description="Ethernet Header, VLAN2 PID. VLANs are not allowed for GM, BC, SC. Allowance for TC is in Appendix I however,
    "G.8275.1" referenceSectionNumber="6.2.7">
  <field description="Ethernet Header, VLAN2 PCP. VLANs are not allowed for GM, BC, SC. Allowance for TC is in Appendix I however,
    "G.8275.1" referenceSectionNumber="6.2.7">
  <field description="Ethernet Header, VLAN2 DEI. VLANs are not allowed for GM, BC, SC. Allowance for TC is in Appendix I however,
    "G.8275.1" referenceSectionNumber="6.2.7">
  <field description="Ethernet Header, VLAN2 VID. VLANs are not allowed for GM, BC, SC. Allowance for TC is in Appendix I however,
    "G.8275.1" referenceSectionNumber="6.2.7">
```

Note: Tests and conditions can only refer to the current message, that is, each test applies only to a single message type.

The content of the `field` element is described in detail in "XML Syntax - `field`" below.

XML Syntax – ipv4Header

This section contains rules that verify the content of the IPv4 header in individual packets. Note that these rules only apply when the capture contains IPv4 headers (these are not present in Paragon-X timing capture files).

The `ipv4Header` element contains a number of `field` elements; each `field` defines a test to be applied to each message in the capture. The `field` element allows tests to be applied only when other conditions are satisfied. There can be only one `ipv4Header` in a rules file but there can be as many `field` elements as required.

```
<ipv4Header>
  <field description="IPv4 Header, IPv4 Version" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="IPv4 Header, Internet Header Length" referenceDocumentNumber="IEEE-1588" referenceSectionNumber="NA">
  <field description="IPv4 Header, Explicit Congestion Notification" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="IPv4 Header, Total IP Packet Length (Sync, Fup, Del-Req, Signaling)" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="IPv4 Header, Total IP Packet Length (Del-Resp, PDel-Req, PDel-Resp, PDel-Resp-Fup)" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="IPv4 Header, Identification" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="IPv4 Header, Flags" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="IPv4 Header, Fragment Offset" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="IPv4 Header, Time To Live" referenceDocumentNumber="IEEE-1588" referenceSectionNumber="D.3">
  <field description="IPv4 Header, Protocol" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="IPv4 Header, Source Address" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="IPv4 Header, Destination Address (E2E Multicast)" referenceDocumentNumber="1588-2008" referenceSectionNumber="NA">
  <field description="IPv4 Header, Destination Address (P2P Multicast)" referenceDocumentNumber="1588-2008" referenceSectionNumber="NA">
  <field description="IPv4 Header, Checksum" referenceDocumentNumber="None" referenceSectionNumber="NA">
</ipv4Header>
```

Note: Tests and conditions can only refer to the current message, that is, each test applies only to a single message type.

The content of the `field` element is described in detail in "XML Syntax - `field`" below.

XML Syntax – ipv6Header

This section contains rules that verify the content of the IPv6 header in individual packets. Note that these rules only apply when the capture contains IPv6 headers (these are not present in Paragon-X timing capture files).

The `ipv6Header` element contains a number of `field` elements; each `field` defines a test to be applied to each message in the capture. The `field` element allows tests to be applied only when other conditions are satisfied. There can be only one `ipv6Header` in a rules file but there can be as many `field` elements as required.

```
<ipv6Header>
  <field description="IPv6 Header, IPv6 Version" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="IPv6 Header, Source Address" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="IPv6 Header, Destination Address (E2E Multicast)" referenceDocumentNumber="1588-2008" referenceSectionNumber="NA">
  <field description="IPv6 Header, Destination Address (P2P Multicast)" referenceDocumentNumber="1588-2008" referenceSectionNumber="NA">
  <field description="IPv6 Header, NextHeader" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="IPv6 Header, Hop Limit" referenceDocumentNumber="1588-2008" referenceSectionNumber="NA">
</ipv6Header>
```

Note: Tests and conditions can only refer to the current message, that is, each test applies only to a single message type.

The content of the `field` element is described in detail in "XML Syntax - `field`" below.

XML Syntax – `tlvData`

This section contains rules that verify the content of TLV data that may be present in a message. Note that these rules will only be applied to a message where the relevant TLV data type has been detected (these will not be present in Paragon-X timing capture files).

The `tlvData` element contains a number of `field` elements; each `field` defines a test to be applied to each message in the capture. The `field` element allows tests to be applied only when other conditions are satisfied. There can be only one `tlvData` in a rules file but there can be as many `field` elements as required.

```
<tlvData>
  <!-- SMPTE SM (2015) TLV -->
  <!-- SMPTE SM (2015) TLV -->
  <!-- SMPTE SM (2015) TLV -->
  <field description="SMPTE SM 2015 lengthField" referenceDocumentNumber="ST 2059-2:2015" referenceSectionNumber="6.2.1" />
  <field description="SMPTE SM 2015 masterLockingStatus" referenceDocumentNumber="ST 2059-2:2015" referenceSectionNumber="6.2.2" />
  <field description="SMPTE SM 2015 timeAddressFlags" referenceDocumentNumber="ST 2059-2:2015" referenceSectionNumber="6.2.3" />
  <field description="SMPTE SM 2015 timeAddressFlags" referenceDocumentNumber="ST 2059-2:2015" referenceSectionNumber="6.2.4" />
  <field description="SMPTE SM 2015 timeAddressFlags" referenceDocumentNumber="ST 2059-2:2015" referenceSectionNumber="6.2.5" />
  <field description="SMPTE SM 2015 timeAddressFlags" referenceDocumentNumber="ST 2059-2:2015" referenceSectionNumber="6.2.6" />
  <field description="SMPTE SM 2015 timeAddressFlags" referenceDocumentNumber="ST 2059-2:2015" referenceSectionNumber="6.2.7" />
  <field description="SMPTE SM 2015 daylightSaving" referenceDocumentNumber="ST 2059-2:2015" referenceSectionNumber="6.2.8" />
  <field description="SMPTE SM 2015 daylightSaving" referenceDocumentNumber="ST 2059-2:2015" referenceSectionNumber="6.2.9" />
  <field description="SMPTE SM 2015 daylightSaving" referenceDocumentNumber="ST 2059-2:2015" referenceSectionNumber="6.2.10" />
  <field description="SMPTE SM 2015 daylightSaving" referenceDocumentNumber="ST 2059-2:2015" referenceSectionNumber="6.2.11" />
  <field description="SMPTE SM 2015 leapSecondJump" referenceDocumentNumber="ST 2059-2:2015" referenceSectionNumber="6.2.12" />
  <field description="SMPTE SM 2015 leapSecondJump" referenceDocumentNumber="ST 2059-2:2015" referenceSectionNumber="6.2.13" />
```

Note: Tests and conditions can only refer to the current message, that is, each test applies only to a single message type.

The content of the `field` element is described in detail in "XML Syntax - `field`" below.

3.5 ToD Elements (**testType** = ToD)

The `ccsaTimeInformationMessage`, `ccsaTimeEventMessage`, `g8271TimeEventMessage`, `g8271TimeAnnounceMessage` and `g8271GnssMessage` message types are used to define the tests that will be performed for ToD messages. In some cases, the capture file loaded into PFV will not contain all the fields that can be tested by the rules file. In this case, these rules will be ignored.

- `ccsaTimeInformationMessage`: Defines tests applied to CCSA Time Information messages.
- `ccsaTimeEventMessage`: Defines the tests applied to CCSA Time Event messages.
- `g8271TimeEventMessage`: Defines the tests applied to ITU-T G.8271 Time Event messages.
- `g8271TimeAnnounceMessage`: Defines the tests applied to ITU-T G.8271 Time Announce messages.
- `g8271GnssMessage`: Defines the tests applied to ITU-T G.8271GNSS Status messages.

The content of these elements is described in detail in the sections below.

XML Syntax - `ccsaTimeInformationMessage`

This section contains rules that verify the content of the ToD fields in individual CCSA Time Information messages. The `ccsaTimeInformationMessage` element contains a number of `field` elements; each `field` defines a test to be applied to each individual message in the capture. The `field` element allows tests to be applied to specific messages or to be applied only when other conditions are satisfied. There can be only one `ccsaTimeInformationMessage` in a rules file but there can be as many `field` elements as required.

```
<ccsaTimeInformationMessage>
  <field description="CCSA Time Information Message length is 16" referenceDocumentNumber="CCSA S615-C1324" referenceSectionNumber="2.2">
  <field description="Leap Seconds should be 17 (or greater in the future)" referenceDocumentNumber="CCSA S615-C1324" referenceSectionNumber="2.2">
  <field description="PPS Status is valid from 0x0 to 0x4" referenceDocumentNumber="CCSA S615-C1324" referenceSectionNumber="2.2">
  <field description="Validates the Frame Check Sequence value" referenceDocumentNumber="CCSA S615-C1324" referenceSectionNumber="2.2">
  <field description="Verifies that the timestamps are sequential (once every second)" referenceDocumentNumber="CCSA S615-C1324" referenceSectionNumber="2.2">
</ccsaTimeInformationMessage>
```

Note: Tests and conditions can only refer to the current message, that is, each test applies only to a single message type.

The content of the `field` element is described in detail in "XML Syntax - `field`" below.

XML Syntax – `ccsaTimeEventMessage`

This section contains rules that verify the content of the ToD fields in individual CCSA Time Event messages. The `ccsaTimeEventMessage` element contains a number of `field` elements; each `field` defines a test to be applied to each individual message in the capture. The `field` element allows tests to be applied to specific messages or to be applied only when other conditions are satisfied. There can be only one `ccsaTimeEventMessage` in a rules file but there can be as many `field` elements as required.

```
<ccsaTimeEventMessage>
  <field description="CCSA Time Event Message length is 16" referenceDocumentNumber="CCSA S615-C1324" referenceSectionNumber="2.2">
  <field description="Time Source Status field is valid from 0x0 to 0x5" referenceDocumentNumber="CCSA S615-C1324" referenceSectionNumber="2.2">
  <field description="Time Source Type field is valid from 0x0 to 0x3" referenceDocumentNumber="CCSA S615-C1324" referenceSectionNumber="2.2">
  <field description="Sample alarm check rule" referenceDocumentNumber="CCSA S615-C1324" referenceSectionNumber="2.2">
  <field description="Validates the Frame Check Sequence value" referenceDocumentNumber="CCSA S615-C1324" referenceSectionNumber="2.2">
</ccsaTimeEventMessage>
```

Note: Tests and conditions can only refer to the current message, that is, each test applies only to a single message type.

The content of the `field` element is described in detail in "XML Syntax - `field`" below.

XML Syntax – g8271TimeEventMessage

This section contains rules that verify the content of the ToD fields in individual ITU-T G.8271 Time Event messages. The `g8271TimeEventMessage` element contains a number of `field` elements; each `field` defines a test to be applied to each individual message in the capture. The `field` element allows tests to be applied to specific messages or to be applied only when other conditions are satisfied. There can be only one `g8271TimeEventMessage` in a rules file but there can be as many `field` elements as required.

```
<g8271TimeEventMessage>
  <field description="Time Event Message length is 14" referenceDocumentNumber="ITU-T G.8271/Y.1366" referenceSectionNumber="4.1" />
  <field description="Verifies that the timestamps are sequential (once every second)" referenceDocumentNumber="ITU-T G.8271/Y.1366" referenceSectionNumber="4.1" />
  <field description="CurrentUTCOffset is greater than 36" referenceDocumentNumber="ITU-T G.8271/Y.1366" referenceSectionNumber="4.1" />
  <field description="Validates the Frame Check Sequence value" referenceDocumentNumber="ITU-T G.8271/Y.1366" referenceSectionNumber="4.1" />
</g8271TimeEventMessage>
```

Note: Tests and conditions can only refer to the current message, that is, each test applies only to a single message type.

The content of the `field` element is described in detail in “XML Syntax - `field`” below.

XML Syntax – g8271TimeAnnounceMessage

This section contains rules that verify the content of the ToD fields in individual ITU-T G.8271 Time Announce messages. The `g8271TimeAnnounceMessage` element contains a number of `field` elements; each `field` defines a test to be applied to each individual message in the capture. The `field` element allows tests to be applied to specific messages or to be applied only when other conditions are satisfied. There can be only one `g8271TimeAnnounceMessage` in a rules file but there can be as many `field` elements as required.

```
<g8271TimeAnnounceMessage>
  <field description="Time Announce Message length is 32" referenceDocumentNumber="ITU-T G.8271/Y.1366" referenceSectionNumber="4.1" />
  <field description="versionPTP. Treating as a 1588 Header field. Spec allows 1 or 2. For the purposes of testing, look for 1." referenceDocumentNumber="ITU-T G.8271/Y.1366" referenceSectionNumber="4.1" />
  <field description="domainNumber. Treating as a 1588 Header field." referenceDocumentNumber="1588-2008" referenceSectionNumber="4.1" />
  <field description="domainNumber. Treating as a 1588 Header field." referenceDocumentNumber="1588-2008" referenceSectionNumber="4.1" />
  <field description="sourcePortIdentity.clockIdentity. Treating as a 1588 Header field." referenceDocumentNumber="1588-2008" referenceSectionNumber="4.1" />
  <field description="sourcePortIdentity.portNumber. Treating as a 1588 Header field." referenceDocumentNumber="1588-2008" referenceSectionNumber="4.1" />
  <field description="grandmasterPriority1. Treating as 1588 Announce message field. Disabled - could be any value 0-255" referenceDocumentNumber="1588-2008" referenceSectionNumber="4.1" />
  <field description="grandmasterPriority2. Treating as 1588 Announce message field. Disabled - could be any value 0-255" referenceDocumentNumber="1588-2008" referenceSectionNumber="4.1" />
  <field description="grandmasterClockQuality.ClockClass. Treating as 1588 Announce message field." referenceDocumentNumber="1588-2008" referenceSectionNumber="4.1" />
  <field description="grandmasterClockQuality.ClockAccuracy. Treating as 1588 Announce message field." referenceDocumentNumber="1588-2008" referenceSectionNumber="4.1" />
  <field description="grandmasterClockQuality.offsetScaledLogVariance. Treating as 1588 Announce message field." referenceDocumentNumber="1588-2008" referenceSectionNumber="4.1" />
  <field description="grandmasterClockIdentity. Treating as 1588 Announce message field." referenceDocumentNumber="1588-2008" referenceSectionNumber="4.1" />
  <field description="Validates the Frame Check Sequence value" referenceDocumentNumber="ITU-T G.8271/Y.1366" referenceSectionNumber="4.1" />
</g8271TimeAnnounceMessage>
```

Note: Tests and conditions can only refer to the current message, that is, each test applies only to a single message type.

The content of the `field` element is described in detail in “XML Syntax - `field`” below.

XML Syntax – g8271GnssMessage

This section contains rules that verify the content of the ToD fields in individual ITU-T G.8271 GNSS Status messages. The `g8271GnssMessage` element contains a number of `field` elements; each `field` defines a test to be applied to each individual message in the capture. The `field` element allows tests to be applied to specific messages or to be applied only when other conditions are satisfied. There can be only one `g8271GnssMessage` in a rules file but there can be as many `field` elements as required.

```
<g8271GnssMessage>
  <field description="Gnss Status Message length is 8" referenceDocumentNumber="ITU-T G.8271/Y.1366" referenceSectionNumber="3.1.1" />
  <field description="Time Source Type field is valid from 0x0 to 0x8" referenceDocumentNumber="ITU-T G.8271/Y.1366" referenceSectionNumber="3.1.1" />
  <field description="Time Source Status field is valid from 0x0 to 0x8" referenceDocumentNumber="ITU-T G.8271/Y.1366" referenceSectionNumber="3.1.1" />
  <field description="Validates the Frame Check Sequence value" referenceDocumentNumber="ITU-T G.8271/Y.1366" referenceSectionNumber="3.1.1" />
</g8271GnssMessage>
```

Note: Tests and conditions can only refer to the current message, that is, each test applies only to a single message type.

The content of the `field` element is described in detail in "XML Syntax - `field`" below.

3.6 Common Elements

The xml elements detailed below are applicable to all formats (PTP and ToD).

XML Syntax - field

The `field` element can be used in `ptpHeader`, `announceMessageBody`, `managementMessageBody`, `todMessage`, `ethernetHeader`, `ipv4Header`, `ipv6Header` and `tlvData` elements. The purpose of `field` is to define a test to be performed on a specified field of each message in the capture. When used in `announceMessageBody`, tests are applied only to ANNOUNCE messages. Similarly, when used in `managementMessageBody`, tests are applied only to MANAGEMENT messages.

```
</field>
<field description="PTP Header, messageLength. For the purposes of testing, look for minimum possible"
  <fieldName>messageLength</fieldName>
  <enabled>true</enabled>
  <tests description="Check that all PTP messages are the minimum valid length. No reference - assumed"
    <AUTO>minMessageLength</AUTO>
  </tests>
  <conditions description="Don't check Signaling messages">
    <NOT_EQUAL>
      <fieldName>messageType</fieldName>
      <value>SIGNALING</value>
    </NOT_EQUAL>
  </conditions>
</field>
```

The `field` element contains the following attributes:

- **description**: A string that describes the test (In rules files provided by Calnex, this highlights the relevant content in the source document).
- **referenceDocumentNumber**: The document number on which the test is based. This is intended to refer to the `referenceSource` element in the `testSpecification` header.
- **referenceSectionNumber**: The section within the reference source document that has been used as the basis for the test.

The `fieldName` element specifies the field to be tested. The fields supported are defined in **Enumerated Values** below.

The `enabled` element contains a boolean value (`true`, `false`, `0` or `1`) that determines whether this test will be executed. This allows the test to be defined in the rules file but disabled if required.

The `tests` element defines the test to be applied to the given `fieldName`. A test is considered a pass if the test result is true.

The `conditions` element determines whether the test will be executed based on the content of other fields in the message, for example, you may only want to check that the `alternateMasterFlag` is set for ANNOUNCE, SYNC, FOLLOW_UP and DEL_REQ messages.

The content of the tests and conditions tags are detailed in "XML Syntax – tests" and "XML Syntax – conditions" below.

XML Syntax – tests

The `tests` element can be used as part of the `ptpGeneral`, `ptpHeader`, `announceMessageBody`, `managementMessageBody`, `todMessage`, `ethernetHeader`, `ipv4Header`, `ipv6Header` and `tlvData` elements.

The `tests` element defines the test to be applied to the specified `messageType` (when used in `ptpGeneral`) or to the specified `fieldName` (when used in `ptpHeader`, `announceMessageBody`, `managementMessageBody`, `ethernetHeader`, `ipv4Header` or `tlvData`). A test is considered a pass if the test result is true.

```
<field description="PTP Header, logMessageInterval - check for 0x7F" referenceDocumentNumber="1588-2008">
  <fieldName>logMessageInterval</fieldName>
  <enabled>true</enabled>
  <conditions description="Applies only to specific message types">
    <tests description="Should be 0x7F">
      <EQUAL>
        <value>0x7F</value>
      </EQUAL>
    </tests>
  </field>
```

The `tests` element is made up of one or more comparisons which may be combined using logical **AND** and **OR** operators. An **OR** element evaluates to true if at least one of its child elements evaluates to true; **AND** evaluates to true if all of its child elements are true. Both logical operators support nesting, meaning that it is possible to have any number of **AND** elements inside an **OR** or vice versa.

There are 5 comparison operators (see Comparison Operators, below). They all follow the same general format.

Each operator has a child `value` element. This specifies either a message type (see Message Types, below) or a numeric (hexadecimal or decimal) value, depending on the context. This `value` will be compared against the specified `fieldName`.

```
</field>
<field description="PTP Header, logMessageInterval - check for NOT 0x7F (Announce)" referenceDocumentNumber="1588-2008">
  <fieldName>logMessageInterval</fieldName>
  <enabled>true</enabled>
  <conditions description="Applies only to ANNOUNCE messages">
    <EQUAL>
      <fieldName>messageType</fieldName>
      <value>ANNOUNCE</value>
    </EQUAL>
  </field>
```

If no `fieldName` is given in the comparison element, it is implicitly assumed that the comparison refers to the `fieldName` defined in the `field` containing it.

```
<ptpHeader>
  <!--Decoded PTP version test (2.0 pass only)-->
  <field>
    <fieldName>decodedPTPVersion</fieldName>
    <enabled>true</enabled>
    <tests failMessage="The current ruleset expects v2.0 PTP messages." description="v2.0 PTP messages ex">
  </field>
  <field description="PTP Header, message type" referenceDocumentNumber="1588-2008" referenceSectionNumber="1588-2008">
    <fieldName>messageType</fieldName>
    <!-- messageType = Announce | Sync | Follow_Up | Delay_Req | Delay_Resp | Pdel_Req | Pdel_Resp | Pde
    <enabled>true</enabled>
    <tests description="Only specific message types are allowed">
      <OR>
        <!-- EQUAL with no field element implicitly refers to the current field-->
        <EQUAL>
        <EQUAL>
        <EQUAL>
        <EQUAL>
        <EQUAL>
        <EQUAL>
        <EQUAL>
        <EQUAL>
      </OR>
    </tests>
  </field>
```


FLAG_CHECK is a special case of a comparison operator. Rather than acting on an entire field, it simply checks a single bit in the flag field. **FLAG_CHECK** has two required child elements:

- **value** is the expected Boolean value of the bit check and must be either **true** or **false**.
- **bit** is the bit in the flag field that will be checked (see PTP Header Fields below for available values).

```
<!-- Flag Fields - Octet 0 -->
<field description="PTP Header, flagfield, octet 0: alternateMasterFlag (Announce, Sync, Fup, Del-Resp). Test
and False are both valid" referenceDocumentNumber="1588-2008" referenceSectionNumber="13.3.2.6">
  <fieldName>alternateMasterFlag</fieldName>
  <enabled>>false</enabled>
  <conditions description="Applies only to specific message types">
    <tests description="alternateMasterFlag should be false">
      <FLAG_CHECK>
    </tests>
  </conditions>
</field>
```

The **AUTO** element causes PFV to run a test that cannot be specified using the existing rules file syntax. For example, using **sequenceId** in an **AUTO** element causes PFV to check that message Sequence Ids are correct and sequential.

XML Syntax – **conditions**

The **conditions** element can be used as part of **ptpHeader**, **announceMessageBody**, **todMessage**, **ethernetHeader**, **ipv4Header**, **ipv6Header** and **tlvData** elements. It is evaluated prior to executing a test; if the **conditions** are true, then the test will be executed, otherwise it will not. The **conditions** element has identical syntax to **tests** (with the exception that **AUTO** is not supported).

```
<field description="PTP Header, logMessageInterval - check for NOT 0x7F (Announce)" referenceDocumentNumber="1588-2008">
  <fieldName>logMessageInterval</fieldName>
  <enabled>>true</enabled>
  <conditions description="Applies only to ANNOUNCE messages">
    <EQUAL>
      <fieldName>messageType</fieldName>
      <value>ANNOUNCE</value>
    </EQUAL>
  </conditions>
  <tests description="Should be the message rate (NOT 0x7F)">
    <NOT_EQUAL>
      <value>0x7F</value>
    </NOT_EQUAL>
  </tests>
</field>
```

In the example above, the test on the **logMessageInterval** will only be performed if the **messageType** is equal to **ANNOUNCE**. For **tests**, the **conditions** can be made up of one or more comparisons which can be combined using logical **AND** and **OR** operators.

3.7 Enumerated Values

Message Types

The supported message types are:

- ANNOUNCE
- SYNC and FOLLOW-UP
- DEL-REQ and DEL-Resp
- PDEL-REQ, PDEL-RESP and PDEL-RESP-FUP
- SIGNALING
- MANAGEMENT

Comparison Operators

The supported comparison operators are:

- EQUAL, NOT_EQUAL
- GREATER, LESS
- FLAG_CHECK (for `ptpHeader` and `todMessage` only. Specifies a test on a single flag in a flag field).
- AUTO (for `ptpHeader`, `announceMessageBody`, `managementMessageBody`, `todMessage`, `ethernetHeader`, `ipv4Header` and `tlvData` only. Specifies an automatic test built-in to PFV).

Logical Operators

The supported logical operators are: AND, OR.

Automatic Tests

There are several automatic tests that are not defined by the rules file. These are performed by PFV in all circumstances. These tests are listed below:

For PTP:

- `controlField`: matches the control field content with the associated `messageType`.
- `validSourcePortId`: checks that the `sourcePortId` field is not all zeroes or ones.
- `sequenceId`: checks that the sequence Id increments correctly.
- `minMessageLength`: checks that the length of the message is the minimum possible for the associated `messageType`.
- `EthernetFcs`: checks the ethernet header frame check sequence.
- `Ipv4headerChecksum`
- `displayWithoutTest`: shows the field in PFV GUI under all circumstances without applying any test to it.

For ToD:

- `SequenceCheck`: verifies that timestamps are sequential (once per second).
- `CrcCheck`: verifies the FCS.

PTP Header Fields

The supported PTP header fields are listed in the table below. The offset column indicates the byte offset of the field from the beginning of the PTP header.

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
transportSpecific				messageType				1	0
reservedField0				versionPTP				1	1
messageLength								2	2
domainNumber								1	4
reservedField1								1	5
See flagField below								2	6
correctionField								8	8
reservedField2								4	16
sourcePortIdentity								10	20
sequenceId								2	30
controlField								1	32
logMessageInterval								1	33

Note: The Calnex PFV provides the facility to show the correction field in encoded and in decoded format. The encoded format shows the value of the correction field captured in the packet(s) bytes. The decoded format shows what the encoded value equates to in picoseconds using the calculation specified in IEEE 1588-2019 section 13.3.2.9.

All flagField bits are supported as detailed below:

Octet	Bit	Name
0	0	alternateMasterFlag
0	1	twoStepFlag
0	2	unicastFlag
0	3	reservedOct0Bit3
0	4	reservedOct0Bit4
0	5	ptpProfileSpecific1
0	6	ptpProfileSpecific2
0	7	reservedOct0Bit7
1	0	leap61
1	1	leap59
1	2	currentUtcOffsetValid
1	3	ptpTimescale
1	4	timeTraceable
1	5	frequencyTraceable
1	6	reservedOct1Bit6
1	7	reservedOct1Bit7

Announce Message Body Fields

The supported ANNOUNCE message body fields are listed in the table below. The offset column indicates the byte offset of the field from the beginning of the PTP header.

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
curUtcOffset								2	44
gmPrior1								1	47
gmClkClass (from gmClockQuality)								1	48
gmClkAcc (from gmClockQuality)								1	49
gmClkOslv (from gmClockQuality)								2	50
gmPrior2								1	52

Announce Message TLV Fields

The supported ANNOUNCE message TLV fields are listed in the table below. The only supported ANNOUNCE message TLV is the China Mobile Specification for Ultra Precision Time Synchronization.

Bits	Octets	Offset
tlvType	2	64
tlvLength	2	66
tlvOrganizationId	3	68
tlvOrganizationSubType	3	71
tlvCmcc5GAccuracyLevel1	2	74
tlvCmcc5GStepsRemovedLevel1	2	76
tlvCmcc5GAccuracyLevel2	2	78
tlvCmcc5GStepsRemovedLevel2	2	80
tlvCmcc5GAccuracyLevel3	2	82
tlvCmcc5GStepsRemovedLevel3	2	84

An AUTO is available to ensure that `tlvCmcc5GStepsRemovedLevel1 <= StepsRemoved` from the PTP header, that is `cmccStepsRemovedLevel1Test`.

Management Message Body Fields

The supported MANAGEMENT message body fields are listed in the table below. The offset column indicates the byte offset of the field from the beginning of the PTP header.

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
MgtTargetPortIdentity								10	34
MgtStartingBoundaryHops								1	44
MgtBoundaryHops								1	45
MgtReserved0				MgtActionField				1	46
MgtReserved1								1	47

Note that only .pcap data files and Paragon-neo generated '.CDF' data files will decode the management message body fields.

Ethernet Header Fields

The supported Ethernet Header fields are listed in the table below. The offset column indicates the byte offset of the field from the beginning of the Ethernet header.

Bits	Octets	Offset
Preamble (not tested)	7	0
SFD (not tested)	1	7
ethDestinationAddress	6	8
ethSourceAddress	6	14
VLAN TPID (0x8100 if VLANs in use)	2	16
VLAN TCI (see below)	2	18
ethType	2	16 or 20
ethFCS	4	VLAN and payload dependent

The VLAN TCI structure is:

Octet 18								Octet 19										
7	6	5	4				3	2	1	0	7	6	5	4	3	2	1	0
ethVlanPriorityCodePoint				ethVlanDropEligibleIndicator				ethVlanId										

IPv4 Header Fields

The supported IPv4 Header fields are listed in the table below. The offset column indicates the byte offset of the field from the beginning of the IPv4 header.

Octet n							Octet n+1							Octets	Offset
0	1	2	3	4	5	6	7	0	1	2	3	4	5		
ipv4Version			ipv4Ihl				ipv4Dscp				ipv4Ecn		2	0	
ipv4TotalLength														2	2
ipv4Identification														2	4
ipv4Flags		ipv4FragmentOffset												2	6
ipv4TimeToLive						ipv4Protocol						2	8		
ipv4HeaderChecksum														2	10
Ipv4SourceAddress														4	12
Ipv4DestinationAddress														4	16

IPv6 Header Fields

The supported IPv6 Header fields are listed in the table below. The offset column indicates the byte offset of the field from the beginning of the IPv6 header.

Octet n							Octet n+1							Octets	Offset
0	1	2	3	4	5	6	7	0	1	2	3	4	5		
ipv6Version			ipv6Dscp				ipv6Ecn		Flow label				2	0	
ipv6FlowLabel														2	2
ipv6Payloadlength														2	4
ipv6NextHeader						ipv6HopLimit						2	6		
ipv6SourceAddress														16	8
ipv6DestinationAddress														24	8

TLV Data Fields

PFV is capable of testing many TLV data types (in addition to the ANNOUNCE TLV data fields that are tested under the 'announceMessageBody' section as detailed above). These are listed in the table below, each row contains TLV field names that may be tested for a common TLV type.

TLV Field Name
tlvUnicastRequestSyncType
tlvUnicastRequestSyncLength
tlvUnicastRequestSyncMessageType
tlvUnicastRequestSyncReserved0
tlvUnicastRequestSyncLogInterMessagePeriod
tlvUnicastRequestSyncDurationField
tlvUnicastGrantSyncType
tlvUnicastGrantSyncLength
tlvUnicastGrantSyncMessageType
tlvUnicastGrantSyncReserved0
tlvUnicastGrantSyncLogInterMessagePeriod
tlvUnicastGrantSyncDurationField
tlvUnicastGrantSyncReserved1
tlvUnicastGrantSyncReserved2
tlvUnicastGrantSyncRenewalInvited
tlvUnicastRequestAnnounceType
tlvUnicastRequestAnnounceLength
tlvUnicastRequestAnnounceMessageType
tlvUnicastRequestAnnounceReserved0
tlvUnicastRequestAnnounceLogInterMessagePeriod
tlvUnicastRequestAnnounceDurationField
tlvUnicastGrantAnnounceType
tlvUnicastGrantAnnounceLength
tlvUnicastGrantAnnounceMessageType
tlvUnicastGrantAnnounceReserved0
tlvUnicastGrantAnnounceLogInterMessagePeriod
tlvUnicastGrantAnnounceDurationField

<p>tlvUnicastGrantAnnounceReserved1</p> <p>tlvUnicastGrantAnnounceReserved2</p> <p>tlvUnicastGrantAnnounceRenewalInvited</p>
<p>tlvUnicastRequestDelRespType</p> <p>tlvUnicastRequestDelRespLength</p> <p>tlvUnicastRequestDelRespMessageType</p> <p>tlvUnicastRequestDelRespReserved0</p> <p>tlvUnicastRequestDelRespLogInterMessagePeriod</p> <p>tlvUnicastRequestDelRespDurationField</p>
<p>tlvUnicastGrantDelRespType</p> <p>tlvUnicastGrantDelRespLength</p> <p>tlvUnicastGrantDelRespMessageType</p> <p>tlvUnicastGrantDelRespReserved0</p> <p>tlvUnicastGrantDelRespLogInterMessagePeriod</p> <p>tlvUnicastGrantDelRespDurationField</p> <p>tlvUnicastGrantDelRespReserved1</p> <p>tlvUnicastGrantDelRespReserved2</p> <p>tlvUnicastGrantDelRespRenewalInvited</p>
<p>tlvUnicastRequestPDelRespType</p> <p>tlvUnicastRequestPDelRespLength</p> <p>tlvUnicastRequestPDelRespMessageType</p> <p>tlvUnicastRequestPDelRespReserved0</p> <p>tlvUnicastRequestPDelRespLogInterMessagePeriod</p> <p>tlvUnicastRequestPDelRespDurationField</p>
<p>tlvUnicastGrantDelRespType</p> <p>tlvUnicastGrantDelRespLength</p>
<p>tlvUnicastGrantPDelRespMessageType</p> <p>tlvUnicastGrantPDelRespReserved0</p> <p>tlvUnicastGrantPDelRespLogInterMessagePeriod</p>

<ul style="list-style-type: none"> tlvUnicastGrantPDelRespDurationField tlvUnicastGrantPDelRespReserved1 tlvUnicastGrantPDelRespReserved2 tlvUnicastGrantPDelRespRenewalInvited
<ul style="list-style-type: none"> Tlv8021ASFollowUp_Type Tlv8021ASFollowUp_Length Tlv8021ASFollowUp_OrgId Tlv8021ASFollowUp_OrgSubType Tlv8021ASFollowUp_CSRO Tlv8021ASFollowUp_GmTimeBaseInd Tlv8021ASFollowUp_LastGmPhaseChange Tlv8021ASFollowUp_ScaledLastGmFreqChange
<ul style="list-style-type: none"> Tlv8021ASMsgIntvlReq_Type Tlv8021ASMsgIntvlReq_Length Tlv8021ASMsgIntvlReq_OrgId Tlv8021ASMsgIntvlReq_OrgSubType Tlv8021ASMsgIntvlReq_LogLinkDelayIntvl Tlv8021ASMsgIntvlReq_LogTimeSyncIntvl Tlv8021ASMsgIntvlReq_LogAnnounceIntvl Tlv8021ASMsgIntvlReq_Flags Tlv8021ASMsgIntvlReq_Reserved0
<ul style="list-style-type: none"> Tlv8021ASGtpCapable_Type Tlv8021ASGtpCapable_Length Tlv8021ASGtpCapable_OrgId Tlv8021ASGtpCapable_OrgSubType Tlv8021ASGtpCapable_LogGtpCapMsgIntvl Tlv8021ASGtpCapable_Flags Tlv8021ASGtpCapable_Reserved0
<ul style="list-style-type: none"> Tlv8021ASGtpCapMsgIntvlReq_Type Tlv8021ASGtpCapMsgIntvlReq_Length

<p>Tlv8021ASGtpCapMsgIntvlReq_OrgId</p> <p>Tlv8021ASGtpCapMsgIntvlReq_OrgSubType</p> <p>Tlv8021ASGtpCapMsgIntvlReq_LogGtpCapMsgIntvl</p> <p>Tlv8021ASGtpCapMsgIntvlReq_Reserved0</p>
<p>TlvPathTrace_Type</p> <p>TlvPathTrace_Length</p> <p>TlvPathTrace_PathSequence</p>
<p>TlvC37238_2011_Type</p> <p>TlvC37238_2011_Length</p> <p>TlvC37238_2011_OrgId</p> <p>TlvC37238_2011_OrgSubType</p> <p>TlvC37238_2011_GrandmasterId</p> <p>TlvC37238_2011_GrandmasterTimeInaccuracy</p> <p>TlvC37238_2011_NetworkTimeInaccuracy</p> <p>TlvC37238_2011_Reserved</p>
<p>TlvC37238_2017_Type</p> <p>TlvC37238_2017_Length</p> <p>TlvC37238_2017_OrgId</p> <p>TlvC37238_2017_OrgSubType</p> <p>TlvC37238_2017_GrandmasterId</p> <p>TlvC37238_2017_Reserved1</p> <p>TlvC37238_2017_TotalTimeInaccuracy</p> <p>TlvC37238_2017_Reserved2</p>
<p>TlvATOI_Type</p> <p>TlvATOI_Length</p> <p>TlvATOI_KeyField</p> <p>TlvATOI_CurrentOffset</p> <p>TlvATOI_JumpSeconds</p> <p>TlvATOI_TimeOfNextJump</p> <p>TlvATOI_DisplayName</p>

TlvATOI_Pad

TlvSmpteSm_2015_Type

TlvSmpteSm_2015_Length

TlvSmpteSm_2015_OrgId

TlvSmpteSm_2015_OrgSubType

TlvSmpteSm_2015_DefaultSystemFrameRate

TlvSmpteSm_2015_DefaultSystemFrameRateNumerator

TlvSmpteSm_2015_DefaultSystemFrameRateDenominator

TlvSmpteSm_2015_MasterLockingStatus

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

CCSA Time Information Message Fields

The supported CCSA Information Message fields are listed in the table below. The offset column indicates the byte offset of the field from the beginning of the message.

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
Sync char 1 (not tested)								1	0
Sync char 2 (not tested)								1	1
Class (not tested)								1	2
Id (not tested)								1	3
Length								2	4
TimeOfWeek								4	6
Reserved1								4	10
Week								2	14
LeapS								1	16
PPSStatus								1	17
TAcc								1	18
Reserved2								1	19
Reserved3								1	20
Reserved4								1	21
FCS								1	22

In addition to the above, UTCTime is a calculated field that can be used to verify a combination of TimeOfWeek and Week.

CCSA Time Event Message Fields

The supported CCSA Event Message fields are listed in the table below. The offset column indicates the byte offset of the field from the beginning of the message.

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
Sync char 1 (not tested)								1	0
Sync char 2 (not tested)								1	1
Class (not tested)								1	2
Id (not tested)								1	3
Length (not tested)								2	4
TimeSourceType								1	6
TimeSourceStatus								1	7
Alarm (also see table below)								2	8
Reserved1								1	10
Reserved2								1	18
Reserved3								1	19
Reserved4								4	20
Reserved5								4	24
FCS								1	28

All Monitor Alarm (Alarm) bits are supported as detailed below:

Octet	Bit	Name
0	0	ReservedOct0Bit0
0	1	AntennaOpen
0	2	AntennaShorted
0	3	NotTrackingSatellites
0	4	ReservedOct0Bit4
0	5	SurveyInProgress
0	6	NoStoredPosition
0	7	LeapSecondPending
1	0	InTestMode
1	1	PositionIsQuestionable
1	2	ReservedOct1Bit2
1	3	AlmanacNotComplete
1	4	PPSGenerated
1	5	ReservedOct1Bit5
1	6	ReservedOct1Bit6
1	7	ReservedOct1Bit7

G.8271 Time Event Message Fields

The supported ITU-T G.8271 Time Event message fields are listed in the table below. The offset column indicates the byte offset of the field from the beginning of the message.

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
Sync char 1 (not tested)								1	0
Sync char 2 (not tested)								1	1
Class (not tested)								1	2
Id (not tested)								1	3
Length								2	4
UTCTime								6	6
Reserved1								1	12
Flags								1	13
CurrentUTCOffset								2	14
Reserved2								4	16
FCS								1	20

The Flags field is supported as detailed below:

Bit	Name
0	Leap61
1	Leap59
2	UtcOffsetValid
3	ReservedBit3
4	TimeTraceable
5	FrequencyTraceable
6	ReservedBit6
7	ReservedBit7

G.8271 Time Announce Message Fields

The supported ITU-T G.8271 Time Announce message fields are listed in the table below. The offset column indicates the byte offset of the field from the beginning of the message.

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
Sync char 1 (not tested)								1	0
Sync char 2 (not tested)								1	1
Class (not tested)								1	2
Id (not tested)								1	3
Length								2	4
versionPTP								1	6
domainNumber								1	7
flagField								2	8
sourceClockIdentity								8	10
sourcePortNumber								2	18
grandmasterPriority1								1	20
grandmasterPriority2								1	21
clockClass								1	22
clockAccuracy								1	23
offsetScaledLogVariance								2	24
grandmasterClockIdentity								8	26
stepsRemoved								2	34
timeSource								1	36
Reserved0								1	37
FCS								1	38

G.8271 GNSS Status Message Fields

The supported ITU-T G.8271 GNSS Status message fields are listed in the table below. The offset column indicates the byte offset of the field from the beginning of the message.

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
Sync char 1 (not tested)								1	0
Sync char 2 (not tested)								1	1
Class (not tested)								1	2
Id (not tested)								1	3
Length								2	4
TimeSourceType								1	6
TimeSourceStatus								1	7
Alarms								2	8
Reserved0								4	10
FCS								1	14

The Alarms field is supported as detailed below:

Octet	Bit	Name
0	0	ReservedOct0Bit0
0	1	AntennaOpen
0	2	AntennaShorted
0	3	NotTrackingSatellites
0	4	ReservedOct0Bit4
0	5	SurveyInProgress
0	6	NoStoredPosition
0	7	LeapSecondPending
1	0	InTestMode
1	1	Uncertain
1	2	ReservedOct1Bit2
1	3	AlmanacNotComplete
1	4	PPSGenerated
1	5	ReservedOct1Bit5
1	6	ReservedOct1Bit6
1	7	ReservedOct1Bit7

3.8 Editing a Rules File and the XML Schema

An XML schema is a description of an XML document, typically expressed in terms of constraints on the structure and content of the document above and beyond the basic syntax constraints imposed by XML itself. These constraints are generally expressed using some combination of grammatical rules governing the order of elements, boolean predicates that the content must satisfy, data types governing the content of elements and attributes, and more specialized rules such as uniqueness constraints.

The PFV application is delivered with a schema file (Rules Validation.xsd) for the XML rules files. This is used by PFV to validate rules files when they are loaded and, if an error is found, provides some information to help resolve the issue. The .xsd file used by PFV for this purpose is located in the same folder as the pre-defined rules files.

The schema file is also located in the user-defined rules folder. This allows schema-aware editors to be used for editing rules files. Schema-aware editors can provide error checking of rules files prior to the file being loaded into PFV. Some of these editors may also automatically suggest the correct syntax and structure while editing.



Calnex Solutions plc
Oracle Campus
Linlithgow EH49 7LR
United Kingdom

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

calnexsol.com

© Calnex Solutions, 2024

This information is subject
to change without notice.

Apr 2024