

Calnex PFV - Field Verifier



The screenshot displays the Calnex PFV Field Verifier interface. The main window shows a list of captured packets with columns for Packet #, Arrival Time, Message Type, and various protocol fields. The summary panel at the bottom provides a breakdown of message rates and counts for different message types.

Packet #	Arrival Time	Message Type	transportSpecific	versionPTP	reservedField0	messageLength	domainNumber	reservedField1
344	6.043408440	ANNOUNCE	0x0	0x2	0x0	0x40	0x18	0x0
345	6.062509000	DEL-REQ	0x0	0x2	0x0	0x2c	0x18	0x0
346	6.062776680	DEL-RESP	0x0	0x2	0x0	0x36	0x18	0x0
347	6.091444810	SYNC	0x0	0x2	0x0	0x2c	0x18	0x0
348	6.125000000	DEL-REQ	0x0	0x2	0x0	0x2c	0x18	0x0
349	6.125237240	DEL-RESP	0x0	0x2	0x0	0x36	0x18	0x0
350	6.134441800	PDEL-REQ	0x4	0x2	0x4	0x0	0x0	0x2
351	6.152445735	SYNC	0x0	0x2	0x0	0x2c	0x18	0x0
352	6.171428300	ANNOUNCE	0x0	0x2	0x0	0x40	0x18	0x0
353	6.187500000	DEL-REQ	0x0	0x2	0x0	0x2c	0x18	0x0
354	6.187783915	DEL-RESP	0x0	0x2	0x0	0x36	0x18	0x0
355	6.218852120	PDEL-RESP	0x0	0x0	0x0	0x97	0x0	0x1
356	6.218450680	SYNC	0x0	0x2	0x0	0x2c	0x18	0x0
357	6.250000000	DEL-REQ	0x0	0x2	0x0	0x2c	0x18	0x0
358	6.250209750	DEL-RESP	0x0	0x2	0x0	0x36	0x18	0x0
359	6.287428690	SYNC	0x0	0x2	0x0	0x2c	0x18	0x0
360	6.299419610	ANNOUNCE	0x0	0x2	0x0	0x40	0x18	0x0
361	6.312500000	DEL-REQ	0x0	0x2	0x0	0x2c	0x18	0x0
362	6.312780120	DEL-RESP	0x0	0x2	0x0	0x36	0x18	0x0
363	6.351443800	SYNC	0x0	0x2	0x0	0x2c	0x18	0x0
364	6.372000000	DEL-REQ	0x0	0x2	0x0	0x2c	0x18	0x0
365	6.373236655	DEL-RESP	0x0	0x2	0x0	0x36	0x18	0x0
366	6.415443545	SYNC	0x0	0x2	0x0	0x2c	0x18	0x0
367	6.435406310	ANNOUNCE	0x0	0x2	0x0	0x40	0x18	0x0

Message Type	Average Message Rate (msg/sec)	Total Counts
SYNC	15.47	53351
FOLLOW-UP	N/A	0
DEL-REQ	16.03	1683
ANNOUNCE	7.75	0
PDEL-REQ	0.50	0
PDEL-RESP-FUP	1.25	0

Summary: Total Pass Rate: 26.84% (FAIL)

GETTING STARTED GUIDE

Table of Contents

1	Introduction	4
2	Using the PTP Field Verifier	5
2.1	<i>Launching the Application.....</i>	5
	From Paragon-X.....	5
	From P100G	5
	From the Start Menu.....	5
2.2	<i>Loading a File</i>	6
2.3	<i>Rules Selection</i>	7
2.4	<i>The Main Display.....</i>	7
2.5	<i>Column Organisation</i>	9
2.6	<i>Report Generation.....</i>	10
2.7	<i>Loading a Licence File.....</i>	11
3	Customising the Rules File	13
	Accessing the P100G Filesystem	13
3.1	<i>File Locations.....</i>	13
3.2	<i>XML Syntax – Top-Level Structure.....</i>	14
3.3	<i>PTP Elements (testType = PTP).....</i>	16
	XML Syntax - ptpGeneral.....	16
	XML Syntax - ptpHeader	17
	XML Syntax – announceMessageBody.....	18
	XML Syntax – announceMessageCMCC5GTLV.....	18
	XML Syntax – ethernetHeader	19
	XML Syntax – ipv4Header	19
	XML Syntax – ipv6Header	19
3.4	<i>ToD Elements (testType = ToD)</i>	21
	XML Syntax - ccsaTimeInformationMessage.....	21
	XML Syntax – ccsaTimeEventMessage	21
	XML Syntax – g8271TimeEventMessage	21
	XML Syntax – g8271TimeAnnounceMessage.....	22
	XML Syntax – g8271GnssMessage	22
3.5	<i>Common Elements</i>	24
	XML Syntax - field.....	24
	XML Syntax – tests	24
	XML Syntax – conditions	26
3.6	<i>Enumerated Values.....</i>	27
	Message Types.....	27
	Comparison Operators.....	27
	Logical Operators	27
	Automatic tests	27
	PTP Header Fields.....	27
	Announce Message Body Fields.....	28
	Announce Message TLV Fields	28
	Ethernet Header Fields	29
	IPv4 Header Fields.....	29
	IPv6 Header Fields.....	30
	CCSA Time Information Message Fields.....	30

CCSA Time Event Message Fields	30
G.8271 Time Event Message Fields	31
G.8271 Time Announce Message Fields	32
G.8271 GNSS Status Message Fields	32
3.7 Editing a Rules File and the XML Schema.....	34

1 Introduction

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

The PFV is available as an option with Paragon-X and P100G. In addition, it is available as a stand-alone application. The stand-alone application is able to process .pcap files.

The PFV takes as input a capture (.cpd file captured using Paragon-X or a .pcap) and verifies the fields of PTP/ToD messages in the capture against a selected set of rules, with clear Pass/Fail indications.

Calnex provide rules for the following standards: IEEE1588, ITU-T G.8265.1, ITU-T G.8275.1, G.8275.2, IEEE802.1AS (gPTP), IEEE C 37.238 and (for ToD) CCSA and ITU-T G.8271.

Rules are defined using xml. This provides an easy way for the user to supply their own custom rules that can be applied in addition to pre-defined versions. All PTP header fields and key Announce message body fields can be verified. The xml definitions provide an extremely flexible mechanism for checking the contents of PTP and ToD message fields.

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

Direction	Packet#	Arrival Time	Message Type	PTP Body Fields		
				eld	controlField	logMessageInterval
←	91	1.381374565	DEL-REQ	0x1	127	origTstamp= 2013 312 06:21:28.811411025
→	92	1.381892960	DEL-RESP	0x3	127	recvTstamp= 2013 312 06:21:28.811412455 reqPortId:clkId=0x2 r
→	93	1.410024220	SYNC	0x0	127	origTstamp= 2013 312 06:21:28.840060336
→	94	1.440009925	SYNC	0x0	127	origTstamp= 2013 312 06:21:28.870046040
←	95	1.443874565	DEL-REQ	0x1	127	origTstamp= 2013 312 06:21:28.873911025
→	96	1.444348360	DEL-RESP	0x3	127	recvTstamp= 2013 312 06:21:28.873912455 reqPortId:clkId=0x2 r
→	97	1.470118740	SYNC	0x0	127	origTstamp= 2013 312 06:21:28.900154856
→	98	1.470274120	ANNOUNCE	0x5	1	origTstamp= 2013 312 06:21:28.899950520 curUtcOffset=35 gmPric
←	99	1.479832005	SIGNALING	0x5	127	tgtPortId:clkId=0x2 origTstamp= 2013 312 06:21:28.899950520
→	100	1.480273335	ANNOUNCE	0x5	1	origTstamp= 2013 312 06:21:28.899950520 curUtcOffset=35 (ERROR)
→	101	1.480306015	SIGNALING	0x5	127	tgtPortId:clkId=0x2 gmPrior1=0
←	102	1.480436805	SIGNALING	0x5	127	tgtPortId:clkId=0x2 gmClkClass=80 (ERROR)
→	103	1.480804975	SYNC	0x0	127	origTstamp= 2013 312 06:21:28.900154856 gmClkCAcc=0x20
←	104	1.480812805	SIGNALING	0x5	127	origTstamp= 2013 312 06:21:28.900154856 gmClkOslv=0x0
→	105	1.480837960	SIGNALING	0x5	127	tgtPortId:clkId=0x2 gmPrior2=0
→	106	1.481139485	SIGNALING	0x5	127	tgtPortId:clkId=0x2 tgtPortId:port#=1
→	107	1.502429175	SYNC	0x0	127	origTstamp= 2013 312 06:21:28.932465288
←	108	1.506374565	DEL-REQ	0x1	127	origTstamp= 2013 312 06:21:28.936411025
→	109	1.507189375	DEL-RESP	0x3	127	recvTstamp= 2013 312 06:21:28.936412455 reqPortId:clkId=0x2 r
→	110	1.530179415	SYNC	0x0	127	origTstamp= 2013 312 06:21:28.960215528
→	111	1.562131870	SYNC	0x0	127	origTstamp= 2013 312 06:21:28.992167984
←	112	1.568874565	DEL-REQ	0x1	127	origTstamp= 2013 312 06:21:28.998911025
→	113	1.570817805	DEL-RESP	0x3	127	recvTstamp= 2013 312 06:21:28.998912455 reqPortId:clkId=0x2 r
→	114	1.590147630	SYNC	0x0	127	origTstamp= 2013 312 06:21:29.020183744
→	115	1.621819580	SYNC	0x0	127	origTstamp= 2013 312 06:21:29.051855696
←	116	1.631374565	DEL-REQ	0x1	127	origTstamp= 2013 312 06:21:29.061411025

Calnex Average Message Rate (msg/sec)

SYNC	33.35	FOLLOW-UP	N/A	DEL-REQ	16.00	Total Counts	Packets	4656
DEL-RESP	16.00	ANNOUNCE	0.51	SIGNALING	3824.1	Errored Packets	36	

FAIL
 Total Pass Rate: 99.22%

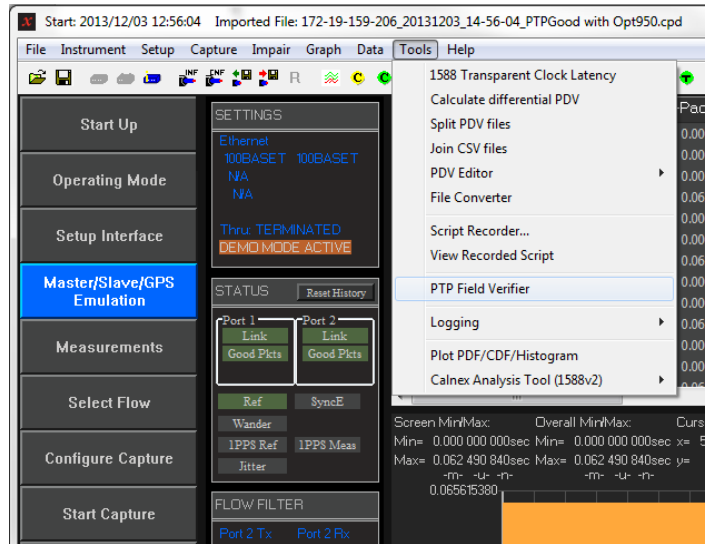
In addition, the PFV is able to generate a report (in PDF format) that documents the overall pass/fail status along with details of any failures.

2 Using the PTP Field Verifier

2.1 Launching the Application

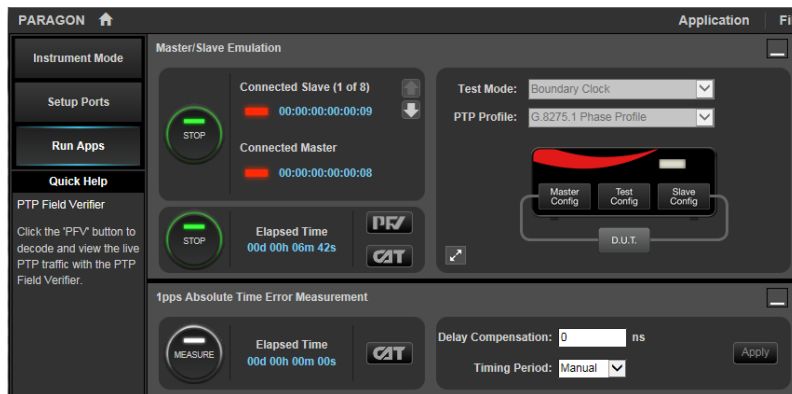
From Paragon-X

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



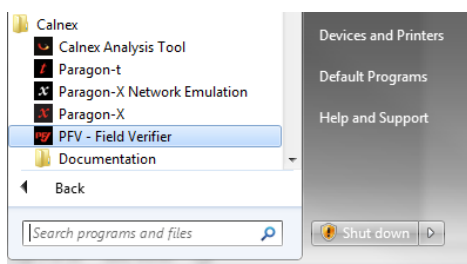
From P100G

The PFV can be launched from P100G using the **PFV** button in the **Master Slave Emulation** application:



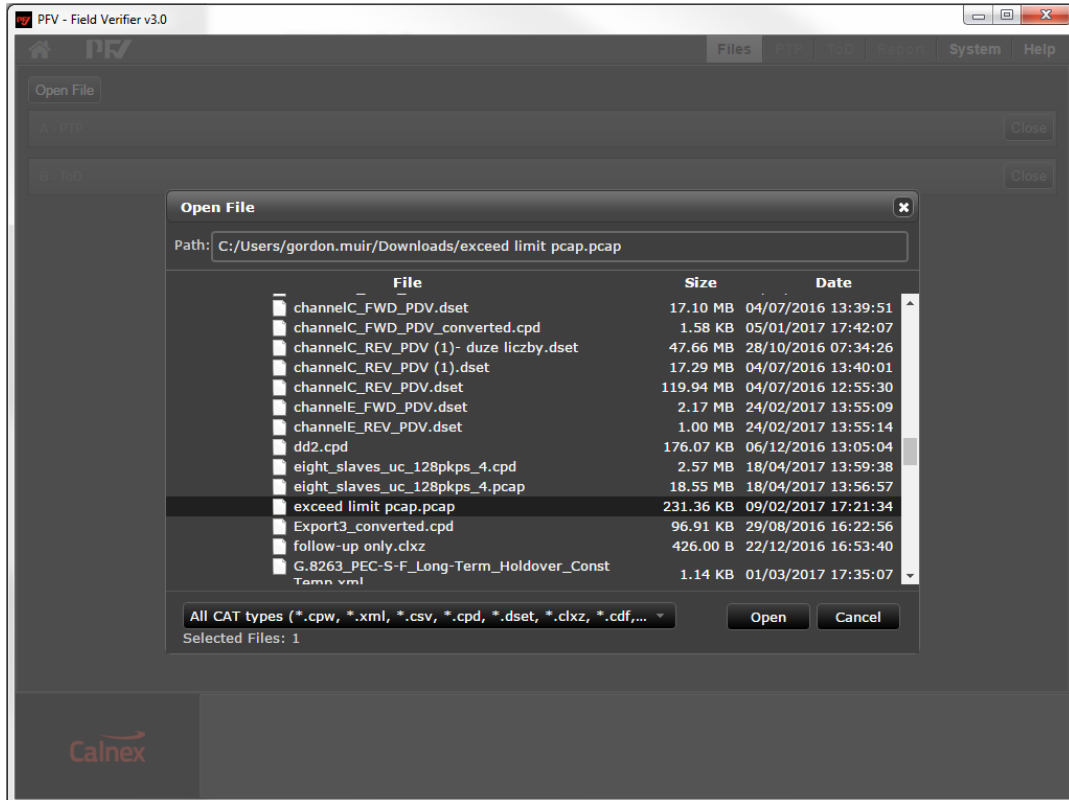
From the Start Menu

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

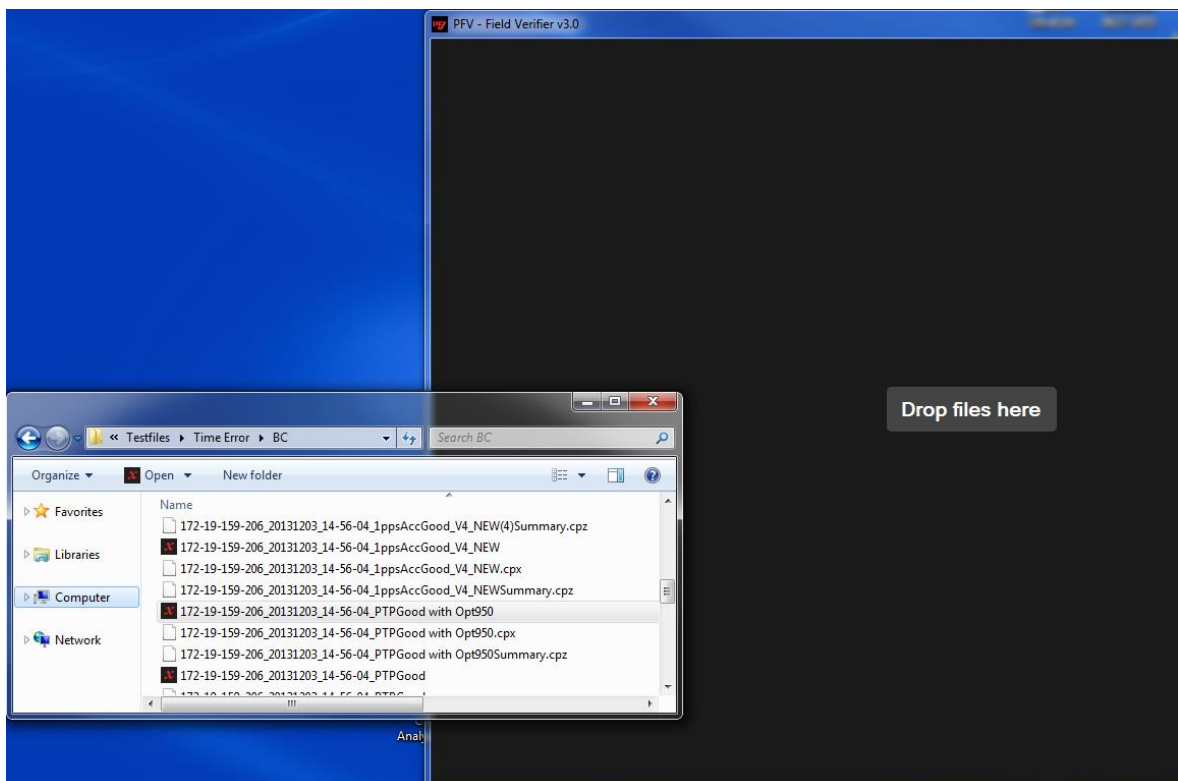


2.2 Loading a File

When launched from the Paragon-X application or P100G, the PFV will automatically load any current PTP or CCSA capture file. If there is no existing capture or the PFV has been launched directly, then a capture file (.cpd or .pcap) can be loaded using the **Files** menu:

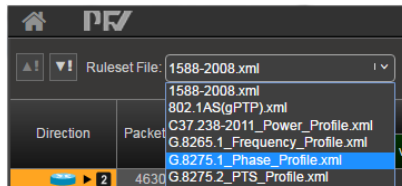


Alternatively, if using the stand-alone version, a file can be loaded using drag and drop:



2.3 Rules Selection

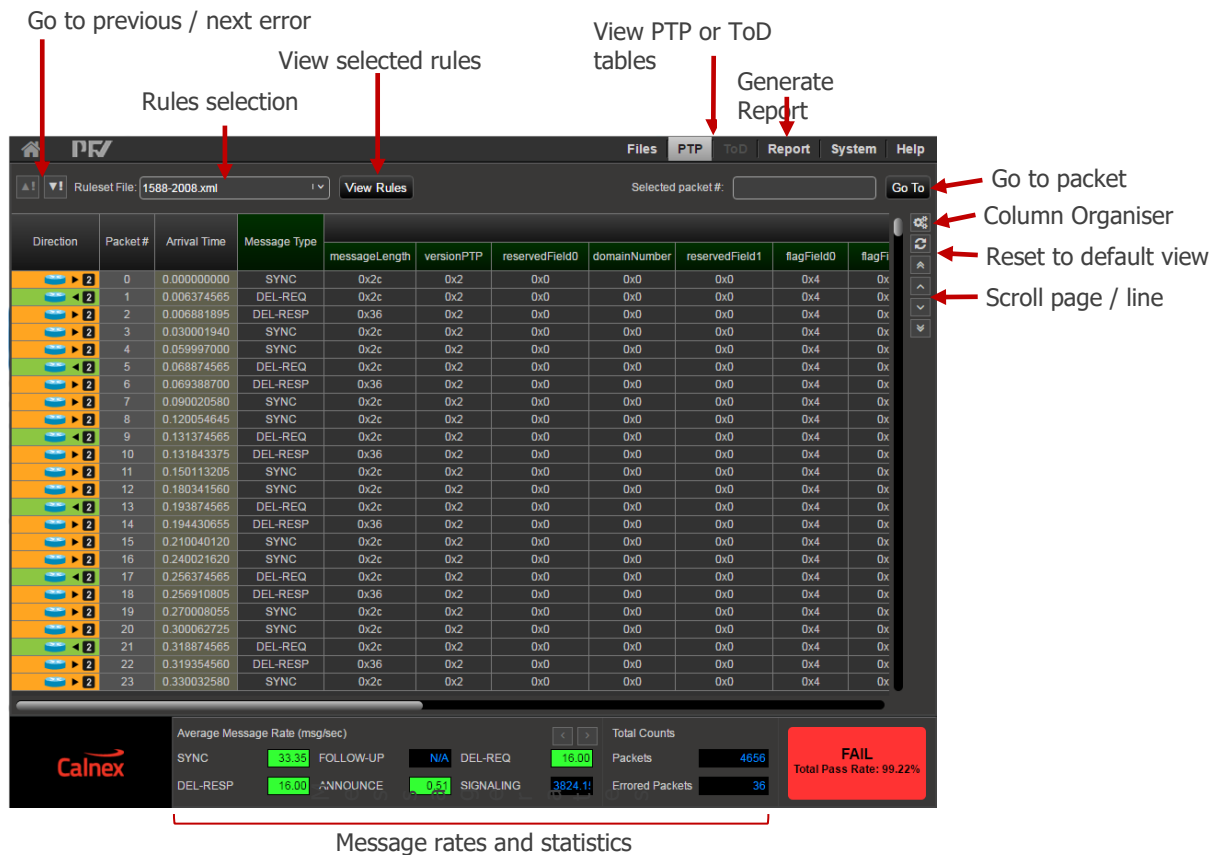
Rules files can be easily selected from the pull-down menu. Changing the rules file will cause the capture to be re-analysed (note that this may take some time for large captures).



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

2.4 The Main Display

The main display of the PFV is shown below:



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 may be seen by hovering over the field:

PFV - Field Verifier v3.0

Files PTP ToD Report System Help

Ruleset File: 1588-2008.xml View Rules Selected packet #: 100 Go To

Direction	Packet #	Arrival Time	Message Type	Field	controlField	logMessageInterval	PTP Body Fields
←	91	1.381374565	DEL-REQ		0x1	127	origTstamp= 2013 312 06:21:28.811411025
→	92	1.381892960	DEL-RESP		0x3	127	recvTstamp= 2013 312 06:21:28.811412455 reqPortId:clkId=0x2 r
→	93	1.410024220	SYNC		0x0	127	origTstamp= 2013 312 06:21:28.840060336
→	94	1.440009925	SYNC		0x0	127	origTstamp= 2013 312 06:21:28.870046040
←	95	1.443874565	DEL-REQ		0x1	127	origTstamp= 2013 312 06:21:28.873911025
→	96	1.444348360	DEL-RESP		0x3	127	recvTstamp= 2013 312 06:21:28.873912455 reqPortId:clkId=0x2 r
→	97	1.470118740	SYNC		0x0	127	origTstamp= 2013 312 06:21:28.900154856
→	98	1.470274120	ANNOUNCE		0x5	1	origTstamp= 2013 312 06:21:28.899950520 curUtcOffset=35 gmPrio
←	99	1.479832005	SIGNALING		0x5	127	tgtPortId:clkId=0x2 origTstamp= 2013 312 06:21:28.899950520
→	100	1.480273335	ANNOUNCE		0x5	1	origTstamp= 2013 312 06:21:28.900154856 curUtcOffset=35 (ERROR)
→	101	1.480306015	SIGNALING		0x5	127	origTstamp= 2013 312 06:21:28.900154856 gmPrior1=0
←	102	1.480436805	SIGNALING		0x5	127	origTstamp= 2013 312 06:21:28.900154856 gmClkClass=80 (ERROR)
→	103	1.480804975	SYNC		0x0	127	origTstamp= 2013 312 06:21:28.900154856 gmClkCAcc=0x20
←	104	1.480812805	SIGNALING		0x5	127	origTstamp= 2013 312 06:21:28.900154856 gmClkOslv=0x0
→	105	1.480837960	SIGNALING		0x5	127	origTstamp= 2013 312 06:21:28.900154856 gmPrior2=0
→	106	1.481139485	SIGNALING		0x5	127	origTstamp= 2013 312 06:21:28.900154856 tgtPortId:port#=1
→	107	1.502429175	SYNC		0x0	127	origTstamp= 2013 312 06:21:28.932465288
←	108	1.506374565	DEL-REQ		0x1	127	origTstamp= 2013 312 06:21:28.936411025
→	109	1.507189375	DEL-RESP		0x3	127	recvTstamp= 2013 312 06:21:28.936412455 reqPortId:clkId=0x2 r
→	110	1.530179415	SYNC		0x0	127	origTstamp= 2013 312 06:21:28.960215528
→	111	1.562131870	SYNC		0x0	127	origTstamp= 2013 312 06:21:28.992167984
←	112	1.568874565	DEL-REQ		0x1	127	origTstamp= 2013 312 06:21:28.998911025
→	113	1.570817805	DEL-RESP		0x3	127	recvTstamp= 2013 312 06:21:28.998912455 reqPortId:clkId=0x2 r
→	114	1.590147630	SYNC		0x0	127	origTstamp= 2013 312 06:21:29.020183744
→	115	1.621819580	SYNC		0x0	127	origTstamp= 2013 312 06:21:29.051855696
←	116	1.631374565	DEL-REQ		0x1	127	origTstamp= 2013 312 06:21:29.061411025

Average Message Rate (msg/sec)

SYNC	33.35	FOLLOW-UP	N/A	DEL-REQ	16.00	Total Counts	
DEL-RESP	16.00	ANNOUNCE	0.51	SIGNALING	3824.1	Packets	4656
						Errored Packets	36

FAIL
Total Pass Rate: 99.22%

In the Message Rates and Statistics section, 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.

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

The percentage number of passing packets and an overall Pass/Fail are clearly indicated.

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

PFV Files PTP ToD Report System Help

Ruleset File: CCSA_ToD.xml View Rules Selected packet #: 13 Go To

Sample #	CCSA Information Message										CCSA Event Message				
	Length	UTC Time	Time of Week	Reserved0	Week	LeapS	PPS Status	TAcc	Reserved1	Reserved2	Reserved3	FCS	Time Source Type	Time Source Status	Alarms
24	16	2014-07-02 11:17:21	299857	0	1799	16	Stratum 1 Holdover	10	0	0	0	8	GPS	3D-6x	Antenn...
25	16	2014-07-02 11:17:22	299858	0	1799	16	Stratum 1 Holdover	10	0	0	0	248	GPS	3D-6x	Antenn...
26	16	2014-07-02 11:17:23	299859	0	1799	16	Stratum 1 Holdover	10	0	0	0	95	GPS	3D-6x	Antenn...
27	16	2014-07-02 11:17:24	299860	0	1799	16	Stratum 1 Holdover	10	0	0	0	1	GPS	3D-6x	Antenn...
28	16	2014-07-02 11:17:25	299861	0	1799	16	Stratum 1 Holdover	10	0	0	0	166	GPS	3D-6x	Antenn...
29	16	2014-07-02 11:17:26	299862	0	1799	16	Stratum 1 Holdover	10	0	0	0	86	GPS	3D-6x	Antenn...
30	16	2014-07-02 11:17:27	299864	0	1799	17	Stratum 1 Holdover	10	0	0	0	221	GPS	3D-6x	Antenn...
31	16	2014-07-02 11:17:28	299865	0	1799	17	Stratum 1 Holdover	10	0	0	0	122	GPS	3D-6x	Antenn...
32	16	2014-07-02 11:17:29	299866	0	1799	17	Stratum 3 Holdover	10	0	0	0	9	GPS	3D-6x	Antenn...
33	16	2014-07-02 11:17:30	299867	0	1799	17	Stratum 3 Holdover	10	0	0	0	174	GPS	3D-6x	Antenn...
34	16	2014-07-02 11:17:31	299868	0	1799	17	Stratum 3 Holdover	10	0	0	0	240	GPS	3D-6x	Antenn...
35	16	2014-07-02 11:17:32	299869	0	1799	17	Stratum 3 Holdover	10	0	0	0	87	GPS	3D-6x	Antenn...
36	16	2014-07-02 11:17:33	299870	0	1799	17	Stratum 3 Holdover	22	0	0	0	141	GPS	3D-6x	Antenn...
37	16	2014-07-02 11:17:34	299871	0	1799	17	Stratum 3 Holdover	22	0	0	0	42	GPS	3D-6x	Antenn...
38	16	2014-07-02 11:17:35	299872	0	1799	17	Stratum 3 Holdover	22	0	0	0	182	GPS	3D-6x	Antenn...
39	16	2014-07-02 11:17:36	299873	0	1799	17	Stratum 3 Holdover	22	0	0	0	17	GPS	3D-6x	Antenn...
40	16	2014-07-02 11:17:37	299874	0	1799	17	Stratum 3 Holdover	22	0	0	0	225	GPS	3D-6x	Antenn...
41	16	2014-07-02 11:17:38	299875	0	1799	17	Stratum 3 Holdover	22	0	0	0	70	GPS	3D-6x	Antenn...
42	16	2014-07-02 11:17:39	299876	0	1799	17	Stratum 3 Holdover	22	0	0	0	24	GPS	3D-6x	Antenn...
43	16	2014-07-02 11:17:40	299877	0	1799	17	Stratum 3 Holdover	22	0	0	0	191	GPS	3D-6x	Antenn...
44	16	2014-07-02 11:17:41	299878	0	1799	17	Stratum 3 Holdover	22	0	0	0	79	GPS	3D-6x	Antenn...
45	16	2014-07-02 11:17:42	299879	0	1799	17	Stratum 3 Holdover	22	0	0	0	232	GPS	3D-6x	Antenn...
46	16	2014-07-02 11:17:43	299880	0	1799	17	Stratum 3 Holdover	22	0	0	0	243	GPS	3D-6x	Antenn...
47	16	2014-07-02 11:17:44	299881	0	1799	17	Stratum 3 Holdover	22	0	0	0	84	GPS	3D-6x	Antenn...

Total Counts
Packets: 124
Errored Packets: 30
FAIL
Total Pass Rate: 75.80%

When analysing a pcap file, the packet headers are also available:


Packet #	Arrival Time	Message Type	Ethernet Header				IPv4 Header													
			DestinationAddress	SourceAddress	EtherType	FCS	Version	IHL	DSCP	ECN	Total Len.	Identifcat.	Flags	Fragment.	TTL	Protocol	Header Chec.	Source IP	Destination IP	transportSp
351	0.000000000	SYNC	0:10:ec:80:e7:8b	0:11:12:b6:6b:e2	0x800	0x964b7b9	4	5	46	0	72	0xc7c0	0x0	0	58	17	0xad7	192.168.100.2	192.168.200.2	0x0
352	0.000000000	DEL_REQ	0:11:12:b6:6b:e2	0:10:ec:80:e7:8b	0x800	0xaa2a9509b	4	5	46	0	72	0x782b	0x0	0	64	17	0x546c	192.168.200.2	192.168.100.2	0x0
353	0.000000000	DEL_RESP	0:10:ec:80:e7:8b	0:11:12:b6:6b:e2	0x800	0x37a8e503	4	5	0	0	82	0xc7c1	0x0	0	58	17	0xb84	192.168.100.2	192.168.200.2	0x0
354	0.000000000	SYNC	0:10:ec:80:e7:8b	0:11:12:b6:6b:e2	0x800	0x793e2a02	4	5	46	0	72	0xc7c2	0x0	0	58	17	0xad5	192.168.100.2	192.168.200.2	0x0
355	0.000000000	DEL_REQ	0:11:12:b6:6b:e2	0:10:ec:80:e7:8b	0x800	0x1f5b1759	4	5	46	0	72	0x782c	0x0	0	64	17	0x546b	192.168.200.2	192.168.100.2	0x0
356	0.000000000	DEL_RESP	0:10:ec:80:e7:8b	0:11:12:b6:6b:e2	0x800	0x1055115	4	5	0	0	82	0xc7c3	0x0	0	58	17	0xb82	192.168.100.2	192.168.200.2	0x0
357	0.000000000	ANNOUNCE	0:10:ec:80:e7:8b	0:11:12:b6:6b:e2	0x800	0x9e9e9e9e2	4	5	0	0	92	0xc7c4	0x0	0	58	17	0xb7f	192.168.100.2	192.168.200.2	0x0
358	0.000000000	SYNC	0:10:ec:80:e7:8b	0:11:12:b6:6b:e2	0x800	0x8d9412bb	4	5	46	0	72	0xc7c5	0x0	0	58	17	0xad2	192.168.100.2	192.168.200.2	0x0
359	0.000000000	SYNC	0:10:ec:80:e7:8b	0:11:12:b6:6b:e2	0x800	0xf71f52c	4	5	46	0	72	0xc7c6	0x0	0	58	17	0xad1	192.168.100.2	192.168.200.2	0x0
360	0.000000000	DEL_REQ	0:11:12:b6:6b:e2	0:10:ec:80:e7:8b	0x800	0x912e719	4	5	46	0	72	0x782d	0x0	0	64	17	0x546a	192.168.200.2	192.168.100.2	0x0
361	0.000000000	DEL_REQ	0:11:12:b6:6b:e2	0:10:ec:80:e7:8b	0x800	0xd7bc0267	4	5	46	0	72	0x782e	0x0	0	64	17	0x5469	192.168.200.2	192.168.100.2	0x0
362	0.000000000	DEL_RESP	0:10:ec:80:e7:8b	0:11:12:b6:6b:e2	0x800	0xb85cc04	4	5	0	0	82	0xc7c7	0x0	0	58	17	0xb7e	192.168.100.2	192.168.200.2	0x0
363	0.000000000	DEL_REQ	0:10:ec:80:e7:8b	0:11:12:b6:6b:e2	0x800	0x7d2cb47f	4	5	0	0	82	0xc7c8	0x0	0	58	17	0xb7d	192.168.100.2	192.168.200.2	0x0
364	0.000000000	SYNC	0:10:ec:80:e7:8b	0:11:12:b6:6b:e2	0x800	0x6184eb44	4	5	46	0	72	0xc7c9	0x0	0	58	17	0xace	192.168.100.2	192.168.200.2	0x0
365	0.000000000	DEL_REQ	0:11:12:b6:6b:e2	0:10:ec:80:e7:8b	0x800	0xca6e24d5	4	5	46	0	72	0x782f	0x0	0	64	17	0x5468	192.168.200.2	192.168.100.2	0x0
366	0.000000000	DEL_RESP	0:10:ec:80:e7:8b	0:11:12:b6:6b:e2	0x800	0x9e9e9817	4	5	0	0	82	0xc7ca	0x0	0	58	17	0xb7b	192.168.100.2	192.168.200.2	0x0
367	0.000000000	SYNC	0:10:ec:80:e7:8b	0:11:12:b6:6b:e2	0x800	0x733882	4	5	46	0	72	0xc7cb	0x0	0	58	17	0xacc	192.168.100.2	192.168.200.2	0x0
368	0.000000000	DEL_REQ	0:11:12:b6:6b:e2	0:10:ec:80:e7:8b	0x800	0xd7f7e27	4	5	46	0	72	0x7830	0x0	0	64	17	0x5467	192.168.200.2	192.168.100.2	0x0
369	0.000000000	DEL_RESP	0:10:ec:80:e7:8b	0:11:12:b6:6b:e2	0x800	0xc16f3511	4	5	0	0	82	0xc7cc	0x0	0	58	17	0xb79	192.168.100.2	192.168.200.2	0x0
370	0.000000000	SYNC	0:10:ec:80:e7:8b	0:11:12:b6:6b:e2	0x800	0x7ad3a34d	4	5	46	0	72	0xc7cd	0x0	0	58	17	0xaca	192.168.100.2	192.168.200.2	0x0
371	0.000000000	DEL_REQ	0:11:12:b6:6b:e2	0:10:ec:80:e7:8b	0x800	0x559ca8c5	4	5	46	0	72	0x7831	0x0	0	64	17	0x5466	192.168.200.2	192.168.100.2	0x0
372	0.000000000	DEL_RESP	0:10:ec:80:e7:8b	0:11:12:b6:6b:e2	0x800	0x5596c348	4	5	0	0	82	0xc7ce	0x0	0	58	17	0xb77	192.168.100.2	192.168.200.2	0x0
373	0.000000000	SYNC	0:10:ec:80:e7:8b	0:11:12:b6:6b:e2	0x800	0xc83c358	4	5	46	0	72	0xc7cf	0x0	0	58	17	0xacc	192.168.100.2	192.168.200.2	0x0
374	0.000000000	SYNC	0:10:ec:80:e7:8b	0:11:12:b6:6b:e2	0x800	0xd72258f	4	5	46	0	72	0xc7d0	0x0	0	58	17	0xacc	192.168.100.2	192.168.200.2	0x0

2.5 Column Organisation

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

Files PTP ToD Report System Help

Selected packet #: Go To



inNumber	reservedField1	flagField0	flagFi
0x0	0x0	0x4	0x
0x0	0x0	0x4	0x
0x0	0x0	0x4	0x
0x0	0x0	0x4	0x

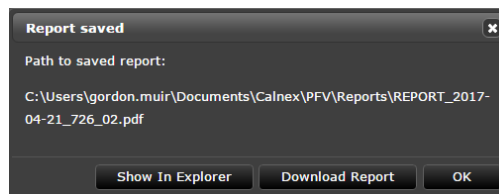
Only the columns that are available in the capture will be shown:

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

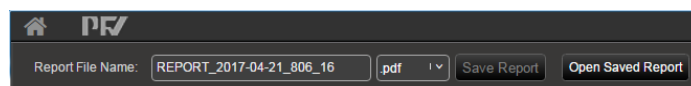
Custom information can be entered into a number of report fields

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

To create the report, click the **Save Report** button.



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 Loading a Licence File

The stand-alone PFV application (i.e. 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-100G (regardless of options installed on the instrument)
- Any pcap containing PTP data with Ethernet or IPv4 encapsulation

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

PFV

Files PTP ToD Report System Help

Options

Status

PFV Keys

The following keys are used to identify your PC. You will be asked to provide one of these keys when purchasing a licence.

Key 1:

Key 2:

Options

To order options, please contact info@calhexsol.com

Option	Description	Expiry Date	Key
PFV-PTP	PTP Profile Conformance Testing for PFV	Permanent	847BEB20C939
PFV-ToD	ToD Conformance Testing for PFV	Permanent	847BEB20C939

When ordering a licence, one of the keys listed on this page will be requested. Your PFV licence is then tied to that key.

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 File Locations

Calnex provide 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 P100G, user-defined rules files should be stored in:

\\<P100G>\Calnex100G\PFV\Rules

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

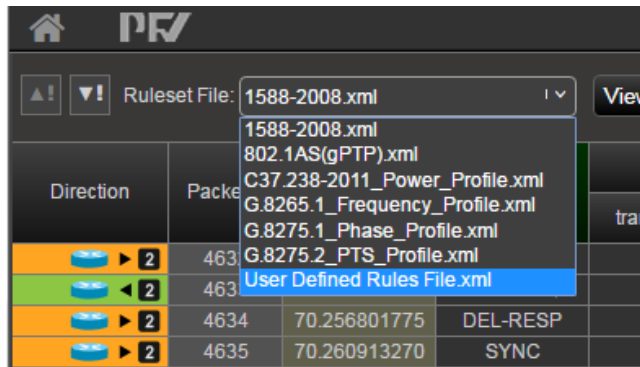
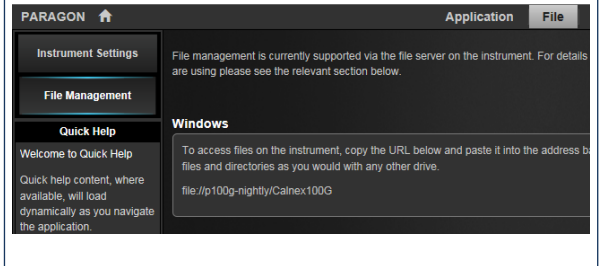
The easiest way to create a custom rules file is to copy one of the pre-defined files (e.g. 1588-2008.xml) to the user-defined folder and re-name it. The new user-defined file can then be edited.

Accessing the P100G Filesystem

The P100G filesystem can be accessed using Windows explorer.

Select File in the top-right menu bar and then File Management.

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



3.2 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="file:Rules Validation.xsd">
  <fileInformation fileVersion="0.1" name="1588-2008 PFV Rules" />
  <referenceSource documentNumber="1588-2008">IEEE Std 1588-2008</referenceSource>
  <ptpGeneral>...</ptpGeneral>
  <ptpHeader>...</ptpHeader>
  <announceMessageBody>...</announceMessageBody>
</testSpecification>
```

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 Any"
  schemaVersion="0.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file:RulesValidation.xsd">
  <fileInformation fileVersion="0.1" name="CCSA ToD PFV Rules" />
  <referenceSource documentNumber="CCSA SG15-C1324">CCSA ToD ITU_T Contribution</referenceSource>
  <todMessage>
  </todMessage>
</testSpecification>
```

The `schemaVersion` attribute to this element specifies the version of the schema associated with This may be used to ensure compatibility with the version of the PFV application. The `noNamespaceSchemaLocation` attribute specifies the location of the schema file (see "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

Bit	Name
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 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 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	Reserved0ct0Bit0
0	1	AntennaOpen
0	2	AntennaShorted
0	3	NotTrackingSatellites
0	4	Reserved0ct0Bit4

Octet	Bit	Name
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

Editing a Rules File and the XML Schema" below). **These attributes should not be changed.**

The `fileInformation` element is intended to be modified by the user 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`, `ethernetHeader`, `ipv4Header` and `todMessage`) are used to define the verification tests that will be performed by the PFV.

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

3.3 PTP Elements (`testType = PTP`)

The `ptpGeneral`, `ptpHeader`, `announceMessageBody`, `ethernetHeader` and `ipv4Header` are used to define the test that will be performed for PTP messages. In some cases, the capture file loaded into the 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
- `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.

The content of these elements are 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 rates between 1 every 2 secs and 128 per second"
    referenceDocumentNumber="1588-2008"
    referenceSectionNumber="A.9.4.2 (see also J.3.2)">
    <enabled>true</enabled>
    <messageType>SYNC</messageType>
    <tests>...</tests>
  </averageMessageRate>
  <averageMessageRate
    description="FUP message rates (sames as SYNC) between 1 every 2 secs and 128 per second"
    referenceDocumentNumber="1588-2008"
    referenceSectionNumber="A.9.4.2 (see also J.3.2)">
    <enabled>true</enabled>
    <messageType>FOLLOW-UP</messageType>
    <tests>...</tests>
  </averageMessageRate>
  <averageMessageRate description="DEL-REQ message" referenceDocumentNumber="1588-2008" referen
  <averageMessageRate description="DEL-RESP messag" referenceDocumentNumber="1588-2008" referen
  <averageMessageRate description="ANNOUNCE messag" referenceDocumentNumber="1588-2008" referen
</ptpGeneral>
```

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 (see also J.3.2)"
  <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>
  <field description="PTP Header, mes" referenceDocumentNumber="1588-2008" referenceSectionNumber="A.9.4.2 (see also J.3.2)">
  <field description="PTP Header, tra" referenceDocumentNumber="1588-2008" referenceSectionNumber="A.9.4.2 (see also J.3.2)">
  <field description="PTP Header, ver" referenceDocumentNumber="1588-2008" referenceSectionNumber="A.9.4.2 (see also J.3.2)">
  <field description="PTP Header, mes" referenceDocumentNumber="1588-2008" referenceSectionNumber="A.9.4.2 (see also J.3.2)">
  <field description="PTP Header, dom" referenceDocumentNumber="1588-2008" referenceSectionNumber="A.9.4.2 (see also J.3.2)">
  <field description="PTP Header, sou" referenceDocumentNumber="1588-2008" referenceSectionNumber="A.9.4.2 (see also J.3.2)">
  <field description="PTP Header, seq" referenceDocumentNumber="1588-2008" referenceSectionNumber="A.9.4.2 (see also J.3.2)">
  <field description="PTP Header, con" referenceDocumentNumber="1588-2008" referenceSectionNumber="A.9.4.2 (see also J.3.2)">
  <field description="PTP Header, log" referenceDocumentNumber="1588-2008" referenceSectionNumber="A.9.4.2 (see also J.3.2)">
  <field description="PTP Header, log" referenceDocumentNumber="1588-2008" referenceSectionNumber="A.9.4.2 (see also J.3.2)">
  <field description="PTP Header, log" referenceDocumentNumber="1588-2008" referenceSectionNumber="A.9.4.2 (see also J.3.2)">
  <field description="PTP Header, log" referenceDocumentNumber="1588-2008" referenceSectionNumber="A.9.4.2 (see also J.3.2)">
  <!-- Reserved Fields (other than flags) -->
  <field description="PTP Header, res" referenceDocumentNumber="1588-2008" referenceSectionNumber="A.9.4.2 (see also J.3.2)">
  <field description="PTP Header, res" referenceDocumentNumber="1588-2008" referenceSectionNumber="A.9.4.2 (see also J.3.2)">
  <field description="PTP Header, res" referenceDocumentNumber="1588-2008" referenceSectionNumber="A.9.4.2 (see also J.3.2)">

```

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" refere
  <field description="Announce Message Body, grandmasterPriority1. Disabled - could be any value 0-255" r
  <field description="Announce Message Body, grandmasterClockQuality.ClockClass" referenceDocumentNumber=
  <field description="Announce Message Body, grandmasterClockQuality.ClockAccuracy" referenceDocumentNumb
  <field description="Announce Message Body, grandmasterClockQuality.offsetScaledLogVariance" referenceDo
  <field description="Announce Message Body, grandmasterPriority2. Disabled - could be any value 0-255" r
```

Note: tests and conditions can only refer to the current ANNOUNCE message e.g. 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">
```

Note: tests and conditions can only refer to the current ANNOUNCE message e.g. 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="None" referenceSectionNumber="NA">
  <field description="Ethernet Header, Destination Address (P2P Multicast)" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="Ethernet Header, Source Address" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="Ethernet Header, EtherType" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="Ethernet Header, VLAN PCP" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="Ethernet Header, VLAN DEI" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="Ethernet Header, VLAN VID" referenceDocumentNumber="None" referenceSectionNumber="NA">
  <field description="Ethernet Header, FCS" referenceDocumentNumber="None" referenceSectionNumber="NA">
</ethernetHeader>
```

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 – 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="D.3.1">
  <field description="IPv4 Header, Differentiated Services Code Point" referenceDocumentNumber="IEEE-1588" referenceSectionNumber="D.3.2">
  <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, 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.3.3">
  <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="D.3.4">
  <field description="IPv4 Header, Destination Address (P2P Multicast)" referenceDocumentNumber="1588-2008" referenceSectionNumber="D.3.5">
  <field description="IPv4 Header, Checksum" referenceDocumentNumber="None" referenceSectionNumber="NA">
</ipv4Header>
```

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 – 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.

3.4 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 the 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 are 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 SG15-C1324"  
  <field description="Leap Seconds should be 17 (or greater in the future)" referenceDocumentNumber="CCSA S  
  <field description="PPS Status is valid from 0x0 to 0x4" referenceDocumentNumber="CCSA SG15-C1324" refere  
  <field description="Validates the Frame Check Sequence value" referenceDocumentNumber="CCSA SG15-C1324" r  
  <field description="Verifies that the timestamps are sequential (once every second)" referenceDocumentNum  
</ccsaTimeInformationMessage>
```

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 – `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 SG15-C1324" referenc  
  <field description="Time Source Status field is valid from 0x0 to 0x5" referenceDocumentNumber="CCSA SG15-C132  
  <field description="Time Source Type field is valid from 0x0 to 0x3" referenceDocumentNumber="CCSA SG15-C1324"  
  <field description="Sample alarm check rule" referenceDocumentNumber="CCSA SG15-C1324" referenceSectionNumber=  
  <field description="Validates the Frame Check Sequence value" referenceDocumentNumber="CCSA SG15-C1324" refere  
</ccsaTimeEventMessage>
```

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 – `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" re
  <field description="Verifies that the timestamps are sequential (once every second)" referenceDocum
  <field description="CurrentUTCOffset is greater than 36" referenceDocumentNumber="ITU-T G.8271/Y.13
  <field description="Validates the Frame Check Sequence value" referenceDocumentNumber="ITU-T G.8271
</g8271TimeEventMessage>
```

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 – `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" re
  <field description="versionPTP. Treating as a 1588 Header field. Spec allows 1 or 2. For the purposes of
  <field description="domainNumber. Treating as a 1588 Header field." referenceDocumentNumber="1588-2008"
  <field description="domainNumber. Treating as a 1588 Header field." referenceDocumentNumber="1588-2008"
  <field description="sourcePortIdentity.clockIdentity. Treating as a 1588 Header field." referenceDocumen
  <field description="sourcePortIdentity.portNumber. Treating as a 1588 Header field." referenceDocumentNu
  <field description="grandmasterPriority1. Treating as 1588 Announce message field. Disabled - could be a
  <field description="grandmasterPriority2. Treating as 1588 Announce message field. Disabled - could be a
  <field description="grandmasterClockQuality.ClockClass. Treating as 1588 Announce message field." refere
  <field description="grandmasterClockQuality.ClockAccuracy. Treating as 1588 Announce message field." ref
  <field description="grandmasterClockQuality.offsetScaledLogVariance. Treating as 1588 Announce message f
  <field description="grandmasterClockIdentity. Treating as 1588 Announce message field." referenceDocumen
  <field description="Validates the Frame Check Sequence value" referenceDocumentNumber="ITU-T G.8271/Y.13
</g8271TimeAnnounceMessage>
```

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 – `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"
  <field description="Time Source Type field is valid from 0x0 to 0x8" referenceDocumentNumber="ITU-T
  <field description="Time Source Status field is valid from 0x0 to 0x8" referenceDocumentNumber="ITU
  <field description="Validates the Frame Check Sequence value" referenceDocumentNumber="ITU-T G.8271
</g8271GnssMessage>
```

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.

3.5 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`, `todMessage`, `ethernetHeader` and `ipv4Header` 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.

```
<field
  description="PTP Header, messageLength. For the purposes of testing, look for minimum possible"
  referenceDocumentNumber="1588-2008"
  referenceSectionNumber="13.3.2.4">
  <fieldName>messageLength</fieldName>
  <enabled>true</enabled>
  <tests description="Check that all ">...</tests>
  <conditions description="Don't check Sig">...</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 section 3.6 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 e.g. 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`, `todMessage`, `ethernetHeader` and `ipv4Header` 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`, `ethernetHeader` or `ipv4Header`). A test is considered a pass if the test result is true.

```
<field description="PTP Header, logMessageInterval - check for NOT 0x7F (Sync,
  <fieldName>logMessageInterval</fieldName>
  <enabled>true</enabled>
  <conditions description="Applies only to">...</conditions>
  <tests description="Should be the message rate (NOT 0x7F)">
    <NOT_EQUAL>
      <value>0x7F</value>
    </NOT_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
  description="PTP Header, logMessageInterval - check for 0x7F"
  referenceDocumentNumber="1588-2008"
  referenceSectionNumber="13.3.2.11">
  <fieldName>logMessageInterval</fieldName>
  <enabled>true</enabled>
  <conditions description="Applies only to specific message types">
    <OR>
      <EQUAL>
        <fieldName>messageType</fieldName>
        <value>DEL-REQ</value>
      </EQUAL>
      <EQUAL>...</EQUAL>
      <EQUAL>...</EQUAL>
      <EQUAL>...</EQUAL>
    </OR>
  </conditions>
</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>
  <field
    description="PTP Header, message type"
    referenceDocumentNumber="1588-2008"
    referenceSectionNumber="13.3.2.2">
    <fieldName>messageType</fieldName>
    <!-- messageType = Announce | Sync | Follow_Up | Delay_Req | Delay_Resp | Pdel_Req -->
    <enabled>true</enabled>
    <tests description="Only specific message types are allowed">
      <OR>
        <!-- EQUAL with no field element implicitly refers to the current field-->
        <EQUAL>
          <value>ANNOUNCE</value>
        </EQUAL>
        <EQUAL>...</EQUAL>
        <EQUAL>...</EQUAL>
        <EQUAL>...</EQUAL>
      </OR>
    </tests>
  </field>
</ptpHeader>

```

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 has to 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)"
  referenceDocumentNumber="1588-2008"
  referenceSectionNumber="13.3.2.6">
  <fieldName>alternateMasterFlag</fieldName>
  <enabled>true</enabled>
  <conditions description="Applies only to specific message types">
  <tests description="alternateMasterFlag should be false">
    <FLAG_CHECK>
      <bit>alternateMasterFlag</bit>
      <value>>false</value>
    </FLAG_CHECK>
  </tests>
</field>

```

The **AUTO** element causes the PFV to run a test that cannot be specified using the existing rules file syntax. For example, using `sequenceId` in an **AUTO** element causes the 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` and `ipv4Header` 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 (Annou
  <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 m">...</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.

3.6 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

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`, `todMessage`, `ethernetHeader` and `ipv4Header` only. Specifies an automatic test built-in to the PFV)

Logical Operators

The supported logical operators are: AND, OR.

Automatic tests

There a number of automatic tests that are not defined by the rules file. These are performed by the 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`

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 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

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
sourcePortIdentity								10	20
sequenceId								2	30
controlField								1	32
logMessageInterval								1	33

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 Synchronisation.

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`.

Ethernet Header Fields

The supported Ethernet Header fields are listed in the 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 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 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	6	7		
ipv6Version				ipv6Dscp				ipv6Ecn		Flow label				2	0		
ipv6FlowLabel																2	2
ipv6Payloadlength																2	4
ipv6NextHeader								ipv6HopLimit								2	6
ipv6SourceAddress																16	8
ipv6DestinationAddress																24	8

CCSA Time Information Message Fields

The supported CCSA Information Message fields are listed in the 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 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

Bits								Octets	Offset
7	6	5	4	3	2	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 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

Bit	Name
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 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 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

Octet	Bit	Name
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

3.7 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 is delivered with a schema file (Rules Validation.xsd) for the XML rules files. This is used by the 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 the PFV. Some of these editors may also automatically suggest the correct syntax and structure while editing.

Calnex Solutions Ltd
Oracle Campus
Linlithgow
West Lothian
EH49 7LR
United Kingdom

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

calnexsol.com

© Calnex Solutions Ltd., 2020

This information is subject to change without notice

March 2020

