Ethernet Packet Generator

TCL API GUIDE

Version 1.4

APG4 APG8 APG208

Hardware Versions

This document applies to the following hardware versions:

Version 1 (APGV1)	**Launched: 2016 with API 161016** **Products: APG4, APG8, APG200, APG208** **Port Speeds: 1Gbps, 10Gbps, 40Gbps** **Processing Cores: 6.X (10Gbps) 7.X (1Gbps)** Note: Cannot mix 10Gbps and 1Gbps ports **Capture Buffer: 64MB/port (10G) 256MB/port (40G)** **Timestamp Resolution: 8ns**
Version 2 (APGV2)	**Launched: 2023 with API 161016** **Products: APG4V2-10, APG8V2-10** **Port Speeds: 1Gbps, 10Gbps** **2.5Gbps, 5Gbps (optional)** **Processing Core: 10.X (10Gbps/1Gbps)** **Capture Buffer: 128MB/port** **Timestamp Resolution: 2.5ns**

Software Versions

This document applies to the following software versions:

	Version 1	**Version 2**
APG API	**161016**	**161016**
APG Unit Firmware	**Version 2.3**	**Version 1.2-2**
APG Processing Core	**Version 6.010E (10Gbps)** **Version 7.0103 (1Gbps)** Datecode: 01 March 2019	**Version 10.2** Datecode: 1 December 2023
APG TCL API	**Version 1.4.1**	**Version 1.4.1**

Revision History

Date	Version	Changes
12 August 2016	0.8	Restricted customer release
7 June 2017	1.0	General Release • Added APG200 • Added Timestamp Analysis (see Section 3.10) • Added API Initialisation (Section 4) • apgGetVersion renamed to apgGetApiVersion (Section 4.2) • apgOpen waits for unit ready (Section 5.1.1) • Removed inconsistencies in apgSaveConfiguration PORTLIST (see Section 6.1.1) • RXTIME replaces TIMESTAMP in apgGetPort CAPTURE PACKET command (see Section 8.2.9) • apgSetPort command simplified (Section 8.3.1) • Added apgSetPort TOPOLOGY command to enable switching between 40G / 4x10G modes (Section 8.3.1) • Added Deep Packet Capture (see Section 8.1.3) • Added Packet/Second Transmit Rate (Section 9.3.2) • Added Multi-Burst Transmit Mode (Section 9.3.2) • apgLoadStream command modified (Section 9.1.1)
1 September 2017	1.1	• Improved topology change behaviour • Improved device connection/disconnection behaviour • Improved capture download performance
8 February 2018	1.1.3	• apgOpen returns value if unable to connect (Section 5.1.1) • Local capture buffer sizes increased to 64KB (10Gbps), and 256KB (40Gbps) (Section 3.9)
27 February 2019	1.2	• Added apgUSBOpen command (Section 5.1) • Added 1Gbps Topology for SFP Ports (Sections 3.3.1 & 8.3.1.1) • Added Copper Transceiver support: Extended apgGetPort MODULE command (Section 8.2.3) Extended apgSetPort command (Section 8.3.1.2) Added apgApplyPort COPPERMODULE (Section 8.4.2)
4 September 2019	1.2.2	• Corrected ETHERNET_II (underscore not hyphen) • Corrected minor typos and updated software versions
21 February 2023	1.3	• Corrected minor typos and updated software versions • Added APGV2 support • Added APGV1 and APGV2 descriptions • Added Transmit Control description (Section 3.5) • Added Inter-Unit Synchronisation description (Section 3.7) • Added IPCONFIG option to apgLoadUnit and apgGetUnit commands (Section 7.1.2) • Added CLOCKSTATUS option to apgLoadUnit and apgGetUnit commands (Section 7.1.4) • Added apgSetUnit command to set clock modes and timestamp reset configuration (Section 7.3) • Added apgApplyUnit command to apply clock mode configuration (Section 7.4) • Added apgControlUnit command to control inter-unit synchronisation (Section 7.5) • Changed apgLoadPort CAPTURE warnings (Section 8.1.3) after download speed improvements in the TCL API

Date	Version	Changes
(cont)	1.3	• apgLoadPort ANALYSIS command arguments modified (Section 8.1.4) • Corrected ADV_SPEED description in the copper module configuration table (Section 8.3.1.2) • Added 'armed' state to port states (Section 8.5) • Added apgControlPort [SYNC COMMAND] with STARTSYNCTX option to to arm ports for inter-unit synchronous start (Section 8.5.2) • Corrected apgControlPort DISABLECAPTURE description (Section 8.5.4) • Corrected ETHERNET_II (underscore not hyphen) (Section 9.2.3.2) • Extended apgGetVariables to include new unit options (Section 10.1.1) • Added apgGetHwVersion function (Section 10.3.1) to return unit version (APGV1 or APGV2) • Added apgConvertTicksToTime (Section 10.4.1) to convert clock ticks to time depending APG unit version.
31 January 2024	1.4	• Corrected Temperature and Fan Status values (Section 7.2.3) • Corrected Module Status values (Section 8.2.2) • Added apgControlPort ZEROPORTSEQ option to reset port sequence number (Section 8.5.1) • Added 2G5bps/5Gbps support (optional feature) • Added Port PRBS Bit Error Rate test mode: Overview (Section 3.6) apgLoadUnit PRBSSTATUS (Section 7.1.8) apgGetUnit PRBSPORTLIST (Section 7.2.6) apgGetPort PRBS (Section 8.2.6) apgSetPort PRBSMODE (Section 8.3.1.3) apgSetPort PRBSPATTERN (Section 8.3.1.3) apgApplyPort PRBS (Section 8.4.3) apgControlPort PRBSCLEAR (Section 8.5.3)

Document Conventions

INFORMATION:
Additional information to clarify functionality or usability

WARNING:
Clarification of unexpected or restricted functionality

Disclaimer

Axtrinet retains the right to make changes to this document at any time, without notice. The information in this document is believed to be accurate and reliable. Axtrinet does not warrant the accuracy of completeness of the information, text, graphics or other items contained within this document.

Axtrinet provides the software and the documentation "as is" without warranties of any kind.

Axtrinet disclaims all warranties and representations of any kind relating to products, software or services provided hereunder, whether express, implied, statutory, including without limitation the implied warranties of merchantability, fitness for a particular purpose, accuracy, or non-infringement of third party rights.

Axtrinet does not warrant that the software will in every case process all data correctly, or that operation of the products, including software, will be uninterrupted, free from error, or secure.

The disclaimers in this section will not apply to the extent prohibited by applicable law.

The software is not designed, intended, or certified for use in components of systems intended for the operation of weapons, weapons systems, nuclear installations, means of mass transportation, aviation, medical systems, devices, implants, or equipment, pollution control, hazardous substances management, or for any other dangerous application in which the failure of the products could create a situation where bodily injury or death may occur. The use of the software in any such application is solely at your own risk.

Copyright statement

Trademarks

Windows is a registered trademark or trademark of Microsoft Corporation, registered in the U.S. and other countries.

ActiveTcl is a registered trademark or trademark of ActiveState, registered in the Canada and other countries.

Axtrinet is a registered trademark of Xentech Solutions Limited, registered in the UK.

Preface

About This Document

This manual describes the structure and commands of the Axtrinet TCL API, and contains the following sections:

Section	Description
1. INTRODUCTION	Summary of the TCL API interface and capabilities
2. INSTALLATION	Installation procedures for the Microsoft™ Windows™ and Linux drivers and TCL API Interface
3. TEST ENVIRONMENT	Introduction to Ethernet packet generation, packet structures and control
4. API INITIALISATION	Initialising the APG TCL API and version command
5. CONNECTION COMMANDS	Open and Close connections to the APG units
6. CONFIGURATION COMMANDS	Save and Apply unit configurations
7. UNIT COMMANDS	Unit-level connection, load, and get TCL commands
8. PORT COMMANDS	Port-level configuration load, get, set and apply TCL commands
9. STREAM COMMANDS	Stream-level configuration load, get, set and apply TCL commands
10. TOOLS	Command and Header tools to simplify access to the internal database variables

Related Documentation

[1]	APG-UG	Axtrinet User Guide (including APG Control Interface)
[2]	APG-HDR	Axtrinet Header Definitions
[3]	APG-SW-TC	Axtrinet APG Software License Terms And Conditions

Visit **www.axtrinet.com/documentation** for the latest documentation.

Glossary

APG	Axtrinet Packet Generator
API	Application Programming Interface
BER	Bit Error Rate
BERT	Bit Error Rate Test
FCS	Frame Checksum
IBG	Inter-Burst Gap
IFG	Inter-Frame Gap
pps	Packets per Second
PPS	Pulse per Second (clock)
PRBS	Pseudo-Random Bit Sequence
QSFP+	Quad Small Form-Factor Pluggable (40Gbps)
RPM	Revolutions per Minute
SFD	Start-of-Frame Delimiter
SFP	Small Form-Factor Pluggable (1Gbps)
SFP+	Small Form-Factor Pluggable (10Gbps)
TCL	Tool Command Language
USB	Universal Serial Bus

Contents

1. Introduction

Thank you for purchasing an Axtrinet™ APG Ethernet Packet Generator.

The Axtrinet APG Ethernet Packet Generators provide compact and affordable 40Gbps and 10Gbps/1Gbps Ethernet Packet Generator/Analysers with a simple-to-use Control Interface and an open TCL API for third party scripting.

Ideally suited to applications in R&D, Test and Manufacturing environments, and 'on the road' with Field Sales and Application Engineers, the Axtrinet APG Ethernet Packet Generators allow reliable and affordable development and testing of:

- Ethernet network equipment such as switches, routers, firewalls and network monitoring devices
- Data storage equipment with 10Gbps/1Gbps and 40Gbps Ethernet interfaces
- Specialist devices such as FPGA accelerator NIC cards and offload appliances
- Ethernet infrastructure installations encompassing cabling and switches

1.1 Main Features

- Highly configurable Ethernet Packet Generation
- Full wire-speed operation on all ports
- Industry standard QSFP+ and SFP+/SFP ports
- Real-time packet counts and error detection
- Packet Capture for post-test analysis
- Simple to use Control Interface for configuration and control
- Clear LED status indication for unit operation and Ethernet traffic generation/reception
- USB 2.0 port for easy set up and local management and 10/100Mbps Ethernet LAN connection for flexible remote management

1.2 Benefits

- Low cost allows multiple units to be deployed in a development environment – one per desk
- Easy to set up and use. Avoids the need for complex vendor specific programming skills
- Ideal for integration into a manufacturing test environment using TCL scripting interface
- Flexible choices of interfaces allows use with different speeds and media types, maximising the investment across multiple projects
- Small size, 1U high (44mm) and 146mm wide, for desk-top or rack shelf mounting (1/3 rack width)

1.3 Software Licences and Support

See the Axtrinet APG Software License Terms & Conditions [3]

Email based software support is included in the purchase price for the first 12 months after delivery. Extended Software Support is available for purchase; please contact Axtrinet or your reseller for more information.

1.4 CONTACT DETAILS

Technical assistance is available from Axtrinet at the following address:

Address:	Xentech Solutions Suite 6 Stanta Business Centre 3 Soothouse Spring St Albans AL3 6PF UK
Phone:	+44 (0)1727 867795
Email:	
Technical Support:	**support@axtrinet.com**
Sales:	**sales@axtrinet.com**
Web Site:	**www.axtrinet.com**

2. Installation

APG TCL API installation process on a host PC running Linux or Windows is described in the APG User Guide [1] Section 2.

2.1 Minimum System Requirements

Processor	Pentium-class processor or equivalent
Memory	2GB (4GB recommended)
Disk Space	15MB
OS	64bit (x86_64) Linux systems Microsoft Windows 7 Microsoft Windows 8.x Microsoft Windows 10.x Microsoft Windows 11.x
Interfaces	Minimum: USB 2.0 Preferred: USB 2.0 & 10/100Base-T

2.2 Prerequisites

In the Windows Environment, a TCL distribution (such as Activestate® ActiveTcl) must be installed before using the APG TCL API.

In the Linux Environment, the TCL package for your 8.6.x distribution must be installed before using the APG TCL API.

2.3 Testing the Installation

2.3.1 Windows Environment

The Axtrinet TCL package is installed in **C:\Program Files\Axtrinet\APG\tcllib**

To test the installation, run "test.tcl" in the \examples directory:

```
% cd C:\Program Files\Axtrinet\APG\examples\
% tclsh test.tcl
APG TCL API APGV1.4.1
Build Date 1706631374
Target API Version 161016
```

The Axtrinet TCL API has been successfully installed if test.tcl completes without errors, and displays the API version, build data and target API version.

2.3.2 Linux Environment

The Axtrinet TCL package is installed in **/usr/share/axtrinet/apg/**

To test the installation, run "test.tcl" in the /examples directory:

```
$ /usr/share/axtrinet/apg/examples/test.tcl
APG TCL API APGV1.4.1
Build Date 1706631374
Target API Version 161016
```

The Axtrinet TCL API has been successfully installed if test.tcl completes without errors, and displays the API version, build data and target API version.

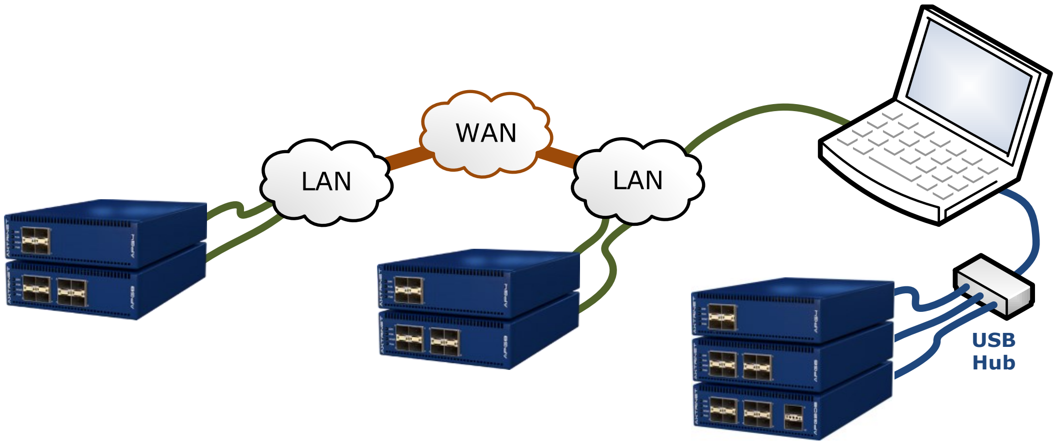
2.4 Example Files

Example files are provided in the demo directory that can be run within your TCL environment:

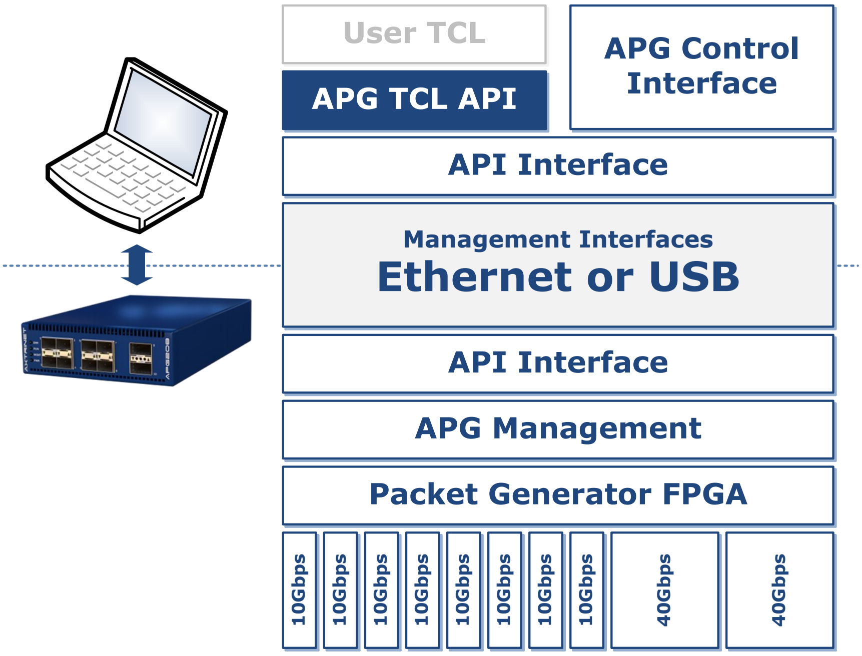
1.	demo_init.tcl	Initialise Axtrinet APG API Package
2.	demo_connection.tcl	Open and Close connections to the Axtrinet APG
3.	demo_save_recall.tcl	Save and load port and stream configuration
4.	demo_unit_commands.tcl	Load and display unit information
5.	demo_port_commands.tcl	Load and display unit and port information. It includes the following examples:
	a) demo_generate_portlist.tcl	Generate list of unit ports and streams
	b) demo_configure_stream.tcl	Configure streams and apply to unit
	c) demo_port_control.tcl	Start/stop traffic, calculate rates, capture packets
	d) demo_port_counters.tcl	Read and display port counters and rates
	e) demo_port_capture.tcl	Read and display captured packets
6.	demo_inter_unit_sync.tcl	Inter-unit connection, clock synchronisation, synchronous timestamp reset, synchronous start, deep capture, download and display.
7.	demo_prbs.tcl	Enable PRBS mode, pattern and monitor status

3. Test Environment & Concepts

The Test Environment consists of one or more Axtrinet Packet Generators. The APG TCL API can connect to single unit over a direct USB connection or multiple units over USB or the Ethernet management interface. The units may be located in the same location, or in geographically separate locations connected by a WAN.



All accessible units can be managed through the same Control Interface or TCL scripting interface. The APG TCL API and the APG Control Interface share the same API to the Axtrinet Packet Generator:



The Axtrinet APG API provides a TCL scripting interface for automated test generation.

The API provides access to:

- unit, port and stream configuration and status
- packet counters (packets, bytes, errors)
- packet capture tools
- port control (start, step, stop)

The Axtrinet APG Control Interface is described in the APG User Guide [1].

3.1 DEFINITIONS

UNIT

A unit is a single physical Axtrinet Packet Generator (APG4, APG8 or APG208).

A unit is assigned a UNITID when the management connection is first opened using the APG TCL API. The UNITID is fixed for the duration of the TCL session, and is used to uniquely identify a unit during the session. The UNITIDs will be reassigned if the TCL session is restarted.

Unit commands enable management connections; provide access to the unit status (See Section 4).

PORT

A PORT is a physical aperture on the unit (SFP+/SFP or QSFP+).

Ports are numbered from 1.

The UNITID is used in conjunction with the PORT number to create the PORTID (eg Unit 2, Port 7 has a PORT ID {2 7}). The PORTID is fixed for the duration of the TCL session, and is used to uniquely identify a port during the session.

Port commands allow access to the modules; port capabilities and configuration; and port counters (see Section 8).

SUBPORT

Where a QSFP+ port can be configured into different topologies (eg 40Gbps or 4x10Gbps), the PORTID is qualified with a SUBPORT.

Subports are numbered from 1.

The UNITID is used in conjunction with the PORT and SUBPORT number to create the SUBPORTID (eg Unit 3, Port 10, Subport 1 has a SUBPORT ID {3 10 1}). The SUBPORTID is fixed for the duration of the TCL session, and is used to uniquely identify a port during the session.

Subports use the Port commands to gain access to the modules; port capabilities and configuration; and port counters (see Section 8).

MODULE

A MODULE is a SFP+/SFP or QSFP+ transceiver, and must be inserted into a port aperture to enable a link. Module configuration and status are accessed with the Port Commands (see Section 8)

STREAM

A transmit stream generates a controlled number of Ethernet frames with a defined length and rate; fixed header configuration with a fixed or varying header contents; and a fixed or varying payload.

The UNITID and PORTID are used in conjunction with the STREAM number to create the STREAMID (eg Unit 1, Port 3 Stream 2 has a STREAMID {1 3 2}).

When a port has subports, the SUBPORTID is also required to form the STREAMID (eg Unit 1, Port 9, Subport 2 Stream 6 has a STREAMID {1 9 2 6}).

The STREAMID is fixed for the duration of the TCL session, and is used to uniquely identify a stream during the session.

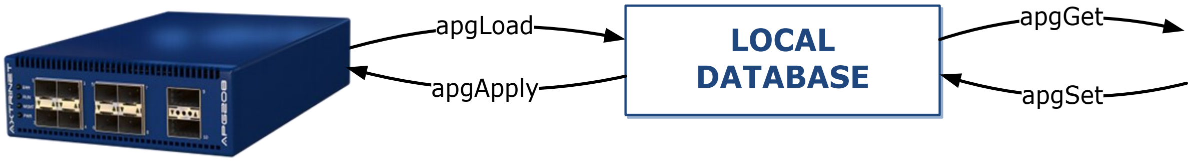
The outputs from the eight stream generators are multiplexed into a single stream for transmission from a port.

Stream commands allow the stream configuration to be read from the unit, modified and written to the unit (See Section 9)

3.2 APG Configuration

The unit, port and stream configurations are stored on the Axtrinet™ APG unit.

The TCL API uses a 'local' database to store the current and modified configurations, before being applied into the unit.



The TCL API provides the tools to **apgLoad** the APG unit configuration and status into the local database, read (**apgGet**) and modify (**apgSet**) the local database, and write (**apgApply**) to the APG unit.

The traffic generator is controlled with the **apgControl** command.

The unit retains its configuration over a power cycle.

3.3 Port Topology

Port Topology defines the physical port configuration (eg 40Gbps, 4x10Gbps), rather than the interface type (eg QSFP+).

Port topology is changed with the **apgSetPort TOPOLOGY** command (see Section 8.3.1).

3.3.1 SFP+/SFP Ports

The SFP+/SFP Port Topology can be 10Gbps or 1Gbps.

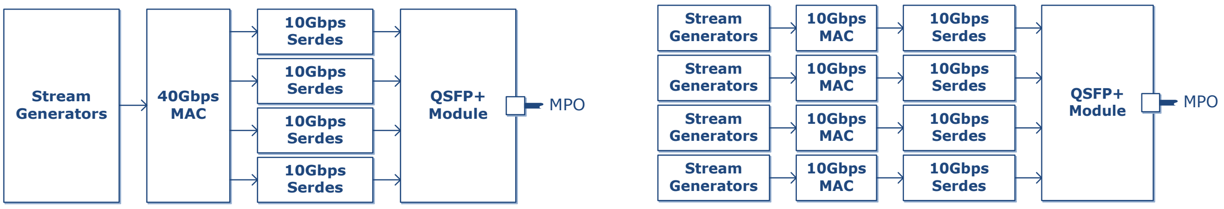
⚠ On a APGV1 unit, 10Gbps and 1Gbps port speeds cannot be mixed within a unit. The unit will reboot after switching SFP+/SFP Topology to load a new FPGA image.

ℹ An APGV2 unit can be configured with a mix of port speeds.

3.3.2 QSFP+ Ports (APG208 only)

The QSFP+ interface topology can be configured in 40Gbps mode (default) or 4X10Gbps mode, where each of the 10Gbps lanes that comprise the 40Gbps link are managed independently.

Changing the port topology of a QSFP+ port changes both the transmit and receive paths.



a) QSFP+ Port in 40Gbps Topology (default)

b) QSFP+ Port in 4x10Gbps Topology

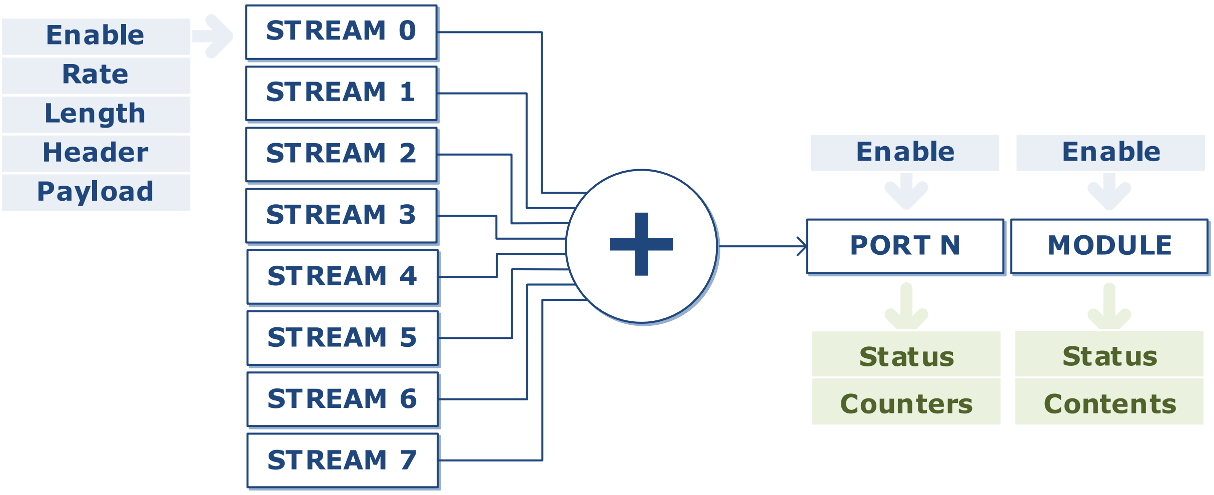
Switching between 40Gbps and 4x10Gbps topologies is performed with the **apgSetPort TOPOLOGY** command.

Mixed topology of the QSFP+ ports is **only** available with Port 9 at 4x10Gbps, and Port 10 at 40Gbps. Setting Port 9 to 40Gbps or Port 10 to 4x10Gbps topologies will automatically switch the other port into the same mode.

3.4 Transmit Configuration

Each port contains a transmit engine that comprises:

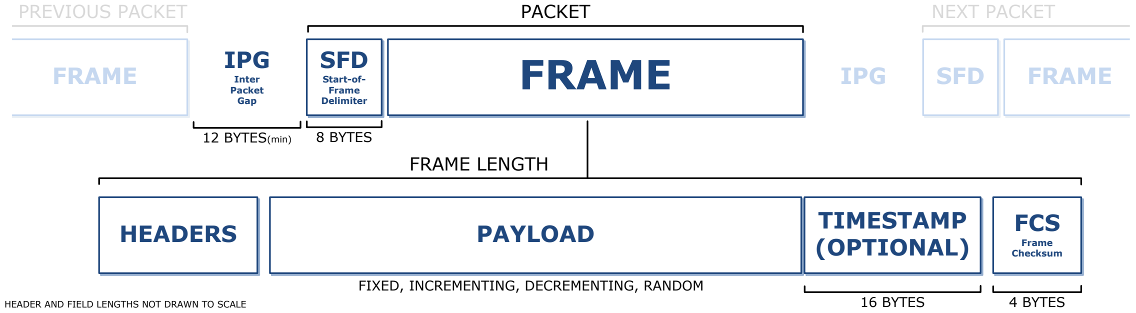
- 8 parallel independent configurable Ethernet stream generators
- Stream multiplexer
- Transmit port configuration and status
- Module configuration and status



The streams are processed in a round-robin sequence, transmitting a packet if it queued and ready to send.

To bring a link up, both the port and module must be enabled.
To transmit a stream, the port, module and stream must be enabled.
The transmit mode must be CONTINUOUS or a non-zero BURST.

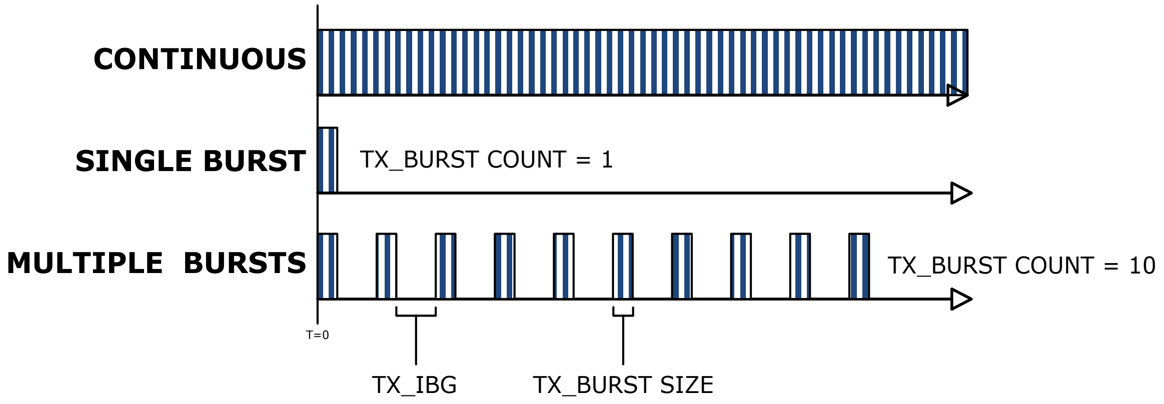
The Stream Generator defines an Ethernet frame:



A transmit stream generates a controlled number of Ethernet frames at a controlled length and rate; with a fixed header configuration, fixed or varying header contents, and a fixed or varying payload.

Streams can be enabled or disabled using **apgGetStream CONFIG**.

The stream transmit mode can be set using **apgSetStream CONFIG**, and defines how the packets are generated: either Continuously; as a Single Burst of **TX_BURST_SIZE** packets; or a Multiple Burst of **TX_BURST_SIZE** packets, repeated **TX_BURST_COUNT** times, separated by **TX_IBG.**



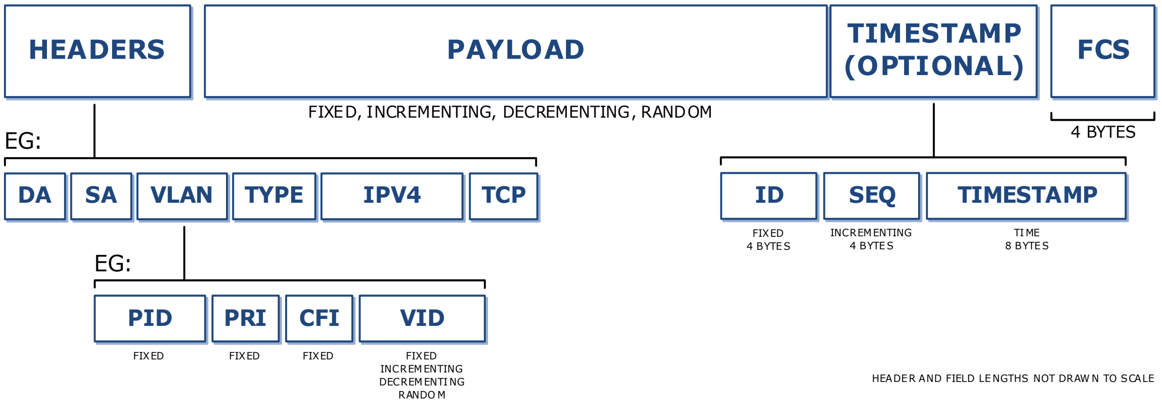
The stream transmit rate can be set using **apgSetStream CONFIG**. The stream rate can be set as in percent, packets per second or clock cycles.

> Care should be taken to ensure that the cumulative stream transmit rates does not exceed the port capacity.
>
> If the cumulative stream transmit rates does exceed the port capacity, the port will transmit at wire rate, but the streams will transmit at a lower rate than configured.

The length defines the total length of the frame in bytes, including the headers, payload, timestamp (if enabled) and 4-byte Frame Checksum (FCS). The length can be fixed, or incrementing, decrementing or random over a range using **apgSetStream CONFIG**.

The HEADER is created by adding header types (eg MAC, VLAN, IPV4) to the stream using **apgSetStream HEADER HEADER_LIST**.

Headers can be configured using **apgSetStream HEADER**.



The payload can be fixed, or incrementing, decrementing or random over a range using **apgSetStream PAYLOAD**.

The payload may additionally include a timestamp and sequence number.

The optional "Timestamp Fields" incorporating the timestamp ID, Sequence Number and Timestamp are enabled on a "per stream" basis using the **apgSetStream PAYLOAD TS_ENABLE** command, and processed using the **apgLoadPort ANALYSIS** and **apgGetPort ANALYSIS** commands.

The Timestamp ID field is fixed at 0x0 in APG Control Interface Version 3.2.

The Sequence Number is set to zero when the unit powers up, and increments continuously for each packet transmitted with the Timestamp field enabled.

The Sequence Number will wrap after 2^{32} (4,294,967,296) packets. Wire-rate 64 byte packets at 40Gbps will take 1 min 12 secs.

On an APGV1 unit, the Sequence Number cannot be reset

On a APGV2 unit, the Sequence Number can be reset with the **apgControlPort ZEROPORTSEQ** command.

It is not possible to reset the Timestamp on a Version 1 unit.

On a Version 1 unit, the Timestamp indicates the time in 8ns cycles since the unit was reset

The timestamp on a Version 2 unit can be reset, and indicates the time in **2.5ns** cycles since the unit or timestamp was reset.

To bring a link up, both the port and module must be enabled.
To transmit a stream, the port, module and stream must be enabled.
The transmit mode must be CONTINUOUS or a non-zero BURST.

A port can be enabled or disabled using **apgSetPort** command.

Port link status and link speed are accessible using **apgGetPort INFO** and **apgGetPort STATUS**, and the transmit counters can be read using **apgGetPort TXSTATS**.

A module can be enabled or disabled using **apgSetPort** command.

The module type, vendor and capabilities are accessible using **apgGetPort MODULE**.

The traffic generator is controlled with the **apgControl** command.

3.5 Port Control

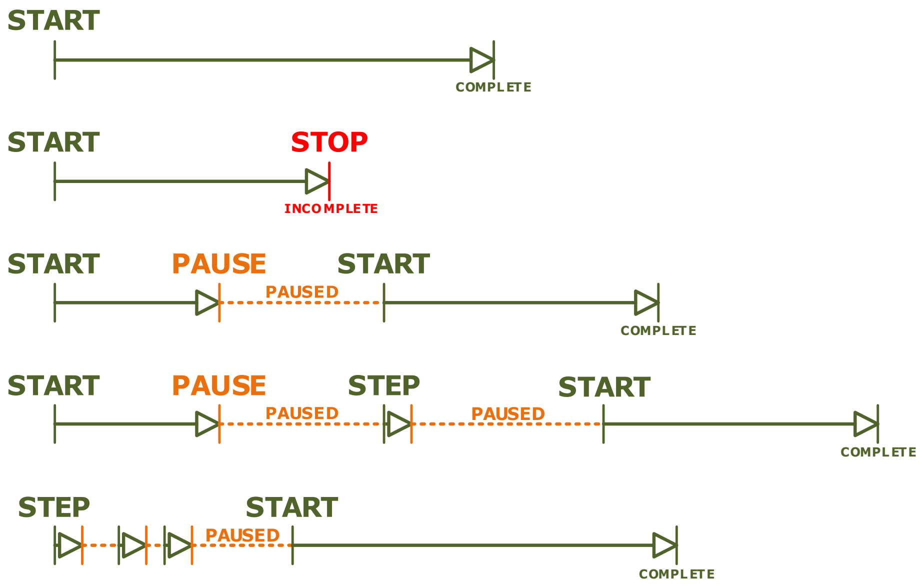
Packet transmission is controlled with the **apgControlPort** command (Section 8.5.1) that operates on ports specified by [PORLIST]. [PORTLIST] can contain a single port, or a list of ports.

The **apgControlPort STARTTX** starts packet generation of the enabled streams.

apgControlPort STOPTX stops the packet generation. **apgControlPort START** will restart the packet generation from the first packet.

apgControlPort PAUSETX interrupts the packet generation. **apgControlPort START** continues the packet generation from where it was paused.

apgControlPort STEPTX generates only the next packet, then pauses.



3.6 PRBS Bit Error Rate Test Mode (APGV2 Only)

A Port can operate in ‘normal’ mode (generating Ethernet Packets from the Stream configuration); or ‘PRBS’ mode to generate a serial bit stream (not Ethernet packets) at the bit rate of the configured port speed.

Each port has separate enables for transmit and receive. A port is in PRBS mode if either the PRBS transmitter or receiver are enabled. The PRBS Mode can be set with the **apgSetPort PRBSMODE** command.

The port supports the following PRBS Patterns:

PRBS7	(x7 + x6 + 1)
PRBS9	(x9 + x5 + 1)
PRBS15	(x15 + x14 + 1)
PRBS23	(x23 + x18 + 1)
PRBS31	(x31 + x28 + 1)

The PRBS Pattern can be set with the **apgSetPort PRBSPATTERN** command.

The **apgSetPort** command **must** be followed with an **apgApplyPort PRBS** command.

The PRBS configuration and status can be read from the Axtrinet unit with the **apgLoadUnit PRBSSTATUS** command, followed by **apgGetPort PRBS** commands.

The configuration can be read with **apgGetPort PRBS MODE** and **PATTERN**.

The PRBS receiver will attempt to lock automatically to the transmitted bit stream. The **apgGetPort PRBS LOCKED** bit is set and the **RUNTIME** timer (in milliseconds) is reset.

The **apgControlPort PRBSCLEAR** command should be used to clear the bit **ERRORS** counter.

The Bit Error Rate (BER) is calculated and can be read with the **apgGetPort PRBS BER** command.

3.7 INTER-UNIT SYNCHRONISATION (APGV2 ONLY)

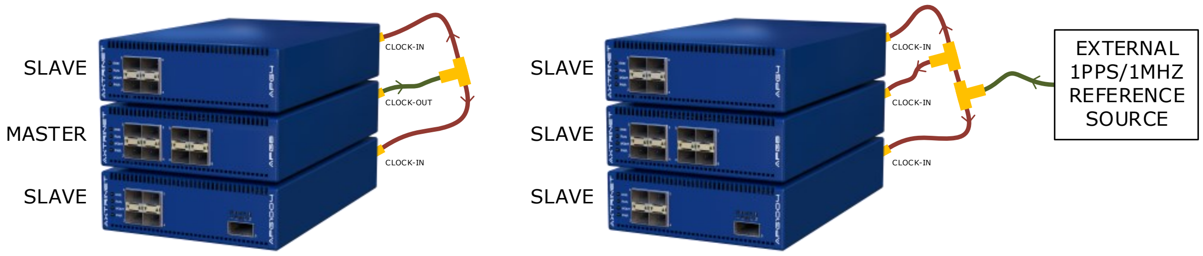
The Axtrinet unit uses an internal oscillator to clock a PLL-based clock synthesiser to generate the system clocks for the Ethernet Packet Generator, Ethernet and USB management interfaces. The clocks on separate APG units are free-running and will not be frequency or phase locked to each other.

Each unit has a SYNC interface and CLOCK-IN / CLOCK-OUT interfaces. The SYNC interfaces on each unit must be connected together; and the CLOCK-OUT from the master unit must be connected to the CLOCK-IN interfaces on the slave units.

3.7.1 Clock Synchronisation

The Clock Interfaces allow multiple units to be connected together to synchronise system clocks to an external 1PPS (1HZ) or 1MHz reference. The clock reference can be generated locally from one of the connected Axtrinet units; or from an external timing reference source.

The electrical specification is defined in the Axtrinet User Guide [1].



a) Multiple APG units synchronised locally

b) Multiple APG units synchronised from external source

Short co-axial cables (<50cm) are recommended to connect the CLOCK-OUT to the CLOCK-IN interfaces. Very short cables (<20cm) are preferable.
For optimum performance, a single 120Ω termination load is recommended at a 'slave' end of a short (<50cm) clock connection.
120Ω terminations loads are recommended at **both** ends of a long clock connection.

The CLOCK-IN mode is set with the **apgSetUnit CLKINMODE** command. The clock input can be set to internal (free-running internal system clock) or external to synchronise the system clocks to a 1PPS or 1MHz signal.

The CLOCK-OUT mode is set with the **apgSetUnit CLKOUTMODE** command. The clock output can be set to 1PPS (1Hz) or 1MHz.

If using a locally generated clock source, **one** unit must be set to the INTERNAL clock source (ie Clock Master); and the other units must be set to EXTERNAL clock source (ie Clock Slaves). If using an external reference source, all units must be set to EXTERNAL clock source. The synchronisation frequency (1PPS or 1MHz) must match on all connected units.

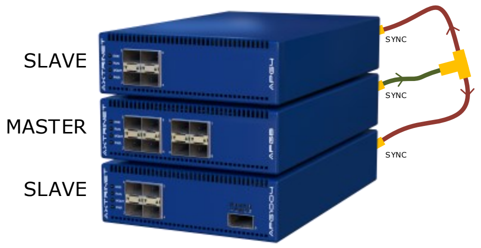
The system clock status is read with the **apgLoadUnit CLOCKSTATUS** followed by **apgGetUnit CLOCKSTATUS** commands.

System clocks on the slave units will be frequency and phase locked to the master clock. If the CLOCK-OUT is enabled on the slave units, the 1PPS or 1MHZ output on the synchronised units will not be phase aligned.

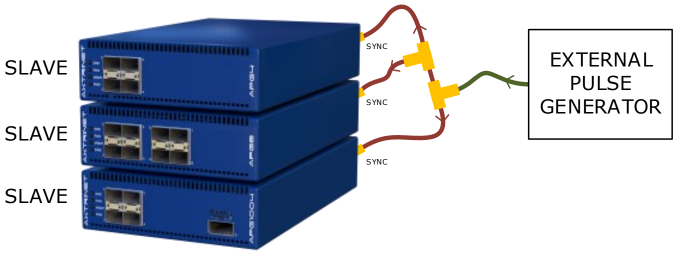
3.7.2 Timestamp Synchronisation

Timestamp synchronisation requires the rear panel SYNC interfaces are connected between the units. The synchronisation pulse can be generated locally from one of the connected Axtrinet units; or from an external pulse generator.

The electrical specification is defined in the Axtrinet User Guide [1].



a) Multiple APG units synchronised locally



b) Multiple APG units synchronised from external source

Very short co-axial cables (<20cm) are recommended to connect the SYNC interfaces. For optimum performance, a single 120Ω termination load is recommended at a 'slave' end of a short (<50cm) sync connection.
120Ω terminations loads are recommended at **both** ends of a long sync connection.

The timestamp reset mode is set with the **apgSetUnit TSRESETMODE** to ZERO or TIME.

Resetting the timestamps on the connected units is a 2-stage process:

1. Arm the connected units with the **apgControlUnit TSSYNCCLEAR** command.
2. Synchronise the timestamps with the synchronisation pulse, either generated locally with the **apgControlUnit SYNCGO** command, or externally from a pulse generator.

If the timestamp reset mode is ZERO, there is no requirement for the time between the **apgControlUnit TSSYNCCLEAR** and the synchronisation pulse.

If the timestamp reset mode is TIME, it is recommended that the **apgControlUnit TSSYNCCLEAR** is immediately followed by a locally generated synchronisation pulse from the **apgControlUnit SYNCGO** command to synchronise the unit timestamp close to the host PC time.

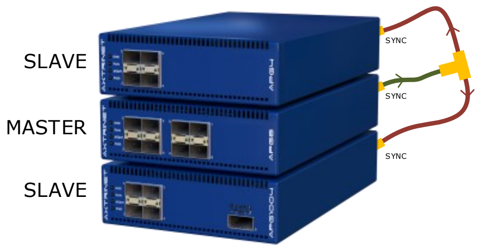
The 'synchronised' ports reset the timestamp on the rising edge of the synchronisation pulse.

If the units are armed with the **apgControlUnit TSSYNCCLEAR** command and the synchronisation event is **not** sent or detected, the unit control functions will be blocked, waiting for the synchronisation event. The armed state can be removed with the **apgControlUnit SYNCCLEARUP** command.

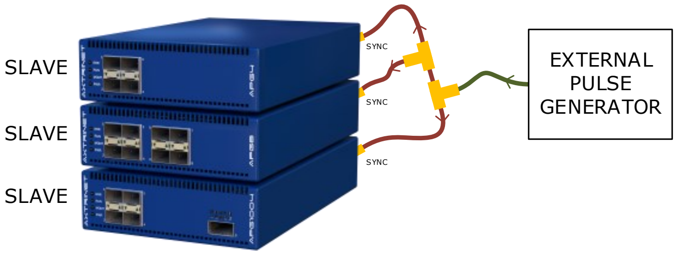
3.7.3 Transmit Synchronisation

Transmit synchronisation requires the rear panel SYNC interfaces are connected between the units. The synchronisation pulse can be generated locally from one of the connected Axtrinet units; or from an external pulse generator.

The electrical specification is defined in the Axtrinet User Guide [1].



a) Multiple APG units synchronised locally



b) Multiple APG units synchronised from external source

Very short co-axial cables (<20cm) are recommended to connect the SYNC interfaces. For optimum performance, a single 120Ω termination load is recommended at a 'slave' end of a short (<50cm) sync connection.
120Ω terminations loads are recommended at **both** ends of a long sync connection.

Transmit synchronisation on the connected units is a 2-stage process:

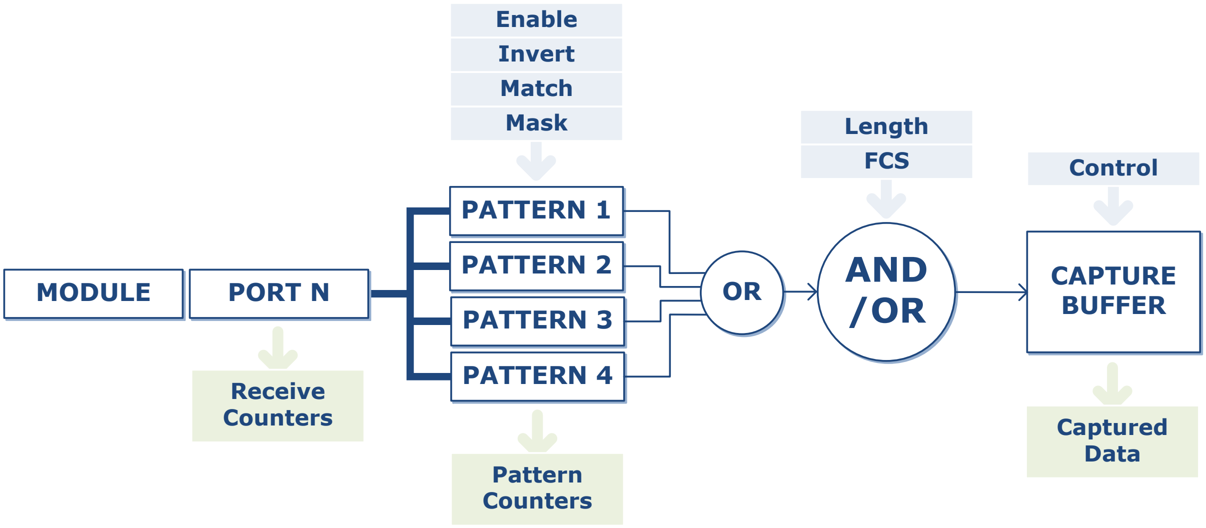
1. Arm **ports** on the connected units with the **apgControlPort STARTSYNCTX** command.
2. Synchronise the start of packet transmission with the synchronisation pulse, either generated locally with the **apgControlUnit SYNCGO** command, or externally from a pulse generator.

The 'synchronised' ports start transmitting on the rising edge of the synchronisation pulse.

The timestamp of the **first** transmitted packet will be within ±1 clock cycle across any port on any unit (APGV1 ±8ns, APGV2 ±2.5ns)

If the units are armed with the **apgControlPort STARTSYNCTX** command and the synchronisation event is **not** sent or detected, the unit control functions will be blocked, waiting for the synchronisation event. The armed state can be removed with the **apgControlUnit SYNCCLEARUP** command.

3.8 RECEIVE PATH



Each port contains a receive path that comprises:

- Module status
- Receive port status and counters
- Configurable Capture Buffer

The port receive counters can be read using **apgGetPort RXSTATS**.

Configurable Capture Filters are not supported in APG TCL API APG V1.3 .1 Software. All received port traffic is forwarded to the capture buffer.

3.9 Packet Capture

A 64KB capture buffer is available per port, that can be enabled using the **apgControlPort PORTCAPTURE** command. The contents of the capture buffer can be downloaded from the unit with **apgLoadPort CAPTURE**, and read with **apgGetPort CAPTURE PACKET**.

The local buffer sizes increased from 16KB and 64KB in the Version 2.1 release

A single port can be enabled to capture to the 'deep' 1GB capture buffer, with the **apgControlPort DEEPCAPTURE** command.

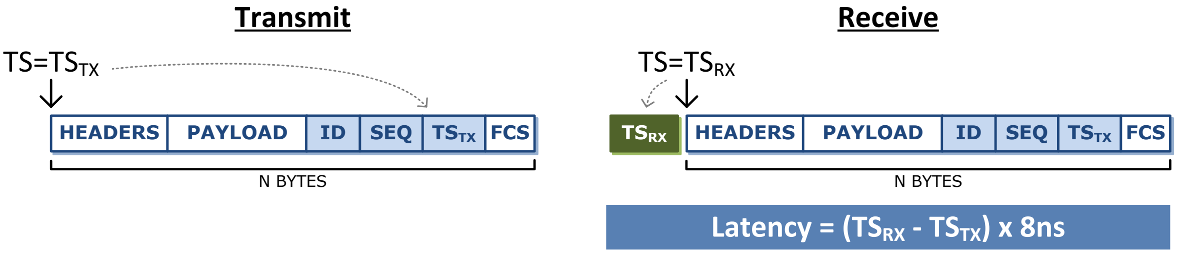
Packet capture is disabled with the **apgControlPort DISABLECAPTURE** command, and cleared with the **apgControlPort CLEARCAPTURE** command.

3.10 Packet Analysis

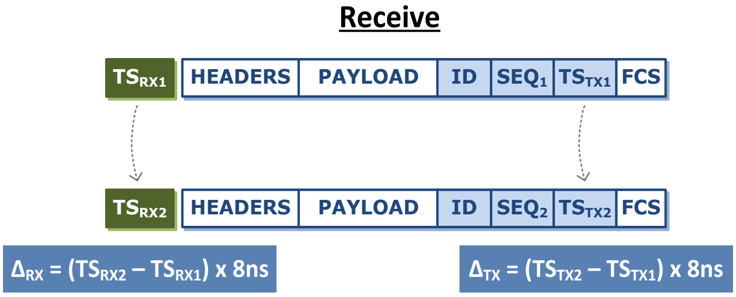
The Packet Analysis function provides basic analysis of the packet timestamp fields from the captured packets, and analysed with the **apgLoadPort ANALYSIS** command.

The 'Timestamp' fields are added to each transmitted packet of a timestamp-enabled stream. The transmit timestamp is the time of the first transmitted bit. The transmit sequence number increments for each packet transmitted from a **port**.

The Timing Analysis provides packet transmit-to-receive timing (latency):



Transmit-to-transmit timing (transmit stability) and receive-to-receive timing (receiver stability) are also performed:



The SEQUENCE numbers are checked for out-of-order packets, gaps and repeated values.

The analysis measurements are cleared using the **apgControlPort CLEARANALYSIS** command. Each time captured data is analysed with **apgLoadPort ANALYSIS**, the results are appended to the existing results.

The analysis results are read using **apgGetPort ANALYSIS**.

4. API Initialisation

The Axtrinet APG TCL API is initialised by sourcing the axtrinetApi.tcl file that contains:

- Path to the APG TCL API
- "package require" declaration
- Import APG TCL API Commands
- Header Definition file processing

4.1 Initialisation

4.1.1 Windows Environment

The APG TCL API is 'sourced' in the Windows environment with the following command:

```
# Instantiate APG TCL API
source "C:/Program Files/Axtrinet/APG/axtrinetApi.tcl"                ← Load API

# Display TCL Version Info
puts "[apgGetApiVersion DESCRIPTION] [apgGetApiVersion VERSION]"      → APG TCL API APG V1.4.1
puts "Build Date [apgGetApiVersion BUILD_DATE]"                       → Build Date 1675361105
puts "Target API Version [apgGetApiVersion API_VERSION]"              → Target API Version 161016
```

4.1.2 Linux Environment

The APG TCL API is 'sourced' in the Linux environment with the following command:

```
# Instantiate APG TCL API
source "/usr/share/axtrinet/apg/axtrinetApi.tcl"                      ← Load API

# Display TCL Version Info
puts "[apgGetApiVersion DESCRIPTION] [apgGetApiVersion VERSION]"      → APG TCL API APG V1.4.1
puts "Build Date [apgGetApiVersion BUILD_DATE]"                       → Build Date 1675361105
puts "Target API Version [apgGetApiVersion API_VERSION]"              → Target API Version 161016
```

4.2 Version Command - apgGetApiVersion

Get (read) the APG TCL API versions.

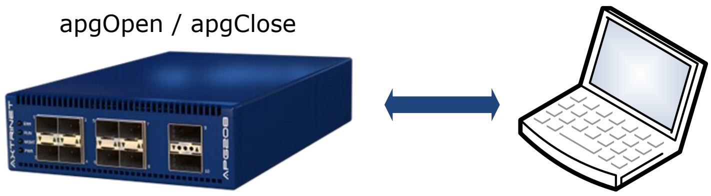
A **LOAD** command is not required before the **apgGetApiVersion**

4.2.1 apgGetApiVersion [VAR]

apgGetApiVersion returns the variable value if successful, otherwise the command will display an error message and exit the TCL environment.

[VAR]	LENGTH	DESCRIPTION	Example
COMPANY	21 char	Xentech Solutions Ltd	
BRAND	8 char	Axtrinet	
DESCRIPTION	11 char	APG TCL API	
VERSION	10 char	Axtrinet TCL API Version currently running	APGV1.4.1
BUILD_DATE	32 bits	Build date of the TCL API	1706631374
API_VERSION	6 char	Target API Version	161016

5. Connection Commands



Connection	**apgOpen [IP-ADDRESS]** **apgUSBOpen [SERIAL]** **apgClose [UNITID]**

Eg:

```
# Instantiate APG TCL API (Windows)
source "C:/Program Files/Axtrinet/APG/axtrinetApi.tcl"            ← Load API

# Open connection to unit at IP address 192.168.1.100
# Returns integer Unit ID
set IPADDRESS 192.168.1.100
set UNITID [apgOpen $IPADDRESS]                                    → UNITID = 1
puts "Opened connection to Unit $UNITID at $IPADDRESS"

# Close connection
apgClose $UNITID                                                   → 1
```

5.1 Open Connection

Open a connection to the Axtrinet Packet Generator over the Ethernet Management Interface using **apgOpen [IP-ADDRESS]**; or the USB Management Interface using **apgUSBOpen [SERIAL]**.

If the connection is successfully opened, **apgLoadUnit STATUS** is polled to determine the hardware status until the unit is READY (**apgLoadUnit STATUS READY** = 1).

The critical unit, port and stream information is then loaded into the local databases using the following commands:

- **apgLoadUnit [UNITID] INFO**
- **apgLoadPort [PORTID] INFO**
- **apgLoadStream [STREAMID]**

If the connection is successfully opened and the critical configuration loaded from the unit, the UNITID of the new unit is returned.

If the connection fails to open, **apgOpen** returns -1, but does not exit the TCL environment. The connection may fail if two APG Control Interface or TCL management connections to the unit, the unit is being upgraded or the unit fails to boot (check front panel LEDs).

5.1.1 apgOpen [IP-ADDRESS]

Opens a connection to the Axtrinet Packet Generator at IP Address [IP-ADDRESS] over the Ethernet Management Interface.

5.1.2 apgUSBOpen [SERIAL]

Opens a connection to the Axtrinet Packet Generator with serial number [SERIAL] over the USB Management Interface.

5.2 CLOSE CONNECTION - APGCLOSE

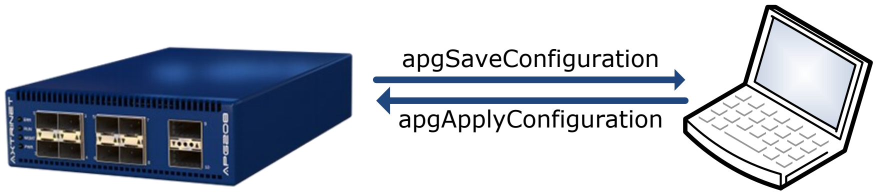
5.2.1 apgClose [UNITID]

Closes the connection to the Axtrinet Packet Generator UNITID.

apgClose returns 1 if successful, otherwise the command will display an error message and exit the TCL environment.

6. Configuration Commands

The configuration commands **apgSaveConfiguration** and **apgApplyConfiguration** allow the unit configuration to be saved to and loaded from an external file.



Configuration **apgSaveConfiguration [UNITID] [PORTLIST] {FILENAME}**
apgApplyConfiguration [UNITID] [FILENAME]

Eg:

```
# Instantiate APG TCL API (Windows)
source "C:/Program Files/Axtrinet/APG/axtrinetApi.tcl"          ← Load API

# Open connection to unit at IP address 192.168.1.100           Open Connection
# Returns integer Unit ID                                        → UNITID = 1
set IPADDRESS 192.168.1.100
set UNITID [apgOpen $IPADDRESS]
puts "Opened connection to Unit $UNITID at $IPADDRESS"

puts "Saving configuration"
set FILENAME APGexample.apg
apgSaveConfiguration $UNITID ALL $FILENAME                       Save Configuration

puts "Closing Connection"
apgClose $UNITID                                                 Close Connection

################################

set UNITID [apgOpen $IPADDRESS]                                  Open Connection
puts "Opened connection to Unit $UNITID at $IPADDRESS"          → UNITID = 2

puts "Loading saved configuration $FILENAME"
apgApplyConfiguration $UNITID $FILENAME                          Load Configuration

puts "Closing Connection"
apgClose $UNITID                                                 Close Connection
```

An example saved configuration file is shown in APPENDIX B - Sample apgSaveConfiguration File for port 1 only.

6.1 Save Configuration

6.1.1 apgSaveConfiguration [UNITID] [PORTLIST] {FILENAME}

Saves the unit UNITID PORTLIST port configuration to FILENAME.

PORTLIST can be "ALL" or in the list of PORTID format, eg `{{ 1 1 } { 1 3 }}`

If FILENAME is not specified, the default filename [SERIAL].config is used where SERIAL is **apgGetUnit [UNITID] INFO SERIAL**.

The saved configuration file contains the expected API_VERSION (**apgGetUnit [UNITID] INFO API_VERSION**) and PRODUCT (**apgGetUnit [UNITID] INFO PRODUCT**)

A 'cut down' example file is shown in Appendix B for a single port and stream. A saved file for the APG208 will contain the configuration for 10 ports each with 8 streams.

The saved configuration can be manually edited to modify the unit configuration.

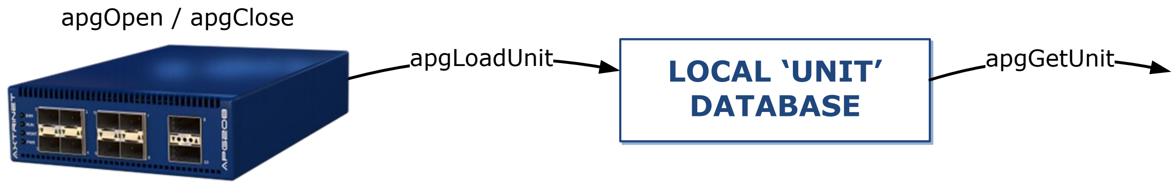
6.2 Apply Configuration

6.2.1 apgApplyConfiguration [UNITID] [FILENAME]

Applies the stored configuration file FILENAME to unit UNITID.

File FILENAME is first scanned for suitability, verifying that the API_VERSION and PRODUCT type match. If there is an API_VERSION mismatch, the commands may not apply. If there is a PRODUCT mismatch, ports and stream configurations may not apply to the unit UNITID.

7. Unit Commands

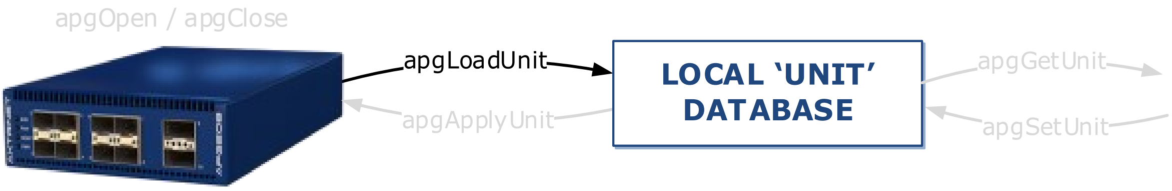


apgLoadUnit	**apgLoadUnit [UNITID] INFO** **apgLoadUnit [UNITID] IPCONFIG** **apgLoadUnit [UNITID] STATUS** **apgLoadUnit [UNITID] CLOCKSTATUS**
	apgLoadUnit [UNITID] PORTSTATUS **apgLoadUnit [UNITID] COUNTERS** **apgLoadUnit [UNITID] RATES** **apgLoadUnit [UNITID] PRBSSTATUS**
apgGetUnit	**apgGetUnit [UNITID] INFO [VAR]** **apgGetUnit [UNITID] IPCONFIG [VAR]** **apgGetUnit [UNITID] STATUS [VAR]** **apgGetUnit [UNITID] CLOCKSTATUS [VAR]** **apgGetUnit [UNITID] TSRESETMODE [VAR]** **apgGetUnit [UNITID] PRBSPORTLIST**
apgSetUnit	**apgSetUnit [UNITID] [VAR] [VAL]**
apgApplyUnit	**apgApplyUnit [UNITID] CLOCKMODE**
apgControlUnit	**apgControlUnit [UNITID] [COMMAND]**

Eg:

```
# Instantiate APG TCL API (Windows)
source "C:/Program Files/Axtrinet/APG/axtrinetApi.tcl"          ← Load API

# Open Connection
set IPADDRESS 192.168.1.100
set UNITID [apgOpen $IPADDRESS]

# Load unit information into local database
apgLoadUnit $UNITID INFO                                        → 1

# Get the unit information from the local database
set TYPE [apgGetUnit $UNITID INFO PRODUCT]                      → APG208
set SN   [apgGetUnit $UNITID INFO SERIAL]                       → APG000006

# Display unit information
puts "Unit $UNITID is an $TYPE with S/N $SN"

# Load unit status into local database
apgLoadUnit $UNITID STATUS                                      → 1

# Display unit status
set UPTIME [apgGetUnit $UNITID STATUS UPTIME]
puts "$SN has been on for [expr $UPTIME / 1000] secs"           → Eg 23000

set READY [apgGetUnit $UNITID STATUS READY]                     → 1
while !$READY {
    puts "Unit $UNITID is still booting"                        ~60sec after turned on
    after 5000
    set READY [apgGetUnit $UNITID STATUS READY]                 → 1
}

# Close connection
apgClose $UNITID                                                → 1
```

7.1 Load Unit Data - apgLoadUnit



Load the UNIT configuration and status from the hardware into the local database.

apgLoadUnit returns 1 if successful, otherwise the command will display an error message and exit the TCL environment.

7.1.1 apgLoadUnit [UNITID] INFO

Loads the unit information into the local database, containing:

- Unit Product ID and Serial Number
- Hardware and Software Versions
- Port Count

The unit information can be read using **apgGetUnit INFO**.

7.1.2 apgLoadUnit [UNITID] IPCONFIG

Loads the unit IP configuration into the local database, containing the IP Address, Mask and Gateway.

The IP Configuration can be read using **apgGetUnit IPCONFIG**.

7.1.3 apgLoadUnit [UNITID] STATUS

Loads the unit status into the local database, containing:

- Uptime (ms)
- Temperature, Fan and Status Flags

The unit status can be read using **apgGetUnit STATUS**.

7.1.4 apgLoadUnit [UNITID] CLOCKSTATUS (APGV2 Only)

Loads the unit clock status into the local database, containing:

- Clock Input and Output Modes
- PLL State

The unit status can be read using **apgGetUnit CLOCKSTATUS**.

7.1.5 apgLoadUnit [UNITID] PORTSTATUS

Loads the unit port status into the local database, containing:

- Link Mode, Status and Speed
- Module Type
- Port Label

The port status can be read using **apgGetPort STATUS**.

7.1.6 apgLoadUnit [UNITID] COUNTERS

Loads the transmit and receive counters for all ports in the unit into the local database, containing:

- Time that the counter is read
- Number of Bytes
- Number of Good Packets
- Number of Underrun Packets
- Number of Packet Fragments
- Number of Packets with Frame Checksum (FCS) Errors
- Number of Frames with No Start-of-Frame Delimiter (SFD)
- Frame Rate (pps)

Port transmit counters can be read using **apgGetPort TXSTATS**.

Port receive counters can be read using **apgGetPort RXSTATS**.

7.1.7 apgLoadUnit [UNITID] RATES {DURATION}

The transmit and receive rates are calculated when **apgLoadUnit RATES** is called.

The rate calculator takes two counter readings, separated by {DURATION} milliseconds (eg 2000 = 2sec), to calculate the transmit and receive frame, byte, bit and error rates. If {DURATION} is omitted, a duration of 1000ms is assumed.

The transmit and receive rates are stored in the local database, containing:

- Transmit Frame, Byte and Bit Rates
- Receive Frame, Byte, Bit and Error Rates

Port rates can be read using **apgGetPort RATES**.

7.1.8 apgLoadUnit [UNITID] PRBSSTATUS (APGV2 Only)

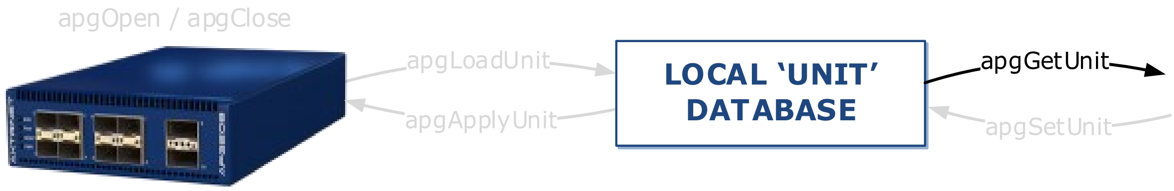
Loads the PRBS configuration and status for all ports in the unit into the local database containing:

- PRBS Mode (Disabled, Transmit Only, Receive only, Transmit and Receive)
- PRBS Pattern (PRBS7, PRBS9, PRBS15, PRBS23, PRBS31)
- Receiver Locked
- Runtime since locked (in milliseconds)
- Bit Errors since locked

A list of ports that support PRBS BERT mode can be read using **apgGetUnit PORTLIST**.

The PRBS configuration and status can read using **apgGetPort PRBS**.

7.2 Get Unit Data - apgGetUnit



Get (read) the UNIT configuration and status from the local database.

apgGetUnit returns the variable value if successful, otherwise the command will display an error message and exit the TCL environment.

7.2.1 apgGetUnit [UNITID] INFO [VAR]

The APG unit information can be read with the **apgGetUnit INFO** command.

The apgGetUnit INFO command must be preceded by at least one **apgLoadUnit INFO**, otherwise the command will display an error message and exit the TCL environment.

The unit INFO is static for the duration of the test session, and should only need to be read once at the start of the test.

[VAR]	LENGTH	DESCRIPTION
API_VERSION	32 bits	API Version of the Firmware eg 161016
PORT_COUNT	8 bits	Number of physical ports on the unit
API_MINIMUM	32 bits	Minimum API supported by the unit **[apgGetApiVersion API_VERSION]** must be greater than **API_MINUMUM**, and should ideally match.
SERIAL	16 chars	Serial Number eg APG000006
PRODUCT	12 chars	Product Code eg APG208
FW_VERSION	8 chars	Firmware Version loaded onto the UNIT eg 1.2-1
FW_BUILDDATE	32 bits	Build timestamp of the Firmware Version loaded onto the UNIT eg 1457534868
FPGA_VERSION	8 chars	Firmware Version loaded onto the UNIT eg 6.010E
FPGA_BUILDDATE	32 bits	Build timestamp of the Firmware Version loaded onto the UNIT eg 1455890890
HW_VERSION	8 chars	Hardware Version of the Unit eg B.00.00

7.2.2 apgGetUnit [UNITID] IPCONFIG [VAR]

The APG unit status can be read with the **apgGetUnit IPCONFIG** command.

The **apgGetUnit IPCONFIG** command must be preceded by at least one **apgLoadUnit IPCONFIG**, otherwise the command will display an error message and exit the TCL environment.

The unit IPCONFIG is static for the duration of the test session, and should only need to be read once at the start of the test.

[VAR]	LENGTH	DESCRIPTION	
IPADDRESS	32 bits	IP Address	eg 0xC0A8C8A7 = 192.168.200.167
IPMASK	32 bits	IP Mask	eg 0xFFFFFF00 = 255.255.255.0
IPGATEWAY	32 bits	IP Gateway	eg 0xC0A8C801 = 192.168.200.1

It is not possible to set the IP Configuration with the APG TCL API.
The IP Configuration can only be set from the APG Control Interface.

7.2.3 apgGetUnit [UNITID] STATUS [VAR]

The APG unit status can be read with the **apgGetUnit STATUS** command.

The **apgGetUnit STATUS** command must be preceded by at least one **apgLoadUnit STATUS**, otherwise the command will display an error message and exit the TCL environment.

The unit STATUS is dynamic, and should be re-loaded before reading.

[VAR]	LENGTH	DESCRIPTION
UPTIME	64 bits	Time since the unit was powered up (in ms)
TEMP	2 bits	Temperature Flags, calculated from the preset $T_{OPERATE}$ (58°C), $T_{WARNING}$ (65°C) and $T_{SHUTDOWN}$ (85°C) temperature thresholds: 1 = OK $T \le T_{OPERATE} + 3$ 2 = HIGH $T_{OPERATE} + 3 \le T < T_{WARNING}$ 3 = WARNING $T_{WARNING} \le T < T_{SHUTDOWN}$ 4 = ALARM $T > T_{WARN}$ The temperature thresholds are set by the unit firmware and cannot be changed.
FAN	1 bit	1 = OK 2 = WARNING (fan speed > 8000rpm) 3 = ALARM (fan has failed or is running at full speed)
SELFTEST	1 bit	0 = Self-Test Failed 1 = Self-Test Passed Self-Test verifies that the internal power supplies are within tolerance and that the CPU interfaces are operational.
READY	1 bit	0 = Unit not ready (FPGA Booting) 1 = Unit Ready (FPGA Booted)

7.2.4 apgGetUnit [UNITID] CLOCKSTATUS [VAR] (APGV2 Only)

The APG unit clock status can be read with the **apgGetUnit CLOCKSTATUS** command.

The **apgGetUnit CLOCKSTATUS** command must be preceded by at least one **apgLoadUnit CLOCKSTATUS**, otherwise the command will display an error message and exit the TCL environment.

If using the EXT1PPS clock mode, PLLSTATE reports the dynamic internal PLL state, and should be re-loaded before reading.

[VAR]	LENGTH	DESCRIPTION
CLKINMODE	3 bits	0 = Not Set (use default) 1 = Internal (default) 3 = External 1PPS (1Hz) 4 = External 1MHz
CLKOUTMODE	3 bits	0 = Not Set (use default) 1 = 1PPS (1Hz) output (default) 2 = 1MHz output
PLLSTATE	4 bits	0 = Invalid 1 = Unknown 2 = Internal 3 = Holdover 4 = Acquiring 5 = Tracking 6 = Locked

If using an 'External' clocking mode, the PLL Status will progress through HOLDOVER → ACQUIRING → TRACKING → LOCKED states while the APG system clocks lock to the input clock.

The internal system clocks will lock to a 1PPS input clock in ~2 minutes.
The internal system clocks will lock to a 1MHz input clock in ~2 seconds.

7.2.5 apgGetUnit [UNITID] TSRESETMODE (APGV2 Only)

The APG unit timestamp reset mode can be read with the **apgGetUnit TSRESETMODE** command.

The TSRESETMODE is a local TCL API variable that is used to calculate the timestamp reset value for the **apgControlUnit TSCLEAR** or **apgControlUnit TSSYNCCLEAR** commands, and is not stored on the APG unit.

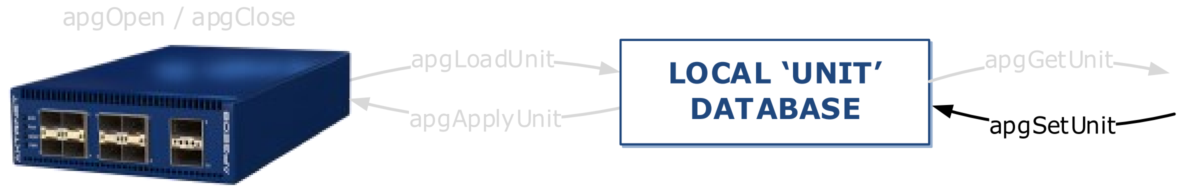
apgGetUnit TSRESETMODE returns the **apgSetUnit TSRESETMODE** setting "ZERO" or "TIME".

7.2.6 apgGetUnit [UNITID] PRBSPORTLIST (APGV2 Only)

A list PRBS enabled ports can be read with the apgGetUnit PRBSPORTLIST command.

The **apgGetUnit PRBSPORTLIST** command must be preceded by at least one **apgLoadUnit PRBSSTATUS**, otherwise the command will display an error message and exit the TCL environment.

7.3 Set Unit Data – apgSetUnit (APGv2 Only)



apgSetUnit returns the set value if successful, otherwise the command will display an error message and exit the TCL environment.

The unit configuration is applied to the unit using **apgApplyUnit**.

Eg:

```
# Instantiate APG TCL API (Windows)
source "C:/Program Files/Axtrinet/APG/axtrinetApi.tcl"          ← Load API

# Open Connections to 2 units
set UNITID1 [apgOpen 192.168.1.100]
set UNITID2 [apgOpen 192.168.1.101]

# Load unit information into local database
apgLoadUnit $UNITID1 INFO                                        → 1
apgLoadUnit $UNITID2 INFO

# Set UNIT1 to INTernal clock mode (clock master)
# and generate a 1MHZ output clock on CLKOUT
apgSetUnit $UNITID1 CLKINMODE INT
apgSetUnit $UNITID1 CLKOUTMODE 1MHZ
apgApplyUnit $UNITID1 CLOCKMODE                                  → 1

# Set UNIT2 to EXTernal clock mode (clock slave)
# expecting 1MHZ clock
apgSetUnit $UNITID2 CLKINMODE EXT1MHZ
apgSetUnit $UNITID2 CLKOUTMODE NONE
apgApplyUnit $UNITID2 CLOCKMODE                                  → 1

# Check Clock Status is LOCKED                                   ← Wait
after 1000                                                       ← Load Clock Status
apgLoadUnit $UNITID2 CLOCKSTATUS                                 ← Get Clock Status
set CLOCKSTATE [apgGetUnit $UNITID2 CLOCKSTATUS PLLSTATE]
if { $CLOCKSTATE != 6 } { error "Not locked after 1sec" }       ← Check if locked

# Reset timestamps to zero
foreach UNITID { $UNITID1 $UNITID2 } {
    apgSetUnit $UNITID TSRESETMODE ZERO                          ← Set mode to ZERO
    apgControlUnit TSSYNCCLEAR $UNITID                           ← Arm units
}
# Synchronise timestamps from APG1
apgControlUnit SYNCGO $UNITID1                                   ← SYNC timestamps

# Synchronise Transmit on Port1 on both units
set PORTLIST {{1 1 0} {2 1 0}}
apgControlPort STARTSYNCTX $PORTLIST                             ← Arm units
# Synchronise transmit from APG1
apgControlUnit SYNCGO $UNITID1                                   ← SYNC transmit

# Stop Traffic
after 1000
apgControlPort STOPTX $PORTLIST                                  ← STOP transmit

# Close connection
foreach UNITID { $UNITID1 $UNITID2 } {apgClose $UNITID}          → 1
```

7.3.1 apgSetUnit [UNITID] [VAR] [VAL] (APGV2 Only)

7.3.1.1 Clock Modes

Sets the clock input and output modes (Section 3.7.1)

[VAR]	[VAL]	Default
CLKINMODE	**INT** = Use internal free-running clock with 1pps on the CLOCK-OUT interface (default) **EXT1PPS** = Synchronise internal clocks to external 1PPS on CLOCK-IN interface **EXT1MHZ** = Synchronise internal clocks to external 1MHZ on CLOCK-IN interface	**INT**
CLKOUTMODE	**1PPS** = Output 1PPS (1Hz) on the CLOCK-OUT interface **1MHZ** = Output 1MHz on the CLOCK-OUT interface	**1PPS**

The **apgSetUnit** clock modes command must be followed with **apgApplyUnit CLOCKMODE** to apply the clock mode onto the unit.

The clock mode configuration and PLL status can be read with the **apgGetUnit CLOCKSTATUS** command.

7.3.1.2 Timestamp Reset

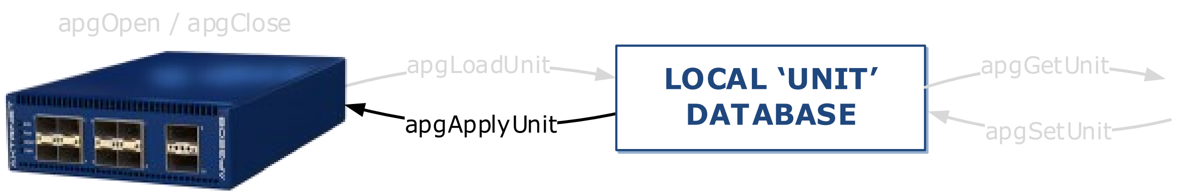
Sets the timestamp reset mode (see Section 3.7.2)

[VAL]	DESCRIPTION	Default
TSRESETMODE	**ZERO** = Zero the timestamp on reset **TIME** = Set the timestamp to the current time on reset	**ZERO**

The TSRESETMODE is a local TCL API variable that is used to calculate the timestamp reset value for the **apgControlUnit TSCLEAR** or **apgControlUnit TSSYNCCLEAR** commands, and is **not** stored on the APG unit. It cannot be applied to the unit.

‘Current Time’ is the Unix Time in APG unit clock cycles (Timestamp Resolution).
On a APGV2 unit, the timestamp resolution is 2.5ns.
The timestamp will be reset to (Unix Time x 1/2.5ns) = Unix Time x 400,000,000

7.4 Apply Unit Data - apgApplyUnit



apgApplyUnit returns 1 if successful, otherwise the command will display an error message and exit the TCL environment.

7.4.1 apgApplyUnit [UNITID] CLOCKMODE

The **apgApplyUnit CLOCKMODE** command must be used after **apgSetUnit** command to apply the clock mode configuration change to the unit.

7.5 CONTROL COMMANDS – APGCONTROLUNIT (APGV2 ONLY)

7.5.1 apgControlUnit [COMMAND] [UNITID]

The **apgControlUnit [COMMAND]** commands have immediate effect.

apgControlUnit returns 1 if successful, otherwise the command will display an error message and exit the TCL environment.

The immediate unit commands are:

[COMMAND]	Description
TSCLEAR	Immediately reset the timestamp on [UNITID]. The timestamp is reset to ZERO or TIME with the **apgSetUnit TSRESETMODE** command.

7.5.2 apgControlUnit [SYNC COMMAND] [UNITID]

The **apgControlUnit [SYNC COMMAND]** command is used to control the synchronous transmit and timestamp reset functions.

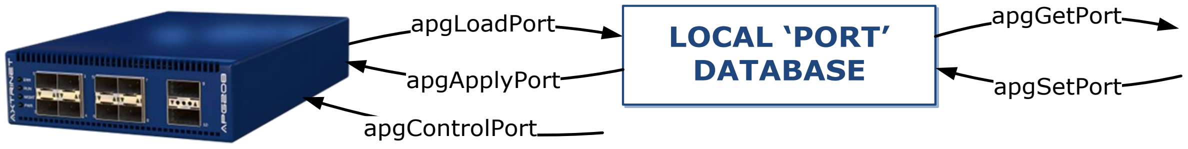
apgControlUnit returns 1 if successful, otherwise the command will display an error message and exit the TCL environment.

The synchronous control commands are:

[COMMAND]	Description
TSSYNCCLEAR	Arm synchronous timestamp reset on [UNITID]. To synchronously reset the timestamps on multiple units, the SYNC interfaces on all units must be connected together with co-axial cables. Synchronous timestamp reset is started on the rising edge of a pulse on the SYNC interface, either from a connected unit with the **apgControlUnit SYNCGO** command; or from an external synchronisation source. The timestamp is reset to ZERO or TIME with the **apgSetUnit TSRESETMODE** command. If the timestamp reset mode is TIME, it is recommended that the **TSSYNCCLEAR** is immediately followed by a locally generated synchronisation pulse from the **apgControlUnit SYNCGO** command to synchronise the unit timestamp close to the host PC time. The armed state can be cleared with the **apgControlUnit SYNCCLEARUP** command. See Section 3.7.2 and the APG User Guide [1] for more information. ⚠ **A unit will stop responding to unit commands until the synchronise event (SYNCGO or external) occurs.** **During script development, this may leave the unit in a state where the script aborts with a "Unit Busy" error message.** **During script development, it is recommended that an apgControlUnit SYNCCLEARUP is used at the start of the script to clear any armed states.**

[COMMAND]	Description
SYNCGO	Generates a synchronisation pulse from [UNITID]. **SYNCGO** generates a single pulse on the rear SYNC interface to trigger the synchronous action on all connected units with primed ports. Synchronous transmission is armed with the **apgControlPort STARTSYNCTX** command. Synchronous timestamp reset is armed with the **apgControlUnit TSSYNCCLEAR** command. See Sections 3.7.2, 3.7.33.7.3 and the APG User Guide [1] for more information.
SYNCCLEARUP	Clear the armed state.

8. PORT COMMANDS



apgLoadPort	**apgLoadPort [PORTID] INFO** **apgLoadPort [PORTID] MODULE** **apgLoadPort [PORTID] CAPTURE** **apgLoadPort [PORTID] ANALYSIS**
apgGetPort	**apgGetPort [PORTID] INFO [VAR]** **apgGetPort [PORTID] STATUS [VAR]** **apgGetPort [PORTID] MODULE [VAR]** **apgGetPort [PORTID] TXSTATS [VAR]** **apgGetPort [PORTID] RXSTATS [VAR]** **apgGetPort [PORTID] RATES [VAR]** **apgGetPort [PORTID] PRBS** **apgGetPort [PORTID] CAPTURE CONFIG [VAR]** **apgGetPort [PORTID] CAPTURE PACKET TOTALPACKETS** **apgGetPort [PORTID] CAPTURE PACKET [VAR] [PKTNUM]** **apgGetPort [PORTID] ANALYSIS [TYPE] [VAR]**
apgSetPort	**apgSetPort [PORTID] [VAR] [VAL]**
apgApplyPort	**apgApplyPort [PORTID] STATE** **apgApplyPort [PORTID] COPPERMODULE** **apgApplyPort [PORTID] PRBS**
apgControlPort	**apgControlPort [COMMAND] [PORTLIST]** **apgControlPort [CAPTURE] [PORTLIST]** **apgControlPort CLEARANALYSIS [PORTLIST]**

Where PORTID = {UNIT.PORT} or {UNIT.PORT.SUBPORT}

Eg:

```
# Instantiate APG TCL API (Windows)
source "C:/Program Files/Axtrinet/APG/axtrinetApi.tcl"
```

← Load API

```
# Open Connection
set IPADDRESS 192.168.1.100
set UNITID [apgOpen $IPADDRESS]
set PORTLIST ""
```

Open Connection

```
# Generate Port List
puts "\nDEMO: Generating Port List for Unit $UNITID"
apgLoadUnit $UNITID INFO
set PORT_COUNT [apgGetUnit $UNITID INFO PORT_COUNT]
for { set PO 1 } { $PO <= $PORT_COUNT } { incr PO } {
  set PORTID [list $UNITID $PO]
  apgLoadPort $PORTID INFO
  set NSUBPORTS [apgGetPort $PORTID INFO NSUBPORTS]
  for { set SP 0 } { $SP <= $NSUBPORTS } { incr SP } {
    set SUBPORTID [list $UNITID $PO $SP]
    if { $SP > 0 } { apgLoadPort $SUBPORTID INFO }
    if { [apgGetPort $SUBPORTID INFO NSUBPORTS] == 0 } {
      lappend PORTLIST $SUBPORTID
    }
  }
}
puts "      -> $PORTLIST"
```

Generate Port list

All ports

All subports

```
# Configure Streams                                                             Configure
puts "\nDEMO: Configure Streams"                                                Streams
set STREAM 0
foreach PORTID $PORTLIST {
  scan $PORTID "%d %d %d" UN PO SP
  puts "      -> Port $PO"
  for { set ST 0 } { $ST < 8 } { incr ST } {
    set STREAMID [linsert $PORTID end $ST]
    if { $ST == 0 } {                                                           Stream 0
      apgSetStream $STREAMID DEFAULT
      apgSetStream $STREAMID CONFIG ENABLE 1                                    Enable
      apgSetStream $STREAMID CONFIG SIZE_MODE RANDOM                            Random Size
      apgSetStream $STREAMID CONFIG PACKET_SIZE 64                              64 -
      apgSetStream $STREAMID CONFIG PACKET_SIZE_MAX 5000                         5000 bytes
      apgSetStream $STREAMID HEADER HEADER_LIST "MACHEADER"                     MACHEADER
      apgSetStream $STREAMID HEADER MACHEADER DA \
        [format "08:%02d:%02d:%02d:00" $UN $PO $SP $ST]
      apgSetStream $STREAMID HEADER MACHEADER SA \
        [format "08:%02d:%02d:%02d:01" $UN $PO $SP $ST]
      apgSetStream $STREAMID PAYLOAD DATA_CONTROL RANDOM                        Random Data
      if { $PO >= 9 } {                                                         If 40G
        apgSetStream $STREAMID PAYLOAD DATA_CONTROL INCREMENT                    Increment
        apgSetStream $STREAMID PAYLOAD PAYLOAD_DATA "01 02 03 04 05 06 07 08"     Data
      }
    } else {
      apgSetStream $STREAMID CONFIG ENABLE 0                                    Disable 1-7
    }
    apgApplyStream $STREAMID                                                    Apply Config
  }
}
```

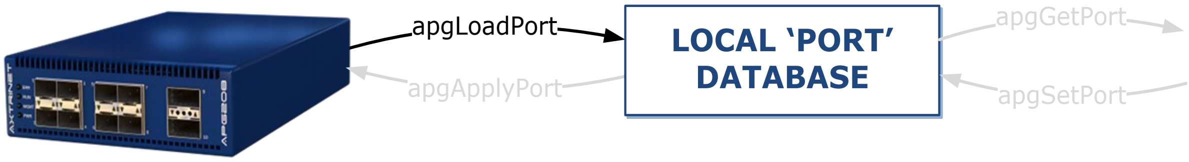
```
# Port Transmit & Capture Control                                               TX Control
puts "\nDEMO: Generating Packets for 3 seconds"
apgControlPort STOPTX $PORTLIST
apgControlPort CLEARCOUNTERS $PORTLIST                                          Clear counts
apgControlPort STARTTX $PORTLIST                                                Start TX
after 1000
puts "DEMO: Calculating data rates over 1000ms"                                 Calculate
apgLoadUnit $UNITID RATES 1000                                                  Rates
puts "DEMO: Clearing Capture Buffers and enabling Capture on all ports "
apgControlPort PORTCAPTURE $PORTLIST
after 1000                                                                      Enable Capture
puts "DEMO: Stopping Traffic"
apgControlPort STOPTX $PORTLIST                                                 Stop TX
```

```
# Port Counters                                                                 Counters
puts "\nDEMO: Display Port Counters"
apgLoadUnit $UNITID COUNTERS                                                    Load
foreach PORTID $PORTLIST {
  scan $PORTID "%d %d %d" UN PO SP
  set TXPKT  [apgGetPort $PORTID TXSTATS GOOD_PACKETS]                          Get counters
  set TXRATE [expr [apgGetPort $PORTID RATES TXBITRATE] / 1000000.0]
  set RXPKT  [apgGetPort $PORTID RXSTATS GOOD_PACKETS]
  if { $TXPKT > 0 } {
    puts [format "      Port %2d TX %7d at %8.2f Mbps, RX %7d packets" \
      $PO $TXPKT $TXRATE $RXPKT]
  }
}
```

```
# Port Capture                                                                  Capture
puts "\nDEMO: Display Captured Packets"
foreach PORTID $PORTLIST {
  scan $PORTID "%d %d %d" UN PO SP
  apgLoadPort $PORTID CAPTURE CONFIG                                            Load config
  set AVAILABLE [apgGetPort $PORTID CAPTURE CONFIG AVAILABLE ]
  if { $AVAILABLE > 0 } {                                                       If data ...
    apgLoadPort $PORTID CAPTURE BUFFER                                          Load buffer
    puts "\n      Port $PO:\n"
    set PKTCOUNT [ apgGetPort $PORTID CAPTURE PACKET TOTALPACKETS ]
    for { set PKT 1 } { $PKT <= $PKTCOUNT } { incr PKT } {
      set LENGTH [apgGetPort $PORTID CAPTURE PACKET LENGTH $PKT]                Display data
      puts -nonewline [format "      %3d: %5d bytes -> " $PKT $LENGTH]
      set DATA [apgGetPort $PORTID CAPTURE PACKET DATA $PKT]
      foreach BYTE $DATA { puts -nonewline [format "%02X " $BYTE] }
      puts ""
    }
  }
}
```

```
# Close connection                                                              Close
apgClose $UNITID                                                                connection
```

8.1 Load Port Data - apgLoadPort



Loads the PORT configuration and status from the hardware into the local database.

apgLoadPort returns 1 if successful, otherwise the command will display an error message and exit the TCL environment.

8.1.1 apgLoadPort [PORTID] INFO

Loads the port information into the local database, containing:

- Port Topology (eg 10Gbps or 1Gbps, 40Gbps or 4x10Gbps)
- Impact of Topology Change flags
- Number of Subports (eg 1 for 10Gbps/1Gbps Ports, 4 for 40Gbps Ports)
- Maximum Number of Streams (8)

The port information can be read using **apgGetPort INFO**.

8.1.2 apgLoadPort [PORTID] MODULE

Loads the module information into the local port database.

Only the ID, VENDOR and DIAGNOSTIC information is available for fibre modules and Direct Attach cables. Additional information is available for copper modules.

The port information can be read using **apgGetPort MODULE**.

8.1.3 apgLoadPort [PORTID] CAPTURE [VAR] {NUMPKT}

Loads the contents of the port capture configuration or buffer into the local port database.

[VAR]	DESCRIPTION
CONFIG	Load the port capture configuration into the local port database.
BUFFER	Load the port capture buffer into the local port database {NUMPKT} is an optional field to define the number of packets to be downloaded from the unit, starting from the first captured packet. If {NUMPKT} is not declared, **<u>all</u>** captured packets are downloaded from the unit.

The port information can be read using **apgGetPort CAPTURE**.

apgLoadPort CAPTURE BUFFER takes <1sec to download and process a full 64KB 10Gbps port capture, and <2secs to download and process a full 256KB 40Gbps port capture.

On a Version 1 unit, a full 64MB DEEP CAPTURE buffer will take ~12min to download.
On a Version 2 unit, a full 128MB DEEP CAPTURE buffer will take ~10min to download.

8.1.4 apgLoadPort [PORTID] ANALYSIS {FIRSTPKT} {NUMPKT} {TSID}

The **apgLoadPort ANALYSIS** function performs port analysis calculations on the captures packets. The **ANALYSIS** function **must** be preceded by a **apgLoadPort CAPTURE BUFFER**.

It is recommended to use the FIRSTPKT and NUMPKT arguments when analysing packets from the deep capture buffer, and keep NUMPKT < 10000. A 10,000 packet analysis will take ~150 seconds.

If {FIRSTPKT} and {NUMPKTS} are excluded, all captured packets are be analysed.

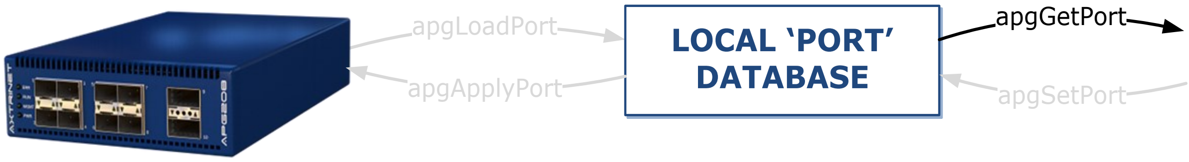
If {NUMPKTS} is excluded, all packets between {FIRSTPKT} and the end of the capture buffer are be analysed.

Only packets that match the timestamp identifier {TSID} are processed. If no {TSID} value is set, the timestamp ID field is ignored.

The latency measurements are stored in the local database, containing transmit-to-transmit, transmit-to-receive and receive-to-receive timing measurements.

Packet latency measurements can be read using **apgGetPort ANALYSIS**.

8.2 GET PORT DATA - APGGETPORT



apgGetPort returns the variable value if successful, otherwise the command will display an error message and exit the TCL environment.

8.2.1 apgGetPort [PORTID] INFO [VAR]

The port information can be read using the **apgGetPort INFO** command.

The **apgGetPort INFO** command must be preceded by at least one **apgLoadPort INFO**, otherwise the command will display an error message and exit the TCL environment.

Port information is relatively static during a test and should only need to be read once at the start of the test, or if the port transceiver or topology is changed.

[VAR]	LENGTH	DESCRIPTION
TOPOLOGY	16 bits	Current port topology: Bit 0-7 = Reserved Bit 8 = Blocked (0x0100) Bit 9 = 1Gbps (0x0200) Bit 10 = Reserved Bit 11 = 10Gbps (0x0800) Bit 12 = 4x10Gbps (0x1000) Bit 13 = 40Gbps (0x2000) Bit 14-15 = Reserved
TOPOLOGY_CAP	16 bits	Port topology capabilities: Bit 0-7 = Reserved Bit 8 = Blocked (0x0100) Bit 9 = 1Gbps (0x0200) Bit 10 = Reserved Bit 11 = 10Gbps (0x0800) Bit 12 = 4x10Gbps (0x1000) Bit 13 = 40Gbps (0x2000) Bit 14-15 = Reserved
NSUBPORTS	8 bits	Number of subports associated with the port
NSTREAMS	8 bits	Maximum number of streams that can be configured on the port

8.2.2 apgGetPort [PORTID] STATUS [VAR]

The port status can be read using the **apgGetPort STATUS** command.

The **apgGetPort STATUS** must always be preceded by at least one **apgLoadUnit PORTSTATUS**, otherwise the command will display an error message and exit the TCL environment.

The **apgGetPort STATUS** is dynamic during a test as the link status changes, and should always be preceded by **apgLoadUnit PORTSTATUS**

[VAR]	LENGTH	DESCRIPTION
LINK	8 bits	0 = No Link 1 = 'Internal' Link OK 2 = 'External' Link OK (copper SFP+ only) 4 = PRBS Mode
MODULE_STATUS	8 bits	Bit 0 = Module Present Bit 1 = Module Fault Bit 2 = LOS Bit 3 = Reserved Bit 4 = Transmit Enable Bit 5 = Rate Select 0 Bit 6 = Rate Select 1 Bit 7 = Reserved
SPEED	16 bits	Bit 0 = 10Mbps Bit 1 = 100Mbps Bit 2 = 1Gbps Bit 3 = 10Gbps Bit 4 = 40Gbps Bit 5 = 100Gbps Bit 6 = 4x10Gbps Bit 7 = 2.5Gbps Bit 8 = 5Gbps Bit 9-15 = Reserved Note that SPEED returns the expected port speed if the link is down, and the actual port speed if the link is up.
SPEED_ABILITY	16 bits	Bit 0 = 10Mbps Bit 1 = 100Mbps Bit 2 = 1Gbps Bit 3 = 10Gbps Bit 4 = 40Gbps Bit 5 = 100Gbps Bit 6 = 4x10Gbps Bit 7 = 2.5Gbps Bit 8 = 5Gbps Bit 9-15 = Reserved

[VAR]	LENGTH	DESCRIPTION
MODULE_SEQUENCE	16 bits	Module Sequence ID → Number of times a module has been inserted. Used to determine whether the module has changed.
MODULE_TYPE	8 bits	0 = No Module 1 = SFP/SFP+ 2 = QSFP+
MODULE_CLASS	8 bits	0 = None 1 = Fibre 2 = Copper 3 = Passive Direct Attach 4 = Active Direct Attach
MODULE_TEMP	16 bits	Module Temperature (°C) Measured in 1/256ths of a degree. (ie divide by 256 to get measured temperature)
LINKMODE	8 bits	Current MAC-to-Module PMA Link Mode: Bit 0 = 1000Base-X (SFP Port only) Bit 1 = SGMII (SFP+ Port only) Bits 3-4 = Reserved Bit 5 = SRLR (QSFP+ only) Note that the unit software will automatically choose the preferred link mode based on the transceiver, but can be over-written by the user.
LINKMODE_CAP	8 bits	MAC-to-Module PMA Link Mode Capabilities: Bit 0 = 1000Base-X (SFP+ Port only) Bit 1 = SGMII (SFP+ Port only) Bits 3-4 = Reserved Bit 5 = SRLR (QSFP+ only)
TIMESTAMPID	32 bits	Timestamp ID field
PORTNAME	12 char	Port Name

8.2.3 apgGetPort [PORTID] MODULE [VAR]

The module information can be read using the **apgGetPort MODULE** command.

The **apgGetPort MODULE** must always be preceded by at least one **apgLoadPort MODULE**, otherwise the command will display an error message and exit the TCL environment.

The **apgGetPort MODULE** is dynamic during a test as the link status changes, and should always be preceded by **apgLoadPort MODULE**.

[VAR]	LENGTH	DESCRIPTION
MODULE_TYPE	8 bits	0 = No Module 1 = SFP/SFP+ 2 = QSFP+
ID	96 chars	Serial ID Fields
VENDOR_ID	32 chars	Vendor-Specific ID field

The following variables are extracted from the ID Fields:

[VAR]	LENGTH	DESCRIPTION
VENDOR_NAME	16 char	Manufacturer
VENDOR_PN	16 char	Manufacturer's Part Number
VENDOR_REV	4 char	Manufacturer's Revision
VENDOR_SN	16 char	Manufacturer's Serial Number
DATECODE	64 bits	Manufacturing Date

If a copper SFP+/SFP module is fitted (MODULE_CLASS = 2 Copper) additional information is available:

[VAR]	LENGTH	DESCRIPTION
VALID	8 bits	0 = Copper Module Not Fitted 1 = Data Valid (Copper module fitted)

If **VALID** is set, the following variables are valid:

[VAR]	LENGTH	DESCRIPTION
LINK	8 bits	0 = No Link 1 = Port Link OK 2 = Port Module Link Ok (internal)
FAULT	8 bits	0 = OK 1 = Fault
Resolved settings:		
RESOLVED	8 bits	0 = Not Resolved (auto-negotiating) 1 = Resolved (auto-negotiation complete)
RESOLVED_SPEED	8 bits	1 = 10Mbps 2 = 100Mbps 4 = 1Gbps 8 = 10Gbps
RESOLVED_DUPLEX	8 bits	1 = Half Duplex 2 = Full Duplex

[VAR]	LENGTH	DESCRIPTION
RESOLVED_MDI	8 bits	2 = MDI 4 = MDIX
RESOLVED_MS	8 bits	4 = Master 8 = Slave
PAIRSWAP	8 bits	0 = Not Swapped 1 = Swapped

Configuration Settings:

CONFIG_SPEED_DUPLEX	8 bits	1 = Auto-Negotiation Enabled 2 = Fixed Mode, Auto-Negotiation Disabled Note: If 0, ignore configuration
CONFIG_FORCED_SPEED	8 bits	Bit 0 = 10Mbps Half Duplex Bit 1 = 10Mbps Full Duplex Bit 2 = 100Mbps Half Duplex Bit 3 = 100Mbps Full Duplex Bit 4 = 1000Mbps Half Duplex Bit 5 = 1000Mbps Full Duplex Bit 6 = Reserved Bit 7 = 10Gbps Full Duplex Bit 8 = 2G5bps Full Duplex Bit 9 = 5Gbps Full Duplex Note: If 0, forced speeds not set
CONFIG_ADV_SPEED	16 bits	Bit 0 = Advertise Valid (otherwise settings ignored) Bit 1 = Advertise 10Mbps Half Duplex Bit 2 = Advertise 10Mbps Full Duplex Bit 3 = Advertise 100Mbps Half Duplex Bit 4 = Advertise 100Mbps Full Duplex Bit 5 = Advertise 1000Mbps Half Duplex Bit 6 = Advertise 1000Mbps Full Duplex Bit 7 = Advertise 10Gbps Full Duplex Bit 8 = Advertise 2.5Gbps Full Duplex Bit 9 = Advertise 5Gbps Full Duplex Note: Advertised capabilities only valid if Bit 0 is set
CONFIG_ADV_PAUSE	8 bits	Bit 0 = Not used Bit 1 = Advertise Symmetric Pause Bit 2 = Advertise Asymmetric Pause
CONFIG_MDI	8 bits	0 = Not Known 1 = Automatic 2 = Force MDI 4 = Force MDIX (Crossover)
CONFIG_MS	8 bits	0 = Ignore 1 = Automatic, prefer MASTER 2 = Automatic, prefer SLAVE 4 = Force MASTER 8 = Force SLAVE

Capabilities:

CAP_SPEED_DUPLEX	8 bits	Bit 0 = Auto-Negotiation Mode Available Bit 1 = Fixed Mode Available Note: If 0, ignore capabilities.

[VAR]	LENGTH	DESCRIPTION
CAP_FORCED_SPEED	8 bits	Bit 0 = Capable of Fixed 10Mbps Half Duplex Bit 1 = Capable of Fixed 10Mbps Full Duplex Bit 2 = Capable of Fixed 100Mbps Half Duplex Bit 3 = Capable of Fixed 100Mbps Full Duplex Bit 4 = Capable of Fixed 1000Mbps Half Duplex Bit 5 = Capable of Fixed 1000Mbps Full Duplex
CAP_ADV_SPEED	16 bits	Bit 0 = Valid (1) Bit 1 = 10Mbps Half Duplex Bit 2 = 10Mbps Full Duplex Bit 3 = 100Mbps Half Duplex Bit 4 = 100Mbps Full Duplex Bit 5 = 1000Mbps Half Duplex Bit 6 = 1000Mbps Full Duplex Bit 7 = 10Gbps Full Duplex Bit 8 = 2.5Gbps Full Duplex Bit 9 = 5Gbps Full Duplex
CAP_ADV_PAUSE	8 bits	Bit 0 = Not used Bit 1 = Advertise Symmetric Pause Bit 2 = Advertise Asymmetric Pause
CAP_MDI	8 bits	0 = Not Known 1 = Automatic 2 = Force MDI 4 = Force MDIX (Crossover)
CAP_MS	8 bits	Bit 0 = Ignore Bit 1 = Automatic, prefer MASTER Bit 2 = Automatic, prefer SLAVE Bit 3 = Force MASTER Bit 4 = Force SLAVE
Remote Transceiver:		
REMOTE_ADV_SPEED	16 bits	Bit 1 = Advertising 10Mbps Half Duplex Bit 2 = Advertising 10Mbps Full Duplex Bit 3 = Advertising 100Mbps Half Duplex Bit 4 = Advertising 100Mbps Full Duplex Bit 5 = Advertising 1000Mbps Half Duplex Bit 6 = Advertising 1000Mbps Full Duplex Bit 7 = Advertising 10Gbps Full Duplex Bit 8 = Advertising 2.5Gbps Full Duplex Bit 9 = Advertising 5Gbps Full Duplex
REMOTE_ADV_PAUSE	8 bits	Bit 0 = Not used Bit 1 = Advertising Symmetric Pause Bit 2 = Advertising Asymmetric Pause

8.2.4 apgGetPort [PORTID] TXSTATS [VAR]

The transmit counters are read with the **apgGetPort TXSTATS** command.

The **apgGetPort TXSTATS** must always be preceded by at least one **apgLoadUnit COUNTERS**, otherwise the command will display an error message and exit the TCL environment.

Eg:

```
# Instantiate APG TCL API (Windows)
source "C:/Program Files/Axtrinet/APG/axtrinetApi.tcl"

# Open Connection                                                         Open
set IPADDRESS 192.168.1.100                                               Connection
set UNITID [apgOpen $IPADDRESS]
set PORTLIST {{$UNITID 1 0} {$UNITID 2 0}}

# Port Counters                                                           Counters
puts "\nDEMO: Display Port Counters"
apgLoadUnit $UNITID COUNTERS                                              Load
foreach PORTID $PORTLIST {
  scan $PORTID "%d %d %d" UN PO SP
  set TXPKT  [apgGetPort $PORTID TXSTATS GOOD_PACKETS]                    Get counters
  set TXRATE [expr [apgGetPort $PORTID RATES TXBITRATE] / 1000000.0]
  set RXPKT  [apgGetPort $PORTID RXSTATS GOOD_PACKETS]
  if { $TXPKT > 0 } {
    puts [format "      Port %2d TX %7d at %8.2f Mbps, RX %7d packets" \
      $PO $TXPKT $TXRATE $RXPKT]
  }
}

# Close connection                                                        Close
apgClose $UNITID                                                          connection
```

The **apgGetPort TXSTATS** will change during a test, and should always be preceded by a **apgLoadUnit COUNTERS** command. If the counters are not re-loaded, **apgGetPort TXSTATS** will return the previous values.

[VAR]	LENGTH	DESCRIPTION
TIMESTAMP	32 bits	Time that the transmit counters are read
BYTES	64 bits	Number of transmitted bytes
GOOD_PACKETS	64 bits	Number of good packets transmitted
PKT_64	32 bits	Count of 64 byte packets transmitted
PKT_65_128	32 bits	Count of 65-128 byte packets transmitted
PKT_129_256	32 bits	Count of 129-256 byte packets transmitted
PKT_257_512	32 bits	Count of 257-512 byte packets transmitted
PKT_513_1024	32 bits	Count of 513-1024 byte packets transmitted
PKT_1025_1536	32 bits	Count of 1025-1536 byte packets transmitted
PKT_1537_9000	32 bits	Count of 1537-9000 byte packets transmitted
PKT_9001_MAX	32 bits	Count of >9001 byte packets transmitted
TX_RUNTIME	32 bits	Time in seconds since the transmit command was sent

8.2.5 apgGetPort [PORTID] RXSTATS [VAR]

The receive counters are read with the **apgGetPort RXSTATS** command.

The **apgGetPort RXSTATS** must always be preceded by at least one **apgLoadUnit COUNTERS**, otherwise the command will display an error message and exit the TCL environment.

The **apgGetPort RXSTATS** will change during a test, and should always be preceded by a **apgLoadUnit COUNTERS** command. If the counters are not re-loaded, **apgGetPort RXSTATS** will return the previous values.

[VAR]	LENGTH	DESCRIPTION
TIMESTAMP	32 bits	Time that the receive counters are read
BYTES	64 bits	Number of received bytes
GOOD_PACKETS	64 bits	Number of good packets received
PKT_UNDERSIZE	32 bits	Number of under-sized (<64 Bytes) packets received
PKT_FRAGMENTS	32 bits	Number of packets fragments received
PKT_FCSERROR	32 bits	Number of packets received with Frame Checksum (FCS) Errors
PKT_NOSFD	32 bits	Number of packets received without a valid Start-of-Frame Delimiter (SFD)
PKT_64	32 bits	Count of 64 byte packets received
PKT_65_128	32 bits	Count of 65-128 byte packets received
PKT_129_256	32 bits	Count of 129-256 byte packets received
PKT_257_512	32 bits	Count of 257-512 byte packets received
PKT_513_1024	32 bits	Count of 513-1024 byte packets received
PKT_1025_1536	32 bits	Count of 1025-1536 byte packets received
PKT_1537_9000	32 bits	Count of 1537-9000 byte packets received
PKT_9001_MAX	32 bits	Count of >9001 byte packets received

8.2.6 apgGetPort [PORTID] RATES [VAR]

The transmit and receive data rates are read with the **apgGetPort RATES** command.

apgGetPort RATES must always be preceded by at least one **apgLoadUnit RATES**, otherwise the command will display an error message and exit the TCL environment.

apgGetPort RATES will change during a test, and should always be preceded by a **apgLoadUnit RATES** command. If the rates are not re-loaded, **apgGetPort RATES** will return the previous values.

[VAR]	LENGTH	DESCRIPTION
TXPKTRATE	32 bits	Transmit Packet Rate (packets/sec)
TXBYTERATE	64 bits	Transmit Byte Rate (Bytes/sec) Derived from the Transmit BYTE count
TXBITRATE	64 bits	Transmit Bit Rate (Bits/sec) Bits on the wire, including Inter-Frame Gap (IFG) and Start-of-Frame (SFD) delimiter.
RXPKTRATE	32 bits	Receive Packet Rate (packets/sec)
RXERRORRATE	32 bits	Receive FCS Errored Packet Rate (packets/sec). Divide the returned value by 16384 to get the FCS errored packets per second.
RXBYTERATE	64 bits	Receive Byte Rate (Bytes/sec) Derived from the Receive BYTE count
RXBITRATE	64 bits	Receive Bit Rate (Bits/sec) Bits on the wire, including Inter-Frame Gap (IFG) and Start-of-Frame (SFD) delimiter.

8.2.7 apgGetPort [PORTID] PRBS [VAR] (APGV2 Only)

The PRBS configuration and status are read with the **apgGetPort PRBS** command.

apgGetPort PRBS must always be preceded by at least one **apgLoadUnit PRBSSTATUS**, otherwise the command will display an error message and exit the TCL environment.

apgGetPort PRBS will change during a test, and should always be preceded by a **apgLoadUnit PRBSSTATUS** command. If the status is not reloaded, **apgGetPort PRBS** will return the previous values.

[VAR]	LENGTH	DESCRIPTION
PRBSMODE	2 bits	Separate transmit and receive enables. Note that the port is in PRBS mode if either is set. Bit 0 = TXENABLE Bit 1 = RXENABLE Value: 0 = DISABLED; 1 = TXONLY; 2 = RXONLY; 3 = TXRX
PRBSPATTERN	3 bits	0 = PRBS7 1 = PRBS9 2 = PRBS15 3 = PRBS23 4 = PRBS31 Note that the same patterns **must** be set on the transmit and receive ports.
LOCKED	1 bit	PRBS Locked state, where: 1 = Locked; 0 = Not Locked
RUNTIME	32 bits	Runtime in ms since locked or cleared
ERRORS	32 bits	Bit errors since locked or cleared

[VAR]	LENGTH	DESCRIPTION
BER		Calculated Bit Error Rate from ERRORS / BITRATE, where BITRATE is the transmit rate of the encoded data on the interface. For the supported port speeds, the bitrate is: Port Speed — Bitrate 10Mbps — 10Mbps 100Mbps — 100Mbps 1Gbps — 1.25Gbps 2.5Gbps — 3.125Gbps 5Gbps — 5.15625Gbps 10Gbps — 10.3125Gbps

The PRBS status (locked, runtime, errors and BER) can be reset with the **apgControlPort PRBSCLEAR** command.

8.2.8 apgGetPort [PORTID] CAPTURE CONFIG [VAR]

Read the capture configuration, buffer and available memory sizes.

apgGetPort CAPTURE CONFIG must always be preceded by at least one **apgLoadPort CAPTURE CONFIG**, otherwise the command will display an error message and exit the TCL environment.

Eg:

```
# Instantiate APG TCL API (Windows)
source "C:/Program Files/Axtrinet/APG/axtrinetApi.tcl"
```

```
# Open Connection                                                                 Open
set IPADDRESS 192.168.1.100                                                       Connection
set UNITID [apgOpen $IPADDRESS]
set PORTLIST {{$UNITID 1 0} {$UNITID 2 0}}
```

```
# Capture Control                                                                 TX Control
puts "\nDEMO: Generating Packets"
apgControlPort STOPTX $PORTLIST
apgControlPort STARTTX $PORTLIST                                                  Start TX
puts "DEMO: Clearing Capture Buffers and enabling Capture on all ports "
apgControlPort PORTCAPTURE $PORTLIST                                              Enable
after 100                                                                         Capture
puts "DEMO: Stopping Traffic"
apgControlPort STOPTX $PORTLIST                                                   Stop TX
```

```
# Port Capture                                                                    Capture
puts "\nDEMO: Display Captured Packets"
foreach PORTID $PORTLIST {
  scan $PORTID "%d %d %d" UN PO SP
  apgLoadPort $PORTID CAPTURE CONFIG                                              Load config
  set AVAILABLE [apgGetPort $PORTID CAPTURE CONFIG AVAILABLE ]
  if { $AVAILABLE > 0 } {                                                         If data ...
    apgLoadPort $PORTID CAPTURE BUFFER                                            Load buffer
    puts "\n      Port $PO:\n"
    set PKTCOUNT [ apgGetPort $PORTID CAPTURE PACKET TOTALPACKETS ]
    for { set PKT 1 } { $PKT <= $PKTCOUNT } { incr PKT } {
      set LENGTH [apgGetPort $PORTID CAPTURE PACKET LENGTH $PKT]                  Display
      puts -nonewline [format "      %3d: %5d bytes -> " $PKT $LENGTH]            data
      set DATA [apgGetPort $PORTID CAPTURE PACKET DATA $PKT]
      foreach BYTE $DATA { puts -nonewline [format "%02X " $BYTE] }
      puts ""
    }
  }
}
```

```
# Close connection                                                                Close
apgClose $UNITID                                                                  connection
```

apgGetPort CAPTURE CONFIG will change while the capture buffer is filling, and should always be preceded by a **apgLoadPort CAPTURE CONFIG** command.

[VAR]	LENGTH	DESCRIPTION
AVAILABLE	32 bits	Filled capture data available for upload (bytes)
BUFFERSIZE	32 bits	Total memory available for packet capture (bytes)
FLAGS	2 bits	0 = Capture Disabled 1 = Local Port Buffer Enabled & Capturing Packets 2 = Capture Disabled 3 = Extended Packet Buffer Enabled & Capturing Packets ℹ The FLAGS are set indirectly using the **apgControlPort [CAPTURE]** commands

Packet capture is controlled with the **apgControlPort PORTCAPTURE**, **apgControlPort DISABLECAPTURE**, **apgControlPort CLEARCAPTURE** commands to start, stop and clear the capture buffer.

The capture buffer is loaded using **apgLoadPort CAPTURE BUFFER**, and read using **apgGetPort CAPTURE PACKET**.

8.2.9 apgGetPort [PORTID] CAPTURE PACKET TOTALPACKETS

TOTALPACKETS (32 bits) is the total number of packets held in the capture buffer.

The capture configuration is loaded with **apgLoadPort [PORTID] CAPTURE BUFFER**.

8.2.10 apgGetPort [PORTID] CAPTURE PACKET [VAR] [PKTNUM]

Read the capture packet data, where [PKTNUM] is the packet number in the capture buffer in the range 1 to [TOTALPACKETS].

The capture configuration is loaded with **apgLoadPort [PORTID] CAPTURE BUFFER**.

[VAR]	LENGTH	DESCRIPTION
RXTIME	64 bits	Receive timestamp of packet [PKTNUM] measured in 8ns intervals since the unit was turned on.
TSMODE	4 bits	Receive Timestamp Flag of packet [PKTNUM] 0xA = Approximated timestamp Anything else = Actual timestamp.
LENGTH	16 bits	Length of packet [PKTNUM]
PKTERROR	1 bit	Packet Error Flag of packet [PKTNUM]: 0 = OK 1 = FCS Error
DATA	[String]	Captured data of packet [PKTNUM] Packet format is space-separated integers

The following variables are also valid if the **TIMESTAMP_CONTROL** is enabled (see Section 9.2.4) the **transmitting** stream:

[VAR]	LENGTH	DESCRIPTION
IDFIELD	32 bits	ID Field extracted from packet [PKTNUM] Bytes [LENGTH]-19 to [LENGTH]-16
SEQNUM	32 bits	Sequence Number extracted from packet [PKTNUM] Bytes [LENGTH]-15 to [LENGTH]-12
TXTIME	64 bits	Transmit timestamp extracted from packet [PKTNUM] Bytes [LENGTH]-11 to [LENGTH]-4, measured in 8ns intervals since the unit was turned on.
DELTA	32 bits	Latency of packet [PKTNUM], calculated by [TSRX]–[TSTX] measured in 8ns ticks.

⚠ **WARNING: ID, SEQ and TXTIME field are extracted on every captured packet, regardless of the validity of the received packet. Care must be taken to ensure that the receive packets contain the expected data before further processing the information.**

Packet capture is enabled with **apgControlPort PORTCAPTURE** command.

The amount of data available for download can be monitored using the **apgLoadPort CAPTURE BUFFER** and **apgGetPort CAPTURE PACKET TOTALPACKETS** to get the total number of captured packets.

8.2.11 apgGetPort [PORTID] ANALYSIS [TYPE] [VAR]

The latency measurements are read with the **apgGetPort ANALYSIS** command.

apgGetPort ANALYSIS must always be preceded by at least one **apgLoadPort ANALYSIS**, otherwise the command will display an error message and exit the TCL environment.

⚠ **Capture Analysis only works across multiple units if a) all of the same version and b) timestamps are synchronised.**

If the measurements are not re-calculated, **apgGetPort ANALYSIS** will return the previous values.

[TYPE]	DESCRIPTION
TXTX	Transmit-to-transmit timestamp calculations
TXRX	Transmit-to-receive timestamp calculations
RXRX	Receive-to-receive timestamp calculations
SEQUENCE	Sequence Number status

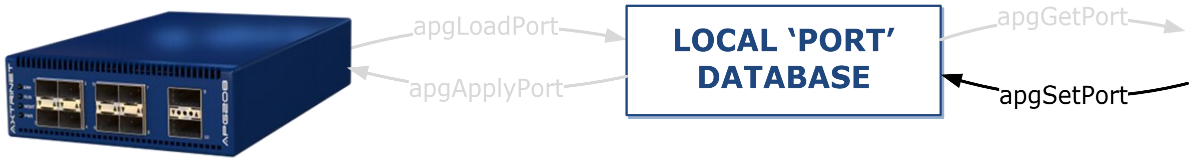
If **[TYPE]** is **TXRX, TXTX** or **RXRX**, the following **[VAR]** are available:

[VAR]	DESCRIPTION
CYCLES	Number of capture cycles to collect samples
SAMPLES	Number of processed latency samples
LIST	List of processed timing deltas
MIN	Minimum latency, measured in 8ns ticks
MAX	Maximum latency, measured in 8ns ticks
MEAN	Mean latency, measured in 8ns ticks
MODE	Mode latency, measured in 8ns ticks
STDDEV	Standard Deviation, measured in 8ns ticks

If **[TYPE]** is **SEQUENCE**, the following **[VAR]** are available:

[VAR]	DESCRIPTION
CYCLES	Number of capture cycles to collect samples
SAMPLES	Number of processed SEQ samples
LIST	List of processed SEQ samples
MIN	Minimum SEQNUM from sample
MAX	Maximum SEQNUM from sample
RANGE	SEQNUM range from sample
MONOTONIC	1=Monotonic 0=Not Monotonic
RPTCOUNT	Repeated Counter, increments if SEQ_n has been seen before
GAPCOUNT	Gap Counter, increments if $SEQ_n > SEQ_{n-1} + 1$
OOOCOUNT	Out-of-Order Counter, increments if $SEQ_n < SEQ_{max}$

8.3 Set Port Configuration - apgSetPort



apgSetPort returns the set value if successful, otherwise the command will display an error message and exit the TCL environment.

8.3.1 apgSetPort [PORTID] [VAR] [VAL]

The **apgSetPort** command allows configuration of the port and the transceiver inserted into the port.

8.3.1.1 Port State

Using the **apgSetPort** command to set the port state **must** be followed with **apgApplyPort STATE** to apply the port configuration onto the unit.

The [VAR] and [VAL] fields can be:

[VAR]	Description	Default
TXENABLE	0 = Module Disabled 1 = Module Enabled	0x1
RATE	[RATE SELECT 1][RATE SELECT 0] 0 = RS1 & RS0 Not Set (low speed operation) 1 = RS1 Not Set, RS0 Set 2 = RS1 Set, RS0 Not Set 3 = RS1 & RS0 Set (high speed operation) ⚠ **Check vendor's recommended settings in the datasheet. Both bits must be set correctly if using dual speed transceivers, otherwise the transceiver may behave unpredictably (eg link problems, packet CRC errors). Both bits should normally be set high for 10Gbps operation, although some vendor's transceivers do not meet the SFP MSA guidelines.**	0x3
TOPOLOGY	BLOCKED = Port Disabled **SFP Ports only:** 1G = Fixed 1Gbps **SFP+ Ports only:** 10G = Fixed 10Gbps **QSFP+ Ports only:** 40G = Fixed 40Gbps 4x10G = Fixed 4x10Gbps	SFP+ Ports: 10G QSFP+ Ports: 40G

[VAR]	Description	Default
LINKMODE	**SFP Ports only:** 1000X = 1000Base-X (1Gbps only) **SFP+ Ports only:** SGMII = SGMII (10Gbps or 1Gbps) **QSFP+ Ports only:** SRLR = 40GBase-SR/LR	SFP+ Ports: SGMII QSFP+ Ports: SRLR

8.3.1.2 Copper Module

Using the **apgSetPort** command to set the copper module configuration state **<u>must</u>** be followed with **apgApplyPort COPPERMODULE** to apply the copper module configuration onto the unit.

The [VAR] and [VAL] fields can be:

[VAR]	Description	Default
AUTONEG	**Auto-Negotiation/Fixed Mode** 0 = Ignore 1 = Auto-Negotiate Speed/Duplex 2 = Fixed Speed/Duplex	1 (auto)
FORCE_SPEED	**Force Speed/Duplex** Bit 0 = Force 10Mbps, Half Duplex Bit 1 = Force 10Mbps, Full Duplex Bit 2 = Force 100Mbps, Half Duplex Bit 3 = Force 100Mbps, Full Duplex Bit 4 = Force 1000Mbps, Half Duplex Bit 5 = Force 1000Mbps, Full Duplex Note: If 0, ignore settings.	0x00 (ignore)
ADV_SPEED	**Advertise Speed/Duplex** Bit 0 = Advertise Valid Bit 1 = Advertise 10Mbps, Half Duplex Bit 2 = Advertise 10Mbps, Full Duplex Bit 3 = Advertise 100Mbps, Half Duplex Bit 4 = Advertise 100Mbps, Full Duplex Bit 5 = Advertise 1000Mbps, Half Duplex Bit 6 = Advertise 1000Mbps, Full Duplex Bit 7 = Advertise 10Gbps, Full Duplex Bit 9 = Advertise 2G5Mbps, Full Duplex Bit 10= Advertise 5GMbps, Full Duplex Note: Advertise Valid bit must be set otherwise the advertised capabilities are ignored.	0x55 (1Gbps) [10/100/1000M] 0xC1 (10Gbps) [10G/1G]
ADV_PAUSE	**Advertise Pause Frames** Bit 0 = Not Used Bit 1 = Advertise Symmetrical Pause Bit 2 = Advertise Asymmetrical Pause	0 (disabled)
MDI	**MDI Crossover** 1 = Automatic 2 = MDI 4 = MDIX (Crossover)	1 (auto)
MS	**MASTER/SLAVE Mode: [1000Base-T only]** 1 = Automatic, prefer MASTER 2 = Automatic, prefer SLAVE 4 = Force MASTER 8 = Force SLAVE	1 (auto, master)

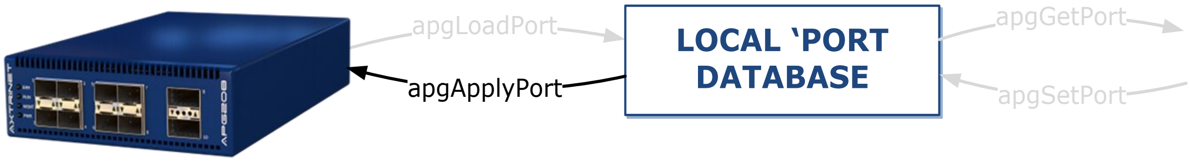
8.3.1.3 PRBS (APGV2 Only)

Using the **apgSetPort** command to set the PRBS state **must** be followed with **apgApplyPort PRBS** to apply the port PRBS configuration onto the unit.

The [VAR] and [VAL] fields can be:

[VAR]	Description	Default
PRBSMODE	Separate transmit and receive enables. Options: DISABLED, TXONLY, RXONLY, TXRX	DISABLED
PRBSPATTERN	Options: PRBS7, PRBS9, PRBS15, PRBS23, PRBS31	PRBS15

8.4 Apply Port Configuration - apgApplyPort



apgApplyPort returns 1 if successful, otherwise the command will display an error message and exit the TCL environment.

Changing the SFP+/SFP port topology between 10Gbps and 1Gbps modes will cause the unit to reset while a different FPGA image is loaded.

8.4.1 apgApplyPort [PORTID] STATE

The **apgApplyPort STATE** command must be used after **apgSetPort** command to apply the port state configuration changes to the unit.

⚠ **WARNING: On a APGV1 unit only, if changing SFP+/SFP port topology between 10Gbps and 1Gbps modes, apgApplyPort can take up to 80sec to respond while the unit reconfigures with the new port speed.**
A APGV2 unit can support different port speeds and does not need to reset.

8.4.2 apgApplyPort [PORTID] COPPERMODULE

The **apgApplyPort COPPERMODULE** command must be used after **apgSetPort** command to apply the copper module configuration changes to the unit.

8.4.3 apgApplyPort [PORTID] PRBS (APGV2 Only)

The **apgApplyPort PRBS** command must be used after **apgSetPort** command to apply the PRBS configuration changes to the unit.

8.5 Control Commands - apgControlPort

Packet generation and capture are controlled with the **apgControlPort** command.

apgControlPort returns 1 if successful, otherwise the command will display an error message and exit the TCL environment.

apgControlPort commands are performed on a group of ports identified by [PORTLIST], where [PORTLIST] is a TCL list of PORTIDs:

Eg:

```
apgControlPort STARTTX {{ 1 1 } { 1 3 }}
```

Starts transmission on ports 1 & 3

A port can be one of five transmit states:

STATE	Description
STOPPED	Traffic generation stopped. Stream configuration is loaded into the port traffic generator when a **STARTTX** or **STEPTX** is issued. Port transmit counters are stopped, although the port receive counters may still be counting if the port is receiving traffic.
ARMED	The port is armed and remains in this state until a **apgControlUnit SYNCGO** or **TSSYNCCLEAR** command is issued to synchronously start transmission on all armed ports; or a **SYNCCLEARUP** command is issued to disarm the port. See Section 3.7 for more information.
TRANSMITTING	The port is generating traffic, and remains in this state until a **STOPTX**, **PAUSETX** or **STEPTX** command is issued. If all of the enabled streams are generating fixed burst length traffic, the port will stop transmitting when all streams have transmitted the **BURST LENGTH** number of packets and move to the **STOPPED** state. A **STEPTX** command transmits the next packet, then PAUSES. Issuing a second **STARTTX** has no effect.
PAUSED	Traffic generation is stopped. Transmission continues when a **STARTTX** or **STEPTX** command is issued, without loading a new stream configuration. The PAUSE state is useful for interrupting a continuous stream to allow the test data to settle and read the counters.
STEP	Transmit the next packet from the stream queue. The port immediately returns to the PAUSED state.

Eg:

```
# Instantiate APG TCL API (Windows)
source "C:/Program Files/Axtrinet/APG/axtrinetApi.tcl"                       ← Load API

# Open Connection                                                            Open
set IPADDRESS 192.168.1.100                                                  Connection
set UNITID [apgOpen $IPADDRESS]
set PORTLIST {{$UNITID 1 0} {$UNITID 2 0}}

# Port Transmit & Capture Control                                            TX Control
puts "\nDEMO: Generating Packets for 3 seconds"
apgControlPort STOPTX $PORTLIST
apgControlPort CLEARCOUNTERS $PORTLIST                                       Clear counts
apgControlPort STARTTX $PORTLIST                                             Start TX
after 1000
puts "DEMO: Calculating data rates over 1000ms"                              Calc Rates
apgLoadUnit $UNITID RATES 1000
puts "DEMO: Clearing Capture Buffers and enabling Capture on all ports "
apgControlPort PORTCAPTURE $PORTLIST                                         Enable
after 1000                                                                   Capture
puts "DEMO: Stopping Traffic"
apgControlPort STOPTX $PORTLIST                                              Stop TX

# Close connection                                                           Close
apgClose $UNITID                                                             connection
```

8.5.1 apgControlPort [COMMAND] [PORTLIST]

Traffic generator control is performed on a group of ports identified by [PORTLIST].

[PORTLIST] is a TCL list of PORTIDs:

Eg: `apgControlPort STARTTX {{ 1 1 } { 1 3 }}` Starts transmission on ports 1 & 3

The following control commands can be used to control the port traffic generator:

[COMMAND]	Description
STARTTX	Start packet transmission on all [PORTLIST] ports. From the STOPPED state, **STARTTX** loads the new stream configuration on all [PORTLIST] ports and start transmitting. From the PAUSED state, packet transmission will start without loading the latest configuration.
STEPTX	Transmit the next packet on all [PORTLIST] ports. From the STOPPED state, **STEPTX** loads the new stream configuration on all [PORTLIST] ports and transmits a single packet. Where multiple streams are enabled, the port will transmit the next packet from the stream sequencer.
PAUSETX	Pause the packet transmission on all [PORTLIST] ports
STOPTX	Stop the packet transmission on all [PORTLIST] ports
CLEARCOUNTERS	Clear the transmit and receive packet counters on all [PORTLIST] ports
ZEROPORTSEQ	Reset the port sequence number on all [PORTLIST] ports. The first packet transmitted from the port will have a sequence number 0x00000001.

8.5.2 apgControlPort [SYNC COMMAND] [PORTLIST] (APGV2 Only)

Arming ports for synchronous transmission is performed on a group of ports identified by [PORTLIST].

The following control command is used to arm the ports for synchronous transmission:

[COMMAND]	Description
STARTSYNCTX	Arm synchronous packet transmission on all [PORTLIST] ports. From the STOPPED state, **STARTSYNCTX** loads the new stream configuration on all [PORTLIST] ports but does not start transmitting. Synchronous transmission is started on the rising edge of a pulse on the SYNC interface, either from a connected unit with the **apgControlUnit SYNCGO** command; or from an external synchronisation source. The armed state can be cleared with the **apgControlUnit SYNCCLEARUP** command. ⚠ **A port will stop responding to port control commands until the synchronous transmit event (SYNCGO or external) occurs.** **During script development, this may leave the port in a state where the script aborts with a "Unit Busy" error message.** **During script development, it is recommended that an apgControlUnit SYNCCLEARUP is used at the start of the script to clear any armed states.**

See Section 3.7.3 and the APG User Guide [1] for more information.

8.5.3 apgControlPort [PRBS COMMAND] [PORTLIST] (APGV2 Only)

The following control command is used to control the PRBS ports:

[COMMAND]	Description
PRBSCLEAR	Clear the RUNTIME and ERRORS on all [PORTLIST] ports.

8.5.4 apgControlPort [CAPTURE] [PORTLIST]

Capture control is performed on a group of ports identified by [PORTLIST].

The following control commands can be used to control the port packet capture:

[CAPTURE]	Description
PORTCAPTURE	Enable the local capture buffers on all [PORTLIST] ports and start capturing packets. Enabling the local capture from a deep capture will clear the deep capture buffer.
DEEPCAPTURE	Enable the extended capture buffer on all [PORTLIST] ports and start capturing packets. Enabling the deep capture from a local capture will clear the local capture buffer.
DISABLECAPTURE	Disable packet capture on all [PORTLIST] ports. The capture buffer is not cleared. The selected capture buffer is still enabled, and can be read using **apgLoadPort CAPTURE BUFFER**. Capture can be restarted with **apgControlPort PORTCAPTURE** (local) or **apgControlPort DEEPCAPTURE** (deep). If the capture buffer has not been cleared, the buffer will contain the already-captured data after restarting.
CLEARCAPTURE	Clear the capture buffers on all [PORTLIST] ports

For example, to capture and display active traffic on connected ports 1 & 5:

Eg:

```
set PORTLIST {{1 2}{1 5}}
apgControlPort PORTCAPTURE $PORTLIST
```

Enable local port capture on both ports.

```
apgControlPort CLEARCAPTURE $PORTLIST
apgControlPort STARTTX $PORTLIST
```

Clear capture buffers and start transmitting

```
foreach PORTID $PORTLIST {apgLoadPort $PORTID CAPTURE BUFFER 100}
```

Load 100 packets from each port buffer

```
foreach SUBPORTID $PORTLIST {
    set PKTCOUNT [ apgGetPort $SUBPORTID CAPTURE PACKET
    TOTALPACKETS ]
```

For each port, get the number of downloaded packets

```
    for { set PKT 1 } { $PKT <= $PKTCOUNT } { incr PKT } {
        puts [apgGetPort $SUBPORTID CAPTURE PACKET DATA
    $PKT]
```

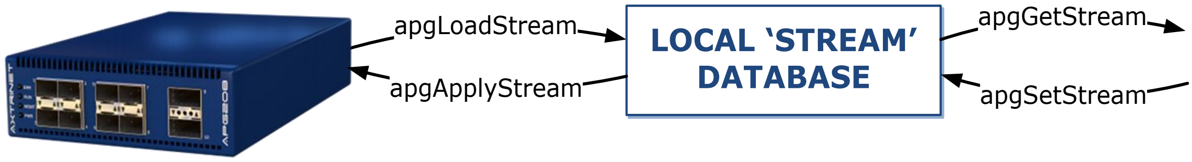
Then display each packet.

```
    }
}
```

8.5.5 apgControlPort CLEARANALYSIS [PORTLIST]

Clear the Packet Analysis measurements on a group of ports identified by [PORTLIST].

9. STREAM COMMANDS



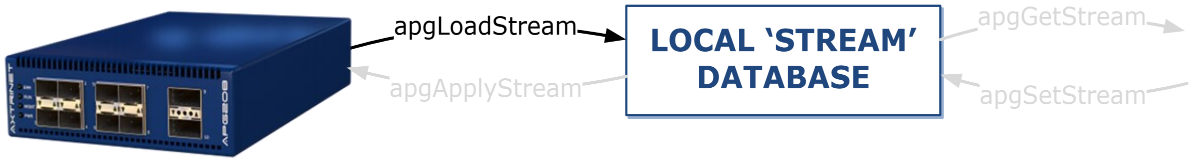
apgLoadStream	**apgLoadStream [STREAMID]**
apgGetStream	**apgGetStream [STREAMID] CONFIG [VAR]** **apgGetStream [STREAMID] HEADER HEADER_LIST** **apgGetStream [STREAMID] HEADER [HDR] [FLD]** **apgGetStream [STREAMID] PAYLOAD [VAR]**
apgSetStream	**apgSetStream [STREAMID] DEFAULT** **apgSetStream [STREAMID] CONFIG [VAR] [VAL]** **apgSetStream [STREAMID] HEADER HEADER_LIST [HDRLIST]** **apgSetStream [STREAMID] HEADER [HDR] [FLD] [VAL]** **apgSetStream [STREAMID] PAYLOAD [VAR] [VAL]**
apgApplyStream	**apgApplyStream [STREAMID]**

Where STREAMID = {UNIT.PORT.STREAM} or {UNIT.PORT.SUBPORT.STREAM}

Eg:

```
# Instantiate APG TCL API (Windows)
source "C:/Program Files/Axtrinet/APG/axtrinetApi.tcl"                      ← Load API

# Open Connection                                                           Open
set IPADDRESS 192.168.1.100                                                 Connection
set UNITID [apgOpen $IPADDRESS]
set PORTLIST {{$UNITID 1 0} {$UNITID 2 0}}

# Configure Streams                                                         Configure
puts "\nDEMO: Configure Streams"                                            Streams
set STREAM 0
foreach PORTID $PORTLIST {
  scan $PORTID "%d %d %d" UN PO SP
  puts "      -> Port $PO"
  for { set ST 0 } { $ST < 8 } { incr ST } {
    set STREAMID [linsert $PORTID end $ST]
    if { $ST == 0 } {                                                       Stream 0
      apgSetStream $STREAMID DEFAULT
      apgSetStream $STREAMID CONFIG ENABLE 1                                Enable
      apgSetStream $STREAMID CONFIG SIZE_MODE RANDOM                        Random Size
      apgSetStream $STREAMID CONFIG PACKET_SIZE 64                          64 -
      apgSetStream $STREAMID CONFIG PACKET_SIZE_MAX 5000                     5000 bytes
      apgSetStream $STREAMID HEADER HEADER_LIST "MACHEADER"                 MACHEADER
      apgSetStream $STREAMID HEADER MACHEADER DA \
        [format "08:%02d:%02d:%02d:%02d:00" $UN $PO $SP $ST]
      apgSetStream $STREAMID HEADER MACHEADER SA \
        [format "08:%02d:%02d:%02d:%02d:01" $UN $PO $SP $ST]
      apgSetStream $STREAMID PAYLOAD DATA_CONTROL RANDOM                    Random Data
      if { $PO >= 9 } {                                                     If 40G
        apgSetStream $STREAMID PAYLOAD DATA_CONTROL INCREMENT                Increment
        apgSetStream $STREAMID PAYLOAD PAYLOAD_DATA "01 02 03 04 05 06 07 08"  Data
      }
    } else {
      apgSetStream $STREAMID CONFIG ENABLE 0                                Disable 1-7
    }
    apgApplyStream $STREAMID                                                Apply Config
  }
}

# Close connection                                                          Close
apgClose $UNITID                                                            connection
```

9.1 Load Stream Configuration - apgLoadStream



Load the STREAM configuration and status from the hardware into the local database.

apgLoadStream returns 1 if successful, otherwise the command will display an error message and exit the TCL environment.

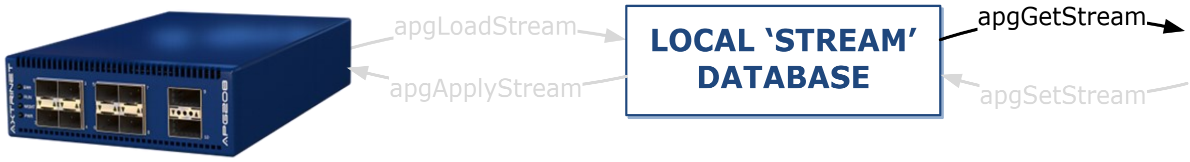
9.1.1 apgLoadStream [STREAMID]

Loads the Stream Information into the local configuration database, containing:

- Stream Configuration
  - Enable
  - Transmit, Rate and Payload Modes
  - Frame Size
- Stream Header
  - Configured Headers
  - Header Settings
- Stream Payload
  - Data, Offset and Length
  - Timestamp

The Stream Information can be read using the **apgGetStream CONFIG**, **HEADER** and **PAYLOAD** commands.

9.2 Get Stream Configuration - apgGetStream



apgGetStream returns the get value if successful, otherwise the command will display an error message and exit the TCL environment.

A stream configuration is defined by the CONFIG, HEADER and PAYLOAD:

CONFIG	Defines the stream state, packet rate and packet size
HEADER	Defines the Ethernet headers (eg MACHEADER) and fields (eg DA)
PAYLOAD	Defines the packet contents following the headers

9.2.1 apgGetStream [STREAMID] CONFIG [VAR]

The stream configuration status, rate and size information is read with the **apgGetStream CONFIG** command.

apgGetStream CONFIG must either be configured using **apgSetStream CONFIG** or be preceded by at least one **apgLoadStream**. If [VAR] has not been defined, **apgGetStream CONFIG** will display an error message and exit the TCL environment.

[VAR]	LENGTH	Description
ENABLE	8 bits	0 = Stream Disabled 1 = Stream Enabled
TX_MODE	8 bits	0 = Continuous 1 = Burst
RATE_MODE	8 bits	0 = TXGAP (Inter Frame Gap) 1 = PPS (Frames Per Second) 2 = Percent
SIZE_MODE	8 bits	0 = Fixed 3 = Random
TX_CONTROL	8 bits	0 = Transmit good frames
RATE_VALUE	64 bits	If RATE_MODE = 0 (TXGAP) → TXGAP = RATE_VALUE * 8 bytes * link speed
		If RATE_MODE = 1 (Packets per Second → PPS = RATE_VALUE pps
		If RATE_MODE = 2 (PERCENT) → PERCENT = RATE_VALUE * 1000 Eg RATE_VALUE = 90500 → 90.5%
TX_BURST_SIZE	32 bits	Number of frames in the burst
TX_BURST_COUNT	32 bits	Number of bursts to transmit
TX_IBG	32 bits	Inter-burst gap (TX_IBG x 5.5ns)
PACKET_SIZE	32 bits	Total frame length (including FCS) in bytes. If SIZE_MODE is not 0 (FIXED), FRAME_SIZE defines the minimum frame length for the increment, decrement or random size.
PACKET_SIZE_MAX	32 bits	Maximum frame length (including FCS) in bytes for the increment, decrement or random size modes. If SIZE_MODE is 0 (FIXED) this value is ignored.

9.2.2 apgGetStream [STREAMID] HEADER HEADER_LIST

Returns a list of configured header names in the order that the headers appear in the frame.

apgGetStream HEADER HEADER_LIST must either be configured using **apgSetStream HEADER HEADER_LIST** or be preceded by at least one **apgLoadStream**. If [VAR] has not been defined, **apgGetStream HEADER HEADER_LIST** will display an error message and exit the TCL environment.

For example, the minimum set of ethernet headers is:

MACHEADER ETHERNET_II USERDEFINED

The full list of headers is defined in the latest Header Definition document [2].

For example: A header structure containing a MAC Header [MACHEADER] and an Ethertype [ETHERNET_II] will return "MACHEADER ETHERNET_II"

The maximum number of headers in the header list is 10.

The maximum length of the headers is 64 bytes.

9.2.3 apgGetStream [STREAMID] HEADER [HDR] [FLD]

Returns the value of a header field configuration.

apgGetStream HEADER must either be configured using **apgSetStream HEADER** or be preceded by at least one **apgLoadStream**. If [FLD] has not been defined, **apgGetStream HEADER** will display an error message and exit the TCL environment.

The returned value depends on the configuration mode (FIX, INC etc).

apgGetStream HEADER returns a single value if [MODE] is FIXed.

apgGetStream HEADER returns a list if [MODE] is not FIXed, containing the following fields:

"[VAL] [MODE] [MIN] [STEP] [MAX]"

Where:

Variable	Description	
[VAL]	Initial Value	
[MODE]	The header field modes are defined in the Header Definition document [2]. For example, the MACHEADER DA modes are "FIX INC" where:	
	FIX	Fixed [FLD] value [VAL]
	INC	Incrementing [FLD] value [VAL], starting at [VAL] with each byte incrementing to [MAX], wrapping to [MIN] to continue incrementing
[MIN]	Minimum range for INC [FLD] value wraps to [MAX] in decrementing mode.	
[STEP]	Byte-wise step	
[MAX]	Maximum range for INC . [FLD] value wraps to [MIN] (zero) in incrementing mode.	

The full list of [HDR] headers is defined in the latest Header Definition document [2].

The header fields [FLD] for the basic set of ethernet headers are shown below as examples. The full list of headers is defined in the latest Header Definition document [2].

apgGetStream HEADER returns the variable value in the following formats:

Format	Minimum	Maximum	Comments
MAC-ADDRESS	00:00:00:00:00:00	FF:FF:FF:FF:FF:FF	
HEX4	0x0	0xFFFF	
HEX2ARRAY	0x0	0xFF	Variable length HEX2 values separated by spaces

9.2.3.1 MACHEADER

[FLD]	DESC	SIZE	FORMAT	MODE
DA	MAC Destination Address	6 bytes	MAC-ADDRESS	FIX INC
SA	MAC Source Address	6 bytes	MAC-ADDRESS	FIX INC

9.2.3.2 ETHERNET_II

[FLD]	DESC	SIZE	FORMAT	MODE
ETHERTYPE	Ethernet-II Type	2 bytes	HEX4	FIX INC

9.2.3.3 USERDEFINED

The USERDEFINED header is a variable length configurable field that can be used to implement any fixed custom header field.

[FLD]	DESC	SIZE	FORMAT	MODE
DATA	Data Values	Variable	HEX2ARRAY	--

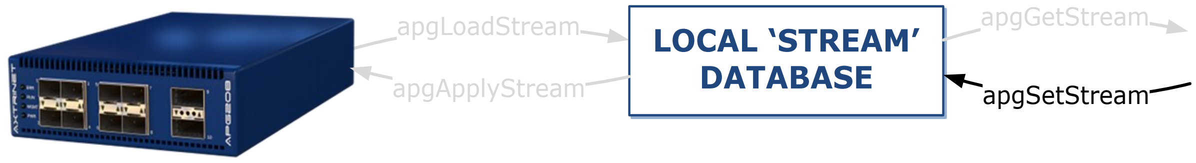
9.2.4 apgGetStream [STREAMID] PAYLOAD [VAR]

Returns the value of the payload fields.

apgGetStream PAYLOAD must either be configured using **apgSetStream PAYLOAD** or be preceded by at least one **apgLoadStream**. If [VAR] has not been defined, **apgGetStream PAYLOAD** will display an error message and exit the TCL environment.

[VAR]	LENGTH	Description
PAYLOAD_DATA	12 Char	Payload values
DATA_CONTROL	8 bits	FIXED, RANDOM, INCREMENT, DECREMENT

9.3 Set Stream Configuration - apgSetStream



apgSetStream returns the set value if successful, otherwise the command will display an error message and exit the TCL environment.

A stream configuration is made up of three parts:

- **apgSetStream CONFIG**
- **apgSetStream HEADER**
- **apgSetStream PAYLOAD**

All three parts have to be correctly defined to successfully configure a stream.

It is recommended that a default stream is created using **apgSetStream DEFAULT** to ensure that all of the variables correctly are defined, then modifying the default configuration.

The stream configuration is applied to the unit using **apgApplyStream**.

Eg:

```
# Instantiate APG TCL API (Windows)
source "C:/Program Files/Axtrinet/APG/axtrinetApi.tcl"                          ← Load API

# Open Connection                                                               Open
set IPADDRESS 192.168.1.100                                                     Connection
set UNITID [apgOpen $IPADDRESS]
set PORTLIST {{$UNITID 1 0} {$UNITID 2 0}}

# Configure Streams                                                             Configure
puts "\nDEMO: Configure Streams"                                                Streams
set STREAM 0
foreach PORTID $PORTLIST {
  scan $PORTID "%d %d %d" UN PO SP
  puts "      -> Port $PO"
  for { set ST 0 } { $ST < 8 } { incr ST } {
    set STREAMID [linsert $PORTID end $ST]
    if { $ST == 0 } {                                                           Stream 0
      apgSetStream $STREAMID DEFAULT
      apgSetStream $STREAMID CONFIG ENABLE 1                                    Enable
      apgSetStream $STREAMID CONFIG SIZE_MODE RANDOM                            Random Size
      apgSetStream $STREAMID CONFIG PACKET_SIZE 64                              64 -
      apgSetStream $STREAMID CONFIG PACKET_SIZE_MAX 5000                         5000 bytes
      apgSetStream $STREAMID HEADER HEADER_LIST "MACHEADER"                     MACHEADER
      apgSetStream $STREAMID HEADER MACHEADER DA \
        [format "08:%02d:%02d:%02d:%02d:00" $UN $PO $SP $ST]
      apgSetStream $STREAMID HEADER MACHEADER SA \
        [format "08:%02d:%02d:%02d:%02d:01" $UN $PO $SP $ST]
      apgSetStream $STREAMID PAYLOAD DATA_CONTROL RANDOM                        Random Data
      if { $PO >= 9 } {                                                         If 40G
        apgSetStream $STREAMID PAYLOAD DATA_CONTROL INCREMENT                    Increment
        apgSetStream $STREAMID PAYLOAD PAYLOAD_DATA "01 02 03 04 05 06 07 08"     Data
      }
    } else {
      apgSetStream $STREAMID CONFIG ENABLE 0                                    Disable 1-7
    }
    apgApplyStream $STREAMID                                                    Apply Config
  }
}

# Close connection                                                              Close
apgClose $UNITID                                                                connection
```

9.3.1 apgSetStream [STREAMID] DEFAULT

Sets the STREAMID configuration to default settings, with the following exceptions:

- STREAM 0 is ENABLED
- MAC DA is set to 08:$PORTID:$SUBPORTID:$STREAMID:00:00
- MAC SA is set to 08:$PORTID:$SUBPORTID:$STREAMID:00:01

9.3.2 apgSetStream [STREAMID] CONFIG [VAR] [VAL]

The **apgSetStream CONFIG** command allows configuration of the stream state, transmit mode, rate mode and packet size settings.

The [VAR] and [VAL] fields can be:

[VAR]	Description	Default	FORMAT
ENABLE	0 = Stream Disabled 1 = Stream Enabled	0	HEX2
TX_MODE	CONTINUOUS (0) BURST (1) MULTIBURST (2)	CONTINUOUS	
TX_BURST_SIZE	Number of frames in the burst	1	INT
TX_BURST_COUNT	Number of bursts to transmit	1	INT
TX_IBG	Inter-burst gap (x5.5ns)	0	INT
RATE_MODE	TXGAP Pseudo-IPG at internal clock frequency PPS Packets per Second PERCENT % maximum bit rate	TXGAP	
RATE_VALUE	If RATE_MODE = TXGAP → TXGAP = RATE_VALUE * 8 bytes * link speed	0	INT
	If RTE_MODE = PPS → PPS = RATE_VALUE pps		INT
	If RATE_MODE = PERCENT → PERCENT = RATE_VALUE * 1000 Eg RATE_VALUE = 90500 → 90.5%		INT
SIZE_MODE	FIXED = Fixed **PACKET_SIZE** byte packets RANDOM = Random length packets between **PACKET_SIZE** and **PACKET_SIZE_MAX** bytes. If **PACKET_SIZE_MAX** is not defined, fixed length **PACKET_SIZE** byte packets will be generated.	FIXED	

[VAR]	Description	Default	FORMAT
PACKET_SIZE	Total packet length (including FCS) in bytes. If SIZE_MODE is not 0 (FIXED), FRAME_SIZE defined the minimum frame length for the increment, decrement or random size. 32 ≤ **PACKET_SIZE** ≤ 16000 bytes If PACKET_SIZE < 32 → 32 If PACKET_SIZE > 16000 → 16000 Packet sizes < 64 bytes will be counted as UNDERSIZE packets. ⚠ **At 10Mbps, the packet size must be less than 1600 bytes**	100	INT
PACKET_SIZE_MAX	Maximum packet length (including FCS) in bytes. If SIZE_MODE is 0 (FIXED) this value is ignored. **PACKET_SIZE_MAX** ≤ 16000 bytes Larger values will be set to 16000.	100	INT

Where variable FORMAT has the follow settings:

FORMAT	Min	Max	
INT	0	N/A	
HEX2	0x0	0xFF	(256 dec)
HEX4	0x0	0xFFFF	(65,535 dec)
HEX8	0x0	0xFFFFFFFF	(4,294,967,296 dec)

9.3.3 apgSetStream [STREAMID] HEADER HEADER_LIST [HDRLIST]

Sets the configured headers in the order that the headers appear in the frame.

For example: A header structure containing a MAC Header [MACHEADER] and an Ethertype [Ethernet_II] is defined with [HDRLIST] set to "MACHEADER ETHERNET_II"

The full list of headers is defined in the latest Header Definition document [2].

The maximum number of headers in the header list is 10.

The maximum length of the headers is 64 bytes.

9.3.4 apgSetStream [STREAMID] HEADER [HDR] [FLD] [VAL]

Sets the value of a FIXED header field, where:

Variable	Description
[HDR]	The full list of [HDR] headers is defined in the latest Header Definition document [2].
[FLD]	The full list of headers fields [FLD] are defined in the latest Header Definition document [2]
[VAL]	Value to be set

apgSetStream variables are set in the following formats:

Format	Minimum	Maximum	Comments
MAC-ADDRESS	00:00:00:00:00:00	FF:FF:FF:FF:FF:FF	
INT	0		Depends on FLD 'size'
HEX2	0x0	0xFF	
HEX4	0x0	0xFFFF	
HEX2ARRAY	0x0	0xFF	Variable length HEX2 values separated by spaces

The maximum total header length is 64 bytes.

9.3.4.1 MACHEADER

[FLD]	DESC	SIZE	FORMAT
DA	MAC Destination Address	6 bytes	MAC-ADDRESS
SA	MAC Source Address	6 bytes	MAC-ADDRESS

Eg:

```
apgSetStream {1 1 0 1} HEADER MACHEADER DA 01:23:45:67:89:AB
apgApplyStream {1 1 0 1}
```

9.3.4.2 ETHERNET_II

[FLD]	DESC	SIZE	FORMAT
ETHERTYPE	Ethernet-II Type	2 bytes	HEX4

Eg:

```
apgSetStream {1 3 0 5} HEADER ETHERNET_II ETHERTYPE 0x0800
apgApplyStream {1 3 0 5}
```

9.3.4.3 USERDEFINED

The USERDEFINED header is a variable length configurable field that can be used to implement any fixed custom header field.

[FLD]	DESC	SIZE	FORMAT
DATA	Data Values	Variable	HEX2ARRAY

Eg:

```
apgSetStream {2 2 0 3} HEADER USERDEFINED "00 11 22 33 44 55"
apgApplyStream {2 2 0 3}
```

9.3.5 apgSetStream [STREAMID] HEADER [HDR] [FLD] [VAL] [MODE] [STEP][MIN][MAX]

Sets the value of a VARIABLE header field, where:

Variable	Description	
[HDR]	The full list of [HDR] headers is defined in the latest Header Definition document [2].	
[FLD]	The header fields [FLD] are defined in the Header Definition document [2]	
[VAL]	Value to be set	
[MODE]	The header field modes are defined in the Header Definition document [2]. For example, the MACHEADER DA modes are:	
	FIX	Fixed [FLD] value [VAL]
	INC	Incrementing [FLD] value [VAL], starting at [VAL] with each byte incrementing to [MAX], wrapping to [MIN] to continue incrementing
[STEP]	Byte-wise step	
[MIN]	Minimum range for INC ⚠ **Always zero in APG TCL API APG V1.4.1 Software.** [FLD] value wraps to [MAX] in decrementing mode.	
[MAX]	Maximum range for INC . [FLD] value wraps to [MIN] (zero) in incrementing mode.	

9.3.6 apgSetStream [STREAMID] PAYLOAD [VAR] [VAL]

Sets the value of the payload fields.

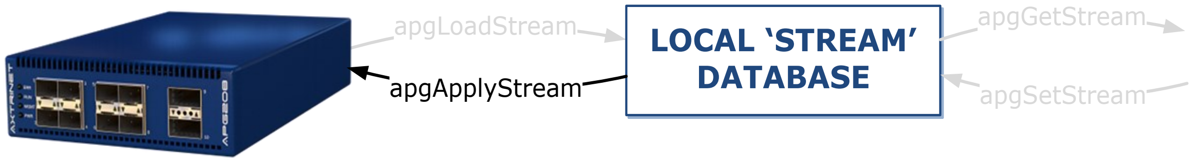
apgGetStream PAYLOAD must either be configured using **apgSetStream PAYLOAD** or be preceded by at least one **apgLoadStream**. If [VAR] has not been defined, **apgGetStream PAYLOAD** will display an error message and exit the TCL environment.

[VAR]	Description	Default	RANGE
DATA_CONTROL	FIXED, RANDOM, INCREMENT, DECREMENT	FIXED	
PAYLOAD_DATA	Payload values	00	HEX2ARRAY
TS_ENABLE	Timestamp Control 0 = Disabled 1 = Enabled	0	INT

Where variable FORMAT has the follow settings:

FORMAT	Min	Max	
HEX2ARRAY	0x0	0xFF	12-byte HEX2 values separated by spaces

9.4 Apply Stream Configuration - apgApplyStream



apgApplyStream returns 1 if successful, otherwise the command will display an error message and exit the TCL environment.

9.4.1 apgApplyStream [STREAMID]

Applies the STREAM CONFIG, HEADER and PAYLOAD configurations to the unit.

10. Tools

The following tools are provided to simplify access to the local data structures.

10.1 Command Tools

10.1.1 apgGetVariables [COMMAND] {FUNCTION}

Returns a space-separated string of valid variables for a command, where:

COMMAND	FUNCTION
VERSION	--
UNIT	INFO STATUS CLOCKSTATUS IPCONFIG
PORT	INFO STATUS TXSTATS RXSTATS RATES SETPORTCONFIG COPPERMODULE
STREAM	CONFIG PAYLOAD

For example, to get the apgGetUnit STATUS variables:

Eg:

```
apgGetVariables UNIT STATUS
```

→ **"UPTIME TEMP FAN SELFTEST READY"**

apgGetVariables can be used to load variables from the local database:

Eg:

```
foreach VAR [apgGetVariables UNIT INFO] {
    set $VAR [apgGetUnit $UNITID INFO $VAR]
}
puts $SERIAL                                  → APG000006
puts $PRODUCT                                 → APG208
```

10.2 Header Tools

10.2.1 apgGetHeaderList [STREAMID]

Returns a space-separated string of headers configured for stream STREAMID.

Eg:

```
apgGetHeaderList { 1 1 0 1 }
```

→ "MACHEADER VLAN_INNER ETHERNET_II"

10.2.2 apgGetHeaderFieldList [HDR]

Returns a space-separated string of fields for the headers configured for stream STREAMID.

Eg:

```
set STREAMID { 1 1 0 2 }
apgLoadStream $STREAMID

set HEADERLIST [apgGetHeaderList $STREAMID]            → "MACHEADER ETHERNET_II"

foreach HDR $HEADERLIST {
  puts "$HDR:[apgGetHeaderFieldList $HDR]"             → MACHEADER: DA SA
                                                       → ETHERNET_II: ETHERTYPE
}
```

10.2.3 apgGetHeaderFieldValue [HDR] [FLD] [VAR]

Returns the value of the header structure field variable, where VAR:

VAR	
LABEL	Header field name
DESCRIPTION	Field description
LENGTH	Length (integer)
SCALE	Scale (eg bits, bytes)
FORMAT	See **apgSetStream** (Section 9.3)
EDITABLE	0 = Fixed (value cannot be changed) 1 = Editable

Eg:

```
set STREAMID { 1 1 0 0 }
apgLoadStream $STREAMID
set HEADERLIST [apgGetHeaderList $STREAMID]

foreach HDR $HEADERLIST {

    set FIELDLIST [apgGetHeaderFieldList $HDR]

    foreach FLD $FIELDLIST {

        foreach VAR { DESCRIPTION LENGTH SCALE FORMAT } {
            set $VAR [apgGetHeaderFieldValue $HDR $FLD $VAR]
        }

        Puts "$HDR $FLD \($DESCRIPTION\)"
        puts " -> $LENGTH $SCALE \($FORMAT\)"

        }
    }
}
```

Displays:

```
MACHEADER DA (MAC Destination Address)
-> 6 bytes (mac-address)
MACHEADER SA (MAC Source Address)
-> 6 bytes (mac-address)

ETHERNET_II ETHERTYPE (The encapsulated protocol type >= 0x600)
-> 2 bytes (hex4)
```

10.3 HARDWARE VERSION TOOLS

10.3.1 apgGetHwVersion [UNITID]

Decodes the string returned from apgGetUnit INFO HW_VERSION to return 'V1' or 'V2'

10.4 TIMESTAMP TOOLS

10.4.1 apgConvertTicksToTime [UNITID] [TICKS] {UNITS}

Converts clock ticks (used by the transmit and receive timestamps) into time.

It uses apgGetHwVersion to determine the Axtrinet unit version (APGV1 or APGV2) and the internal clock period (8ns or 2.5ns).

UNITS	
PS	Picoseconds
NS	Nanoseconds (default)
US	Microseconds
MS	Milliseconds
SEC	Seconds
MIN	Minutes

APPENDIX A - Quick Reference Guide

TCL API VERSION	[VAR]
apgGetApiVersion [VAR]	**COMPANY** **BRAND** **DESCRIPTION** **VERSION** **BUILD_DATE** **API_VERSION**
CONNECTION	**[VAR]**
apgOpen [IP-ADDRESS]	
apgUSBOpen [SERIAL]	
➔ apgClose [UNITID]	
CONFIGURATION	**[VAR]**
apgSaveConfiguration [UNITID] [PORTLIST] {FILENAME}	Eg **PORTLIST = "{{1 2}{1 4}}" or ALL**
apgApplyConfiguration [UNITID] [FILENAME]	
UNIT COMMANDS	**[VAR]**
apgLoadUnit [UNITID] INFO	
➔ apgGetUnit [UNITID] INFO [VAR]	**API_VERSION** **PORT_COUNT** **API_MINIMUM** **SERIAL** **PRODUCT** **HWARE_VERSION** **FPGA_VERSION** **FPGA_BUILDDATE** **FIRMWARE_VERSION** **FIRMWARE_BUILDDATE**
apgLoadUnit [UNITID] IPCONFIG	
➔ apgGetUnit [UNITID] IPCONFIG [VAR]	**IPADDRESS** **IPMASK** **IPGATEWAY**
apgLoadUnit [UNITID] STATUS	
➔ apgGetUnit [UNITID] STATUS [VAR]	**UPTIME** **TEMP** **FAN** **SELFTEST** **READY**
apgLoadUnit [UNITID] CLOCKSTATUS (APGV2 Only)	
➔ apgGetUnit [UNITID] CLOCKSTATUS [VAR] (APGV2 Only)	**CLOCKMODE** **PLLSTATE**
apgLoadUnit [UNITID] PORTSTATUS **apgLoadUnit [UNITID] COUNTERS** **apgLoadUnit [UNITID] RATES**	See Port Commands
apgLoadUnit [UNITID] PRBSSTATUS	
➔ apgGetUnit [UNITID] PRBSPORTLIST	
➔ apgGetPort [PORTID] PRBS	See Port Commands
apgSetUnit [UNITID]	**CLKINMODE** **CLKOUTMODE** **TSRESETMODE**
➔ apgApplyUnit [UNITID]	**CLOCKMODE**
apgControlUnit [COMMAND] [UNITID]	**TSCLEAR**
apgControlUnit [SYNC COMMAND] [UNITID]	**SYNCGO** **SYNCCLEARUP** **TSSYNCCLEAR**

PORT COMMANDS	[VAR]
apgLoadUnit [UNITID] PORTSTATUS	
➔ apgGetPort [PORTID] STATUS [VAR]	LINK MODULE_STATUS SPEED MODULE_SEQUENCE MODULE_TYPE MODULE_CLASS MODULE_TEMP
apgLoadPort [PORTID] INFO	
➔ apgGetPort [PORTID] INFO [VAR]	PORTNAME SPEEDCAP NSUBPORTS NSTREAMS
apgLoadPort [PORTID] MODULE	
➔ apgGetPort [PORTID] MODULE [VAR]	MODULE_TYPE ID VENDOR_ID VENDOR_NAME VENDOR_PN VENDOR_REV VENDOR_SN DATECODE
➔ apgGetPort [PORTID] MODULE [VAR] (continued) (Note: Copper SFPs only)	VALID CONFIG_AUTONEG CONFIG_FORCED_SPEED CONFIG_ADV_SPEED CONFIG_ADV_PAUSE CONFIG_MDI CONFIG_MS LINK FAULT RESOLVED RESOLVED _SPEED RESOLVED _DUPLEX RESOLVED _MDI RESOLVED _MS PAIRSWAP REMOTE_ADV_SPEED REMOTE_ADV_PAUSE CAP_AUTONEG CAP_FORCED_SPEED CAP_ADV_SPEED CAP_ADV_PAUSE CAP_MDI CAP_MS
apgSetPort [PORTID] [VAR] [VAL]	TOPOLOGY RATE TXENABLE LINKMODE PORTNAME
apgSetPort [PORTID] [VAR] [VAL] (Note: Copper SFPs only)	AUTONEG FORCED_SPEED ADV_SPEED ADV_PAUSE MDI MS
➔ apgApplyPort [PORTID]	STATE COPPERMODULE

apgControlPort [COMMAND] [PORTLIST]	**STARTTX** **STEPTX** **PAUSETX** **STOPTX** **CLEARCOUNTERS** **ZEROPORTSEQ** **STARTSYNCTX** **PORTCAPTURE** **DISABLECAPTURE** **CLEARCAPTURE** **CLEARANALYSIS**
apgLoadUnit [UNITID] COUNTERS	
➔ apgGetPort [PORTID] TXSTATS [VAR]	**TIMESTAMP** **BYTES** **GOOD_PACKETS** **PKT64** **PKT65_128** **PKT129_256** **PKT257_512** **PKT513_1024** **PKT1025_1536** **PKT1537_9000** **PKT9001_MAX** **TX_RUNTIME**
➔ apgGetPort [PORTID] RXSTATS [VAR]	**TIMESTAMP** **BYTES** **GOOD_PACKETS** **PKT_UNDERSIZE** **PKT_FRAGMENTS** **PKT_FCSERROR** **PKT_NOSFD** **PKT64** **PKT65_128** **PKT129_256** **PKT257_512** **PKT513_1024** **PKT1025_1536** **PKT1537_9000** **PKT9001_MAX**
apgLoadUnit [UNITID] RATES	
➔ apgGetPort [PORTID] RATES [VAR]	**TXPKTRATE** **TXBYTERATE** **TXBITRATE** **RXPKTRATE** **RXERRORRATE** **RXBYTERATE** **RXBITRATE**
apgLoadUnit [UNITID] PRBSSTATUS	
➔ apgGetPort [PORTID] PRBS	**MODE** **PATTERN** **LOCKED** **RUNTIME** **ERRORS** **BER**
apgSetPort [PORTID] [VAR] [VAL]	**PRBSMODE** **PRBSPATTERN**
➔ apgApplyPort [PORTID] PRBS	

apgLoadPort [PORTID] CAPTURE	
➔ apgGetPort [PORTID] CAPTURE CONFIG [VAR]	**AVAILABLE** **BUFFERSIZE** **FLAGS**
➔ apgGetPort [PORTID] CAPTURE PACKET	**TOTALPACKETS** **TIMESTAMP** **TSMODE** **LENGTH** **PKTERROR** **DATA**
apgLoadPort [PORTID] CAPTURE	
➔ apgGetPort [PORTID] ANALYSIS TXTX [VAR] **➔ apgGetPort [PORTID] ANALYSIS TXRX [VAR]** **➔ apgGetPort [PORTID] ANALYSIS RXRX [VAR]**	**CYCLES** **SAMPLES** **LIST** **MIN** **MAX** **MEAN** **MODE** **STDDEV**
➔ apgGetPort [PORTID] ANALYSIS SEQUENCE [VAR]	**CYCLES** **SAMPLES** **LIST** **MIN** **MAX** **RANGE** **MONOTONIC** **RPTCOUNT** **GAPCOUNT** **OOOCOUNT**

STREAM COMMANDS	**[VAR]**
apgLoadStream [STREAMID]	
➔ apgGetStream [STREAMID] CONFIG [VAR]	**ENABLE** **TX_MODE** **TX_CONTROL** **TX_BURST_SIZE** **TX_BURST_COUNT** **TX_BURST_IBG** **RATE_MODE** **RATE_VALUE** **SIZE_MODE** **PACKET_SIZE** **PACKET_SIZE_MAX**
➔ apgGetStream [STREAMID] HEADER HEADER_LIST	
➔ apgGetStream [STREAMID] HEADER [HDR] [FLD]	See Header Definition document [2] Eg for MACHEADER: **MACHEADER DA** **MACHEADER SA**
➔ apgGetStream [STREAMID] PAYLOAD [VAR]	**PAYLOAD_DATA** **DATA_CONTROL**
apgSetStream [STREAMID] DEFAULT	
➔ apgApplyStream [STREAMID]	
apgSetStream [STREAMID] CONFIG [VAR] [VAL]	**ENABLE** **TX_MODE** **TX_BURST_SIZE** **TX_BURST_COUNT** **TX_BURST_IBG** **RATE_MODE** **RATE_VALUE** **SIZE_MODE** **PACKET_SIZE** **PACKET_SIZE_MAX**
➔ apgApplyStream [STREAMID]	

apgSetStream [STREAMID] HEADER HEADER_LIST [HDRLIST]	Eg **"MACHEADER ETHERNET_II"**
➔ apgApplyStream [STREAMID]	
apgSetStream [STREAMID] HEADER [HDR] [FLD] [VAL]	See Header Definition document [2] Eg for MACHEADER: **MACHEADER DA 08:00:00:11:22:33** **MACHEADER SA 08:00:00:11:22:34**
➔ apgApplyStream [STREAMID]	
apgSetStream [STREAMID] HEADER [HDR] [FLD] [VAL] [MODE] [STEP][MIN][MAX]	See Header Definition document [2] Eg for MACHEADER DA: **MACHEADER DA 08:00:00:11:22:33** **INC 0:0:1:1:1:1 0:0:0:0:0:0** **FF:FF:FF:FF:FF:FF**
➔ apgApplyStream [STREAMID]	
apgSetStream [STREAMID] PAYLOAD [VAR] [VAL]	**PAYLOAD_DATA** **DATA_CONTROL**
➔ apgApplyStream [STREAMID]	
TOOLS	**Options**
apgGetVariables [COMMAND] {FUNCTION}	**VERSION** **UNIT INFO** **UNIT STATUS** **PORT INFO** **PORT STATUS** **PORT TXSTATS** **PORT RXSTATS** **PORT RATES** **PORT SETPORTCONFIG** **PORT COPPERMODULE** **STREAM CONFIG** **STREAM PAYLOAD**
apgGetHeaderList [STREAMID]	
apgGetHeaderFieldList [HDR]	See Header Definition document [2] Eg **HDR = MACHEADER**
apgGetHeaderFieldValue [HDR] [FLD] [VAR]	See Header Definition document [2] **LABEL** **DESCRIPTION** **LENGTH** **SCALE** **FORMAT** **EDITABLE**

APPENDIX B - SAMPLE APGSAVECONFIGURATION FILE

The output file from **apgSaveConfiguration [UNITID] [PORTLIST] {FILENAME}** is shown below. In this example, the default filename [SERIAL].apg was generated.

```
# FILENAME: APG000010.apg
```

The unit constraints API_VERSION and PRODUCT are mandatory, and the corresponding **apgApplyConfiguration [UNITID] [FILENAME]** will fail if either are missing, or do not match in the target unit [UNITID]. API_VERSION and PRODUCT **must** be in the following format:

```
# API_VERSION 160629
# PRODUCT     APG208
```

The UNITID in the saved file is not a valid TCL construct, and is automatically replaced with the **apgSaveConfiguration [UNITID]** command.

Eg: If [UNITID] = 4, `set PORTID {UNITID 1}` is processed as `set PORTID {4 1}`

The full configuration file APG000010.apg is shown below:

```
######################################################################################
#
# FILENAME: APG000010.apg
#
# Generated on Mon, 08 Aug 2016 at 14:27:05
#
# AXTRINET APG000010 CONFIGURATION FILE
#
######################################################################################
```

```
# UNIT CONSTRAINTS ###################################################################

# API_VERSION 160629
# PRODUCT     APG208
```

Unit API Version and Product Type.

```
# PORT 1 ############################################################################

set PORTID {UNITID 1}

apgSetPort $PORTID UPDATE 6
apgSetPort $PORTID MODULE_STATUS 113
apgSetPort $PORTID SPEED 8

apgApplyPort $PORTID STATE
```

Port Configuration for Unit UNITID Port 1

Apply port config

```
# STREAM 1.0.0 ##################################

set STREAMID {UNITID 1 0 0}
```

Stream Configuration for Port 1.0 Stream 0

```
apgSetStream $STREAMID CONFIG ENABLE 1
apgSetStream $STREAMID CONFIG TX_MODE 0
apgSetStream $STREAMID CONFIG RATE_MODE 2
apgSetStream $STREAMID CONFIG SIZE_MODE 3
apgSetStream $STREAMID CONFIG TX_CONTROL 0
apgSetStream $STREAMID CONFIG RATE_VALUE 86015
apgSetStream $STREAMID CONFIG TX_BURST_SIZE 1
apgSetStream $STREAMID CONFIG TX_BURST_COUNT 1
apgSetStream $STREAMID CONFIG TX_IBG 1
apgSetStream $STREAMID CONFIG PACKET_SIZE 64
apgSetStream $STREAMID CONFIG PACKET_SIZE_MAX 16000
```

Stream configuration

```
apgSetStream $STREAMID HEADER HEADER_LIST "MACHEADER VLAN_INNER VLAN_INNER \
  VLAN_INNER VLAN_INNER ETHERNET_II USERDEFINED16"
```

Stream Header List Header Configurations

```
apgSetStream $STREAMID HEADER MACHEADER DA 00:DD:01:01:00:00
apgSetStream $STREAMID HEADER MACHEADER SA 00:55:01:01:00:00 INC 01:02:03:04:05:06 \
  00:00:00:00:00:00 0A:0B:0C:0D:0E:0F
```

MACHEADER

apgSetStream $STREAMID HEADER VLAN_INNER PROTOCOL_ID 0x8100 apgSetStream $STREAMID HEADER VLAN_INNER PRIORITY 0 apgSetStream $STREAMID HEADER VLAN_INNER CFI 0 apgSetStream $STREAMID HEADER VLAN_INNER VID 1 INC 0 1 4095	VLAN #1
apgSetStream $STREAMID HEADER VLAN_INNER/2 PROTOCOL_ID 0x8101 apgSetStream $STREAMID HEADER VLAN_INNER/2 PRIORITY 0 apgSetStream $STREAMID HEADER VLAN_INNER/2 CFI 0 apgSetStream $STREAMID HEADER VLAN_INNER/2 VID 1 INC 0 2 10	VLAN #2
apgSetStream $STREAMID HEADER VLAN_INNER/3 PROTOCOL_ID 0x8102 apgSetStream $STREAMID HEADER VLAN_INNER/3 PRIORITY 0 apgSetStream $STREAMID HEADER VLAN_INNER/3 CFI 0 apgSetStream $STREAMID HEADER VLAN_INNER/3 VID 1	VLAN #3
apgSetStream $STREAMID HEADER VLAN_INNER/4 PROTOCOL_ID 0x8103 apgSetStream $STREAMID HEADER VLAN_INNER/4 PRIORITY 0 apgSetStream $STREAMID HEADER VLAN_INNER/4 CFI 0 apgSetStream $STREAMID HEADER VLAN_INNER/4 VID 1 INC 0 4 30	VLAN #4
apgSetStream $STREAMID HEADER ETHERNET_II ETHERTYPE 0x0812	ETHERNET_II
apgSetStream $STREAMID HEADER USERDEFINED16 DATA 0x0000 INC 0x0010 0x0000 0x2000	USERDEFINED16
apgSetStream $STREAMID PAYLOAD PAYLOAD_DATA "00 00 00 00 00 00 00 00" apgSetStream $STREAMID PAYLOAD DATA_CONTROL 0 apgSetStream $STREAMID PAYLOAD TIMESTAMP_CONTROL 0	Payload configuration
apgApplyStream $STREAMID	Apply stream config
# STREAM 1.0.1 ################################## set STREAMID {UNITID 1 0 1} apgSetStream $STREAMID CONFIG ENABLE 0 apgApplyStream $STREAMID	 Stream 1 ... Disabled Apply stream config
# STREAM 1.0.2 ################################## set STREAMID {UNITID 1 0 2} apgSetStream $STREAMID CONFIG ENABLE 0 apgApplyStream $STREAMID	 Stream 2 ... Disabled Apply stream config
# STREAM 1.0.3 ################################## set STREAMID {UNITID 1 0 3} apgSetStream $STREAMID CONFIG ENABLE 0 apgApplyStream $STREAMID	 Stream 3 ... Disabled Apply stream config
# STREAM 1.0.4 ################################## set STREAMID {UNITID 1 0 4} apgSetStream $STREAMID CONFIG ENABLE 0 apgApplyStream $STREAMID	 Stream 4 ... Disabled Apply stream config
# STREAM 1.0.5 ################################## set STREAMID {UNITID 1 0 5} apgSetStream $STREAMID CONFIG ENABLE 0 apgApplyStream $STREAMID	 Stream 5 ... Disabled Apply stream config
# STREAM 1.0.6 ################################## set STREAMID {UNITID 1 0 6} apgSetStream $STREAMID CONFIG ENABLE 0 apgApplyStream $STREAMID	 Stream 6 ... Disabled Apply stream config
# STREAM 1.0.7 ################################## set STREAMID {UNITID 1 0 7} apgSetStream $STREAMID CONFIG ENABLE 0 apgApplyStream $STREAMID	 Stream 7 ... Disabled Apply stream config
# END OF FILE ##	

Xentech Solutions Ltd
Suite 6 Stanta Business Centre
3 Soothouse Spring
St Albans
AL3 6PF
United Kingdom

Tel: +44 (0)1727 867795
Email: **support@axtrinet.com**