



Ethernet Packet Generator  
**TCL API GUIDE**  
Version 1.3



**APG4 APG8 APG208**

## Hardware Versions

This document applies to the following hardware versions:

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

## Software Versions

This document applies to the following software versions:

|                            | Version 1   | Version 2   |
|----------------------------|---|---|
| <b>APG API</b>             | <b>161016</b>   | <b>161016</b>   |
| <b>APG Unit Firmware</b>   | <b>Version 2.3</b>  | <b>Version 1.1</b>  |
| <b>APG Processing Core</b> | <p><b>Version 6.010E (10Gbps)</b><br/><b>Version 7.0103 (1Gbps)</b><br/>Datecode: 01 March 2019</p> | <p><b>Version 10.0</b><br/>Datecode: 12 December 2022</p> |
| <b>APG TCL API</b>         | <b>Version 1.3.1</b>  | <b>Version 1.3.1</b>                                      |

## Revision History

| Date             | Version | Changes  |
|------------------|---------|--|
| 12 August 2016   | 0.8     | Restricted customer release  |
| 7 June 2017      | 1.0     | General Release <ul style="list-style-type: none"> <li>• Added APG200</li> <li>• Added Timestamp Analysis (see Section 3.9)</li> <li>• Added API Initialisation (Section 4)</li> <li>• apgGetVersion renamed to apgGetApiVersion (Section 4.2)</li> <li>• apgOpen waits for unit ready (Section 5.1.1)</li> <li>• Removed inconsistencies in apgSaveConfiguration PORTLIST (see Section 6.1.1)</li> <li>• RXTIME replaces TIMESTAMP in apgGetPort CAPTURE PACKET command (see Section 8.2.8)</li> <li>• apgSetPort command simplified (Section 8.3.1)</li> <li>• Added apgSetPort TOPOLOGY command to enable switching between 40G / 4x10G modes (Section 8.3.1)</li> <li>• Added Deep Packet Capture (see Section 8.1.3)</li> <li>• Added Packet/Second Transmit Rate (Section 9.3.2)</li> <li>• Added Multi-Burst Transmit Mode (Section 9.3.2)</li> <li>• apgLoadStream command modified (Section 9.1.1)</li> </ul> |
| 1 September 2017 | 1.1     | <ul style="list-style-type: none"> <li>• Improved topology change behaviour</li> <li>• Improved device connection/disconnection behaviour</li> <li>• Improved capture download performance</li> </ul>  |
| 8 February 2018  | 1.1.3   | <ul style="list-style-type: none"> <li>• apgOpen returns value if unable to connect (Section 5.1.1)</li> <li>• Local capture buffer sizes increased to 64KB (10Gbps), and 256KB (40Gbps) (Section 3.8)</li> </ul>  |
| 27 February 2019 | 1.2     | <ul style="list-style-type: none"> <li>• Added apgUSBOpen command (Section 5.1)</li> <li>• Added 1Gbps Topology for SFP Ports (Sections 3.3.1 &amp; 8.3.1.1)</li> <li>• Added Copper Transceiver support:               <ul style="list-style-type: none"> <li>Extended apgGetPort MODULE command (Section 8.2.3)</li> <li>Extended apgSetPort command (Section 8.3.1.2)</li> <li>Added apgApplyPort COPPERMODULE (Section 8.4.2)</li> </ul> </li> </ul>   |
| 4 September 2019 | 1.2.2   | <ul style="list-style-type: none"> <li>• Corrected ETHERNET_II (underscore not hyphen)</li> <li>• Corrected minor typos and updated software versions</li> </ul>   |
| 21 February 2023 | 1.3     | <ul style="list-style-type: none"> <li>• Corrected minor typos and updated software versions</li> <li>• Added APGV2 support</li> <li>• Added APGV1 and APGV2 descriptions</li> <li>• Added Transmit Control description (Section 3.5)</li> <li>• Added Inter-Unit Synchronisation description (Section 3.6)</li> <li>• Added IPCONFIG option to apgLoadUnit and apgGetUnit commands (Section 7.1.2)</li> <li>• Added CLOCKSTATUS option to apgLoadUnit and apgGetUnit commands (Section 7.1.4)</li> <li>• Added apgSetUnit command to set clock modes and timestamp reset configuration (Section 7.3)</li> <li>• Added apgApplyUnit command to apply clock mode configuration (Section 7.4)</li> <li>• Added apgControlUnit command to control inter-unit synchronisation (Section 7.5)</li> <li>• Changed apgLoadPort CAPTURE warnings (Section 8.1.3) after download speed improvements in the TCL API</li> </ul>    |

| Date   | Version | Changes  |
|--------|---------|--|
| (cont) | 1.3     | <ul style="list-style-type: none"> <li>• apgLoadPort ANALYSIS command arguments modified (Section 8.1.4)</li> <li>• Corrected ADV_SPEED description in the copper module configuration table (Section 8.3.1.2)</li> <li>• Added 'armed' state to port states (Section 8.5)</li> <li>• Added apgControlPort [SYNC COMMAND] with STARTSYNCTX option to to arm ports for inter-unit synchronous start (Section 8.5.2)</li> <li>• Corrected apgControlPort DISABLECAPTURE description (Section 8.5.3)</li> <li>• Corrected ETHERNET_II (underscore not hyphen) (Section 9.2.3.2)</li> <li>• Extended apgGetVariables to include new unit options (Section 10.1.1)</li> <li>• Added apgGetHwVersion function (Section 10.3.1) to return unit version (APGV1 or APGV2)</li> <li>• Added apgConvertTicksToTime (Section 10.4.1) to convert clock ticks to time depending APG unit version.</li> </ul> |

## 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

Copyright © 2017-2023 Xentech Solutions Limited, all rights reserved. The information contained in this document is the property of Xentech Solutions Limited. No part of this publication shall be reproduced, stored or transmitted in any form or by any means without the prior written permission of Xentech Solutions Limited.

Axtrinet™ is a trading name and registered trademark of Xentech Solutions Ltd.

## 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](http://www.axtrinet.com/documentation) for the latest documentation.

## Glossary

|       |   |
|-------|---|
| APG   | Axtrinet Packet Generator                 |
| API   | Application Programming Interface         |
| FCS   | Frame Checksum                            |
| IBG   | Inter-Burst Gap                           |
| IFG   | Inter-Frame Gap                           |
| pps   | Packets per Second                        |
| 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

|            |  |           |
|------------|--|-----------|
| <b>1.</b>  | <b>INTRODUCTION .....</b>                                  | <b>11</b> |
| <b>1.1</b> | <b>Main Features .....</b>                                 | <b>11</b> |
| <b>1.2</b> | <b>Benefits .....</b>                                      | <b>11</b> |
| <b>1.3</b> | <b>Software Licences and Support.....</b>                  | <b>11</b> |
| <b>1.4</b> | <b>Contact Details .....</b>                               | <b>12</b> |
| <b>2.</b>  | <b>INSTALLATION .....</b>                                  | <b>13</b> |
| <b>2.1</b> | <b>Minimum System Requirements .....</b>                   | <b>13</b> |
| <b>2.2</b> | <b>Prerequisites .....</b>                                 | <b>13</b> |
| <b>2.3</b> | <b>Testing the Installation .....</b>                      | <b>13</b> |
| 2.3.1      | Windows Environment.....                                   | 13        |
| 2.3.2      | Linux Environment.....                                     | 13        |
| <b>2.4</b> | <b>Example Files.....</b>                                  | <b>14</b> |
| <b>3.</b>  | <b>TEST ENVIRONMENT &amp; CONCEPTS .....</b>               | <b>15</b> |
| <b>3.1</b> | <b>Definitions .....</b>                                   | <b>16</b> |
| <b>3.2</b> | <b>APG Configuration .....</b>                             | <b>17</b> |
| <b>3.3</b> | <b>Port Topology .....</b>                                 | <b>17</b> |
| 3.3.1      | SFP+/SFP Ports .....                                       | 17        |
| 3.3.2      | QSFP+ Ports (APG208 only).....                             | 17        |
| <b>3.4</b> | <b>Transmit Configuration.....</b>                         | <b>18</b> |
| <b>3.5</b> | <b>Port Control .....</b>                                  | <b>20</b> |
| <b>3.6</b> | <b>Inter-Unit Synchronisation (APGV2 units only) .....</b> | <b>21</b> |
| 3.6.1      | Clock Synchronisation .....                                | 21        |
| 3.6.2      | Timestamp Synchronisation .....                            | 22        |
| 3.6.3      | Transmit Synchronisation .....                             | 23        |
| <b>3.7</b> | <b>Receive Path.....</b>                                   | <b>24</b> |
| <b>3.8</b> | <b>Packet Capture .....</b>                                | <b>24</b> |
| <b>3.9</b> | <b>Packet Analysis.....</b>                                | <b>25</b> |
| <b>4.</b>  | <b>API INITIALISATION .....</b>                            | <b>26</b> |
| <b>4.1</b> | <b>Initialisation .....</b>                                | <b>26</b> |
| 4.1.1      | Windows Environment.....                                   | 26        |
| 4.1.2      | Linux Environment.....                                     | 26        |
| <b>4.2</b> | <b>Version Command - apgGetApiVersion .....</b>            | <b>26</b> |
| 4.2.1      | apgGetApiVersion [VAR].....                                | 26        |
| <b>5.</b>  | <b>CONNECTION COMMANDS .....</b>                           | <b>27</b> |
| <b>5.1</b> | <b>Open Connection .....</b>                               | <b>27</b> |
| 5.1.1      | apgOpen [IP-ADDRESS] .....                                 | 27        |
| 5.1.2      | apgUSBOpen [SERIAL] .....                                  | 27        |
| <b>5.2</b> | <b>Close Connection - apgClose.....</b>                    | <b>28</b> |
| 5.2.1      | apgClose [UNITID].....                                     | 28        |
| <b>6.</b>  | <b>CONFIGURATION COMMANDS.....</b>                         | <b>29</b> |
| <b>6.1</b> | <b>Save Configuration .....</b>                            | <b>29</b> |
| 6.1.1      | apgSaveConfiguration [UNITID] [PORTLIST] {FILENAME} .....  | 29        |
| <b>6.2</b> | <b>Apply Configuration .....</b>                           | <b>30</b> |
| 6.2.1      | apgApplyConfiguration [UNITID] [FILENAME].....             | 30        |



|            |  |           |
|------------|--|-----------|
| <b>7.</b>  | <b>UNIT COMMANDS.....</b>                                      | <b>31</b> |
| <b>7.1</b> | <b>Load Unit Data - apgLoadUnit .....</b>                      | <b>32</b> |
| 7.1.1      | apgLoadUnit [UNITID] INFO .....                                | 32        |
| 7.1.2      | apgLoadUnit [UNITID] IPCONFIG .....                            | 32        |
| 7.1.3      | apgLoadUnit [UNITID] STATUS.....                               | 32        |
| 7.1.4      | apgLoadUnit [UNITID] CLOCKSTATUS (APGV2 only) .....            | 32        |
| 7.1.5      | apgLoadUnit [UNITID] PORTSTATUS .....                          | 33        |
| 7.1.6      | apgLoadUnit [UNITID] COUNTERS .....                            | 33        |
| 7.1.7      | apgLoadUnit [UNITID] RATES {DURATION} .....                    | 33        |
| <b>7.2</b> | <b>Get Unit Data - apgGetUnit .....</b>                        | <b>34</b> |
| 7.2.1      | apgGetUnit [UNITID] INFO [VAR] .....                           | 34        |
| 7.2.2      | apgGetUnit [UNITID] IPCONFIG [VAR] .....                       | 35        |
| 7.2.3      | apgGetUnit [UNITID] STATUS [VAR] .....                         | 35        |
| 7.2.4      | apgGetUnit [UNITID] CLOCKSTATUS [VAR] (APGV2 only) .....       | 36        |
| 7.2.5      | apgGetUnit [UNITID] TSRESETMODE (APGV2 only).....              | 36        |
| <b>7.3</b> | <b>Set Unit Data - apgSetUnit (APGV2 only) .....</b>           | <b>37</b> |
| 7.3.1      | apgSetUnit [UNITID] [VAR] [VAL] (APGV2 only) .....             | 38        |
| <b>7.4</b> | <b>Apply Unit Data - apgApplyUnit .....</b>                    | <b>38</b> |
| 7.4.1      | apgApplyUnit [UNITID] CLOCKMODE.....                           | 38        |
| <b>7.5</b> | <b>Control Commands - apgControlUnit.....</b>                  | <b>39</b> |
| 7.5.1      | apgControlUnit [COMMAND] [UNITID] (APGV2 only) .....           | 39        |
| 7.5.2      | apgControlUnit [SYNC COMMAND] [UNITID] (APGV2 only).....       | 39        |
| <b>8.</b>  | <b>PORT COMMANDS .....</b>                                     | <b>41</b> |
| <b>8.1</b> | <b>Load Port Data - apgLoadPort.....</b>                       | <b>43</b> |
| 8.1.1      | apgLoadPort [PORTID] INFO .....                                | 43        |
| 8.1.2      | apgLoadPort [PORTID] MODULE .....                              | 43        |
| 8.1.3      | apgLoadPort [PORTID] CAPTURE [VAR] {NUMPKT}.....               | 43        |
| 8.1.4      | apgLoadPort [PORTID] ANALYSIS {FIRSTPKT} {NUMPKT} {TSID} ..... | 44        |
| <b>8.2</b> | <b>Get Port Data - apgGetPort.....</b>                         | <b>44</b> |
| 8.2.1      | apgGetPort [PORTID] INFO [VAR].....                            | 44        |
| 8.2.2      | apgGetPort [PORTID] STATUS [VAR].....                          | 46        |
| 8.2.3      | apgGetPort [PORTID] MODULE [VAR].....                          | 47        |
| 8.2.4      | apgGetPort [PORTID] TXSTATS [VAR] .....                        | 51        |
| 8.2.5      | apgGetPort [PORTID] RXSTATS [VAR] .....                        | 52        |
| 8.2.6      | apgGetPort [PORTID] RATES [VAR].....                           | 53        |
| 8.2.7      | apgGetPort [PORTID] CAPTURE CONFIG [VAR].....                  | 54        |
| 8.2.8      | apgGetPort [PORTID] CAPTURE PACKET TOTALPACKETS .....          | 55        |
| 8.2.9      | apgGetPort [PORTID] CAPTURE PACKET [VAR] [PKTNUM].....         | 55        |
| 8.2.10     | apgGetPort [PORTID] ANALYSIS [TYPE] [VAR] .....                | 56        |
| <b>8.3</b> | <b>Set Port Configuration - apgSetPort .....</b>               | <b>57</b> |
| 8.3.1      | apgSetPort [PORTID] [VAR] [VAL] .....                          | 57        |
| <b>8.4</b> | <b>Apply Port Configuration - apgApplyPort.....</b>            | <b>59</b> |
| 8.4.1      | apgApplyPort [PORTID] STATE .....                              | 59        |
| 8.4.2      | apgApplyPort [PORTID] COPPERMODULE .....                       | 59        |
| <b>8.5</b> | <b>Control Commands - apgControlPort .....</b>                 | <b>59</b> |
| 8.5.1      | apgControlPort [COMMAND] [PORTLIST] .....                      | 61        |
| 8.5.2      | apgControlPort [SYNC COMMAND] [PORTLIST] (APGV2 only) .....    | 61        |
| 8.5.3      | apgControlPort [CAPTURE] [PORTLIST] .....                      | 62        |
| 8.5.4      | apgControlPort CLEARANALYSIS [PORTLIST].....                   | 62        |
| <b>9.</b>  | <b>STREAM COMMANDS.....</b>                                    | <b>63</b> |
| <b>9.1</b> | <b>Load Stream Configuration - apgLoadStream .....</b>         | <b>64</b> |
| 9.1.1      | apgLoadStream [STREAMID].....                                  | 64        |
| <b>9.2</b> | <b>Get Stream Configuration - apgGetStream.....</b>            | <b>64</b> |
| 9.2.1      | apgGetStream [STREAMID] CONFIG [VAR] .....                     | 65        |
| 9.2.2      | apgGetStream [STREAMID] HEADER HEADER_LIST .....               | 66        |

|  |   |           |
|--|---|-----------|
| 9.2.3  | apgGetStream [STREAMID] HEADER [HDR] [FLD] .....                          | 66        |
| 9.2.4  | apgGetStream [STREAMID] PAYLOAD [VAR] .....                               | 67        |
| <b>9.3</b>   | <b>Set Stream Configuration - apgSetStream .....</b>                      | <b>68</b> |
| 9.3.1  | apgSetStream [STREAMID] DEFAULT .....                                     | 69        |
| 9.3.2  | apgSetStream [STREAMID] CONFIG [VAR] [VAL] .....                          | 69        |
| 9.3.3  | apgSetStream [STREAMID] HEADER HEADER_LIST [HDRLIST] .....                | 70        |
| 9.3.4  | apgSetStream [STREAMID] HEADER [HDR] [FLD] [VAL] .....                    | 71        |
| 9.3.5  | apgSetStream [STREAMID] HEADER [HDR] [FLD] [VAL] [MODE] [STEP][MIN][MAX]. | 72        |
| 9.3.6  | apgSetStream [STREAMID] PAYLOAD [VAR] [VAL].....                          | 72        |
| <b>9.4</b>   | <b>Apply Stream Configuration - apgApplyStream.....</b>                   | <b>73</b> |
| 9.4.1  | apgApplyStream [STREAMID].....  | 73        |
| <b>10.</b>   | <b>TOOLS.....</b>   | <b>74</b> |
| <b>10.1</b>  | <b>Command Tools .....</b>  | <b>74</b> |
| 10.1.1   | apgGetVariables [COMMAND] {FUNCTION} .....                                | 74        |
| <b>10.2</b>  | <b>Header Tools.....</b>  | <b>74</b> |
| 10.2.1   | apgGetHeaderList [STREAMID] .....   | 74        |
| 10.2.2   | apgGetHeaderFieldList [HDR] .....   | 75        |
| 10.2.3   | apgGetHeaderFieldValue [HDR] [FLD] [VAR] .....                            | 75        |
| <b>10.3</b>  | <b>Hardware Version Tools.....</b>  | <b>76</b> |
| 10.3.1   | apgGetHwVersion [UNITID] .....  | 76        |
| <b>10.4</b>  | <b>Timestamp Tools .....</b>  | <b>76</b> |
| 10.4.1   | apgConvertTicksToTime [UNITID] [TICKS] {UNITS} .....                      | 76        |
| <b>APPENDIX A - QUICK REFERENCE GUIDE .....</b>            |   | <b>77</b> |
| <b>APPENDIX B - SAMPLE APGSAVECONFIGURATION FILE .....</b> |   | <b>82</b> |

## 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](mailto:support@axtrinet.com)  
Sales: [sales@axtrinet.com](mailto:sales@axtrinet.com)

**Web Site:** [www.axtrinet.com](http://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

|                   |  |
|-------------------|--|
| <b>Processor</b>  | Pentium-class processor or equivalent  |
| <b>Memory</b>     | 2GB (4GB recommended)  |
| <b>Disk Space</b> | 15MB   |
| <b>OS</b>         | 64bit (x86_64) Linux systems<br>Microsoft Windows 7<br>Microsoft Windows 8.x<br>Microsoft Windows 10.x<br>Microsoft Windows 11.x |
| <b>Interfaces</b> | Minimum: USB 2.0<br>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 APGV1.3.1  
Build Date 1675361105  
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 APGV1.3.1  
Build Date 1675361105  
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.

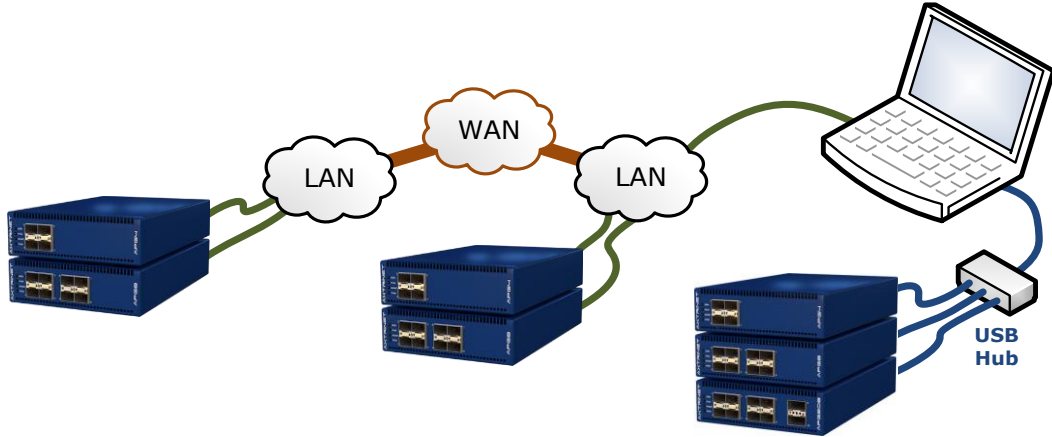
## 2.4 EXAMPLE FILES

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

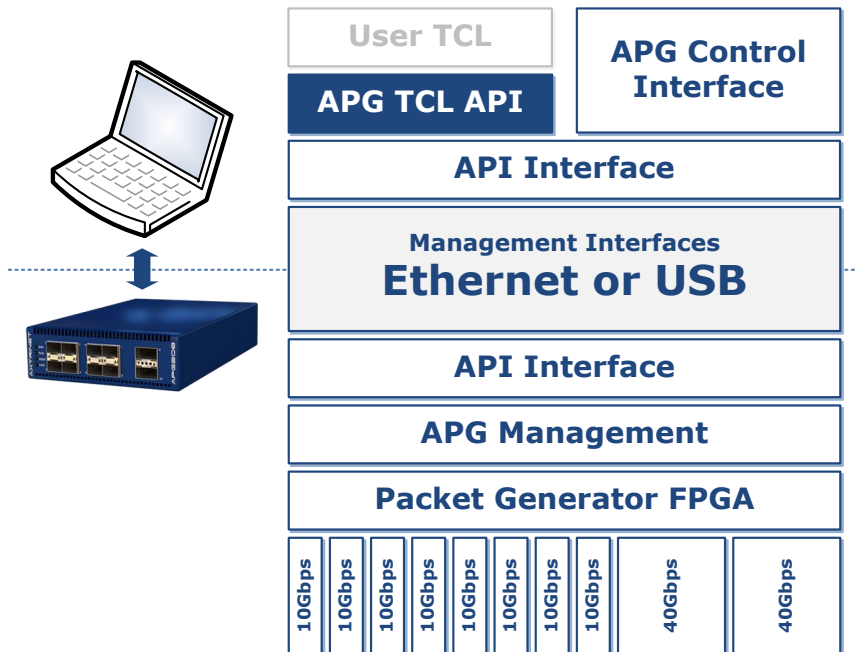
- |    |                               |   |
|----|-------------------------------|---|
| 1. | demo_init.tcl                 | Initialise Axtrinet APG API Package   |
| 2. | demo_connection.tcl           | Open and Close connections to the Axtrinet APG  |
| 3. | demo_save_recall.tcl          | Save and 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.<br>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,<br>synchronous timestamp reset, synchronous start,<br>deep capture, download and display. |

### 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

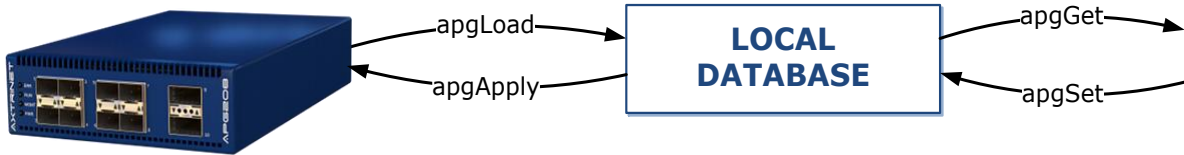
|                |  |
|----------------|--|
| <b>UNIT</b>    | <p>A unit is a single physical Axtrinet Packet Generator (APG4, APG8, APG200 or APG208).</p> <p>A unit is assigned a UNITID when the management connection is first opened using the APG TCL API. The UNITID is fixed for the duration of the TCL session, and is used to uniquely identify a unit during the session. The UNITIDs will be reassigned if the TCL session is restarted.</p> <p>Unit commands enable management connections; provide access to the unit status (See Section 4).</p>  |
| <b>PORT</b>    | <p>A PORT is a physical aperture on the unit (SFP+/SFP or QSFP+).</p> <p>Ports are numbered from 1.</p> <p>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.</p> <p>Port commands allow access to the modules; port capabilities and configuration; and port counters (see Section 8).</p>   |
| <b>SUBPORT</b> | <p>Where a QSFP+ port can be configured into different topologies (eg 40Gbps or 4x10Gbps), the PORTID is qualified with a SUBPORT.</p> <p>Subports are numbered from 1.</p> <p>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.</p> <p>Subports use the Port commands to gain access to the modules; port capabilities and configuration; and port counters (see Section 8).</p>  |
| <b>MODULE</b>  | <p>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)</p>  |
| <b>STREAM</b>  | <p>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.</p> <p>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}).</p> <p>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}).</p> <p>The STREAMID is fixed for the duration of the TCL session, and is used to uniquely identify a stream during the session.</p> <p>The outputs from the eight stream generators are multiplexed into a single stream for transmission from a port.</p> <p>Stream commands allow the stream configuration to be read from the unit, modified and written to the unit (See Section 9)</p> |



### 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.

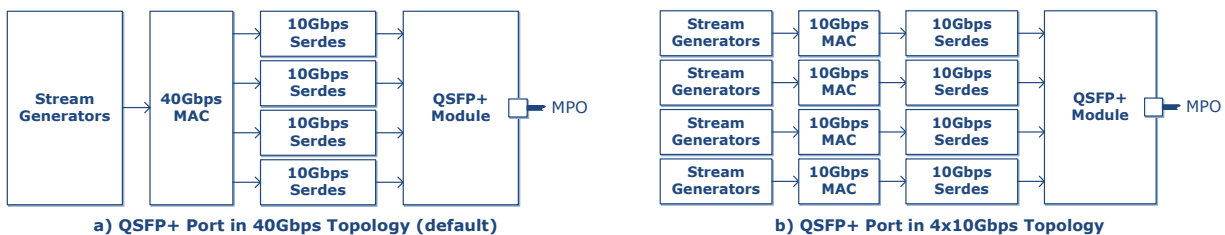


An APGV2 unit can be configured with a mix of port speeds.

#### 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.



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

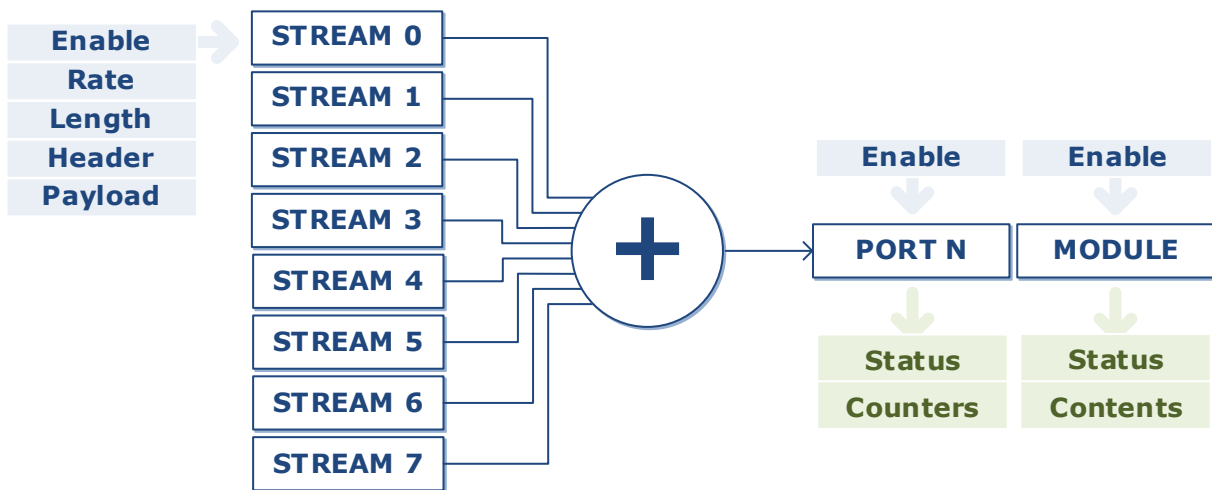


Mixed topology of the QSFP+ ports is **only** available with Port 9 at 4x10Gbps, and Port 10 at 40Gbps. Setting Port 9 to 40Gbps or Port 10 to 4x10Gbps topologies will automatically switch the other port into the same mode.

### 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

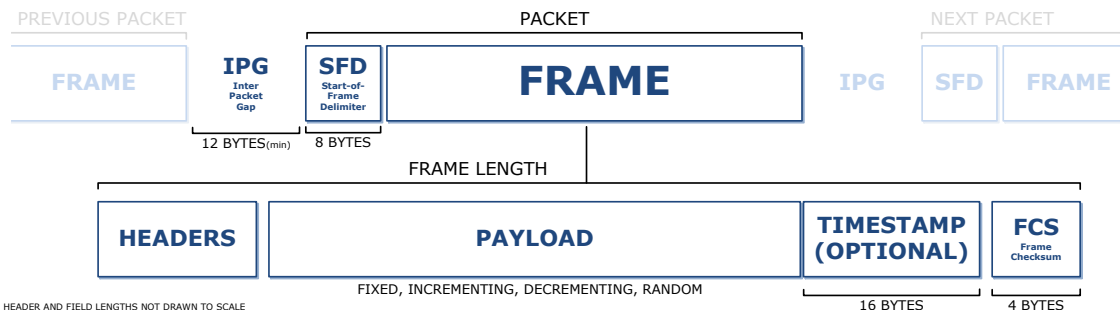


The streams are processed in a round-robin sequence, transmitting a packet if it queued and ready to send.



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.

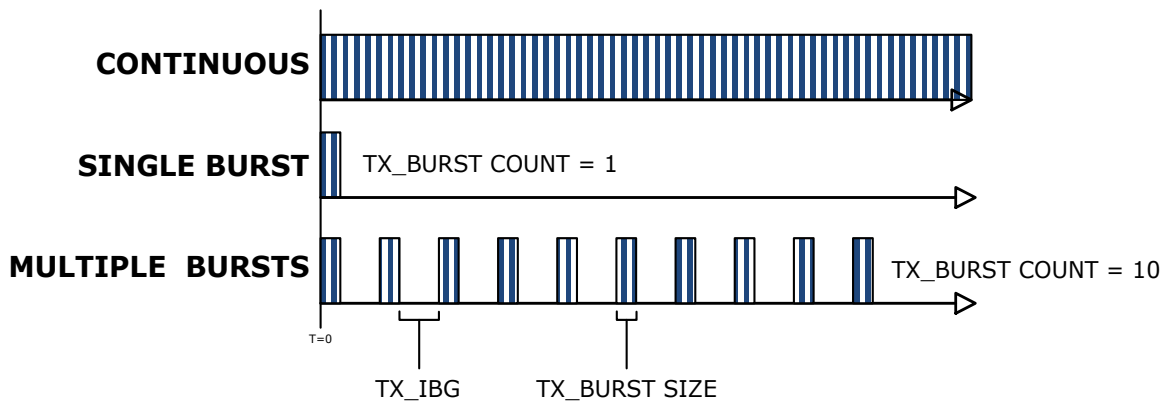
The Stream Generator defines an Ethernet frame:



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.



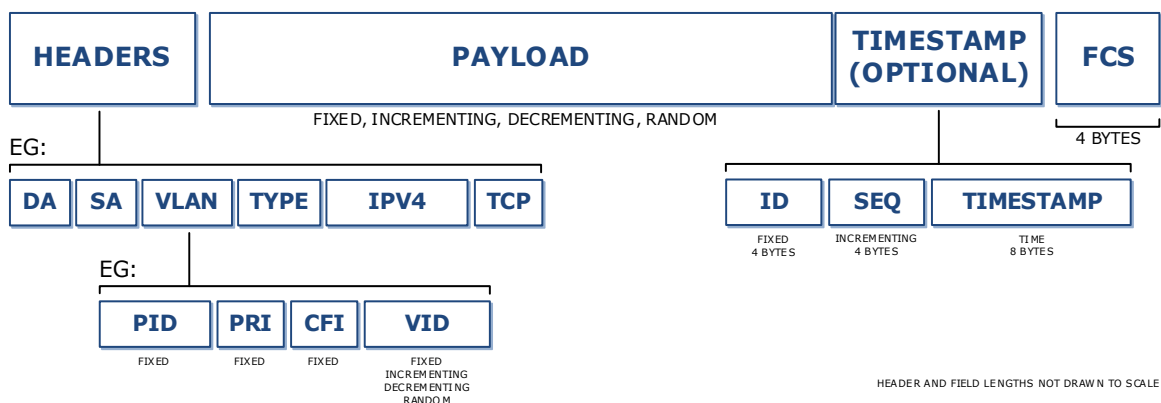
Care should be taken to ensure that the cumulative stream transmit rates does not exceed the port capacity.

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 3.0.**



**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.**

**It is not possible to reset the Sequence Number in APG Control Interface Version 3.0.**



**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

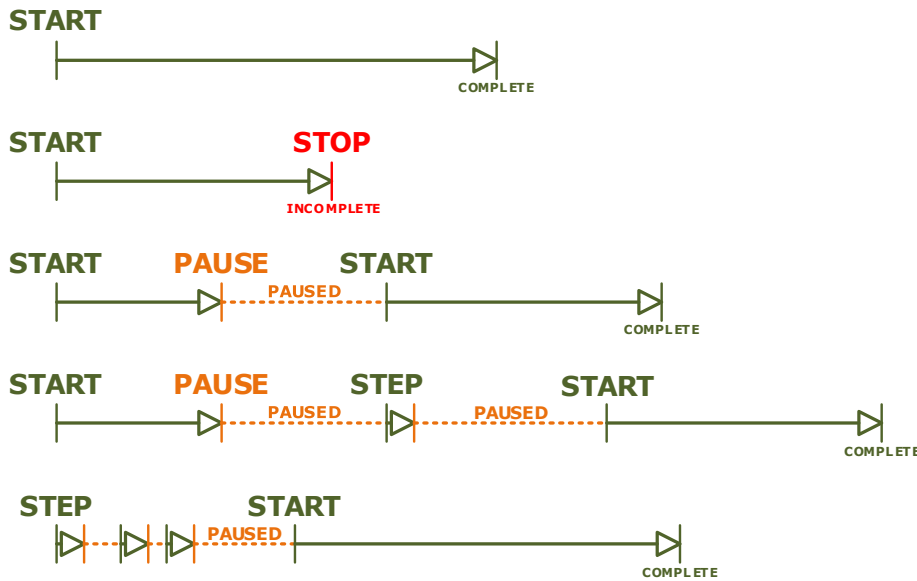
Packet transmission is controlled with the **apgControlPort** command (Section 8.5.1) that operates on ports specified by [PORTLIST]. [PORTLIST] can contain a single port, or a list of ports.

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

**apgControlPort STOPTH** 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 STEPTH** generates only the next packet, then pauses.



### 3.6 INTER-UNIT SYNCHRONISATION (APGV2 UNITS ONLY)

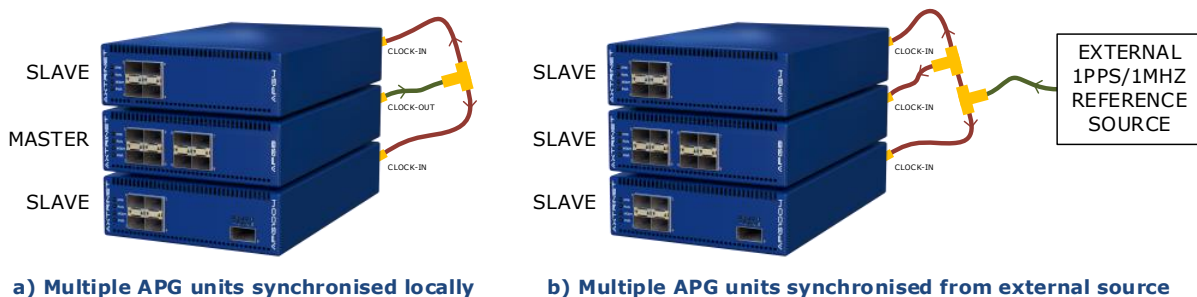
The Axtrinet unit uses an internal oscillator to clock a PLL-based clock synthesiser to generate the system clocks for the Ethernet Packet Generator, Ethernet and USB management interfaces. The clocks on separate APG units are free-running and will not be frequency or phase locked to each other.

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.

#### 3.6.1 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].



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.

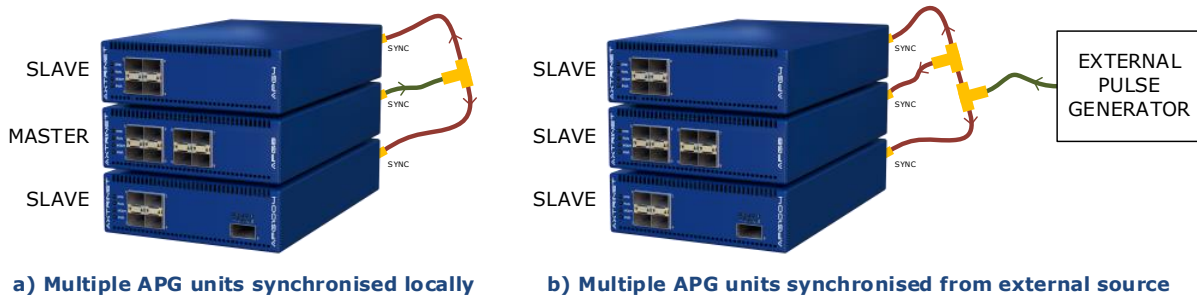


**System clocks on the slave units will be frequency and phase locked to the master clock. If the CLOCK-OUT is enabled on the slave units, the 1PPS or 1MHZ output on the synchronised units will not be phase aligned.**

### 3.6.2 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].



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. Armed 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.

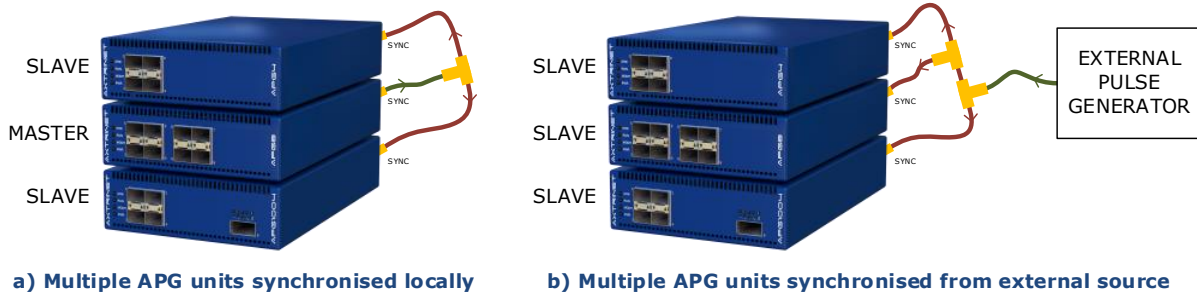
The 'synchronised' ports reset the timestamp on the rising edge of the synchronisation pulse.

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 SYNCLEARUP** command.

### 3.6.3 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].



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.

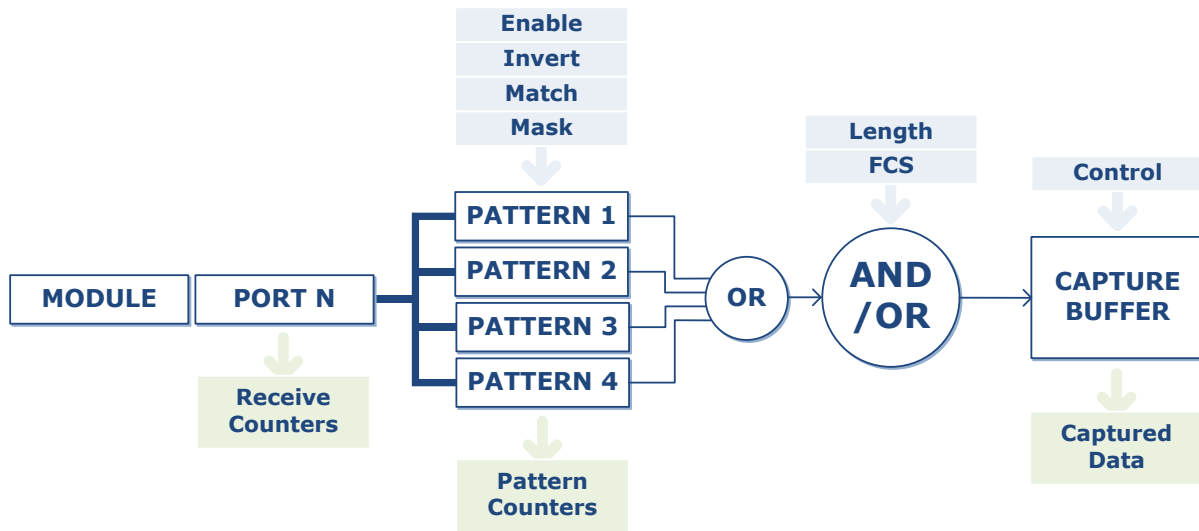
The 'synchronised' ports start transmitting on the rising edge of the synchronisation pulse.



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.

### 3.7 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.**

### 3.8 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.

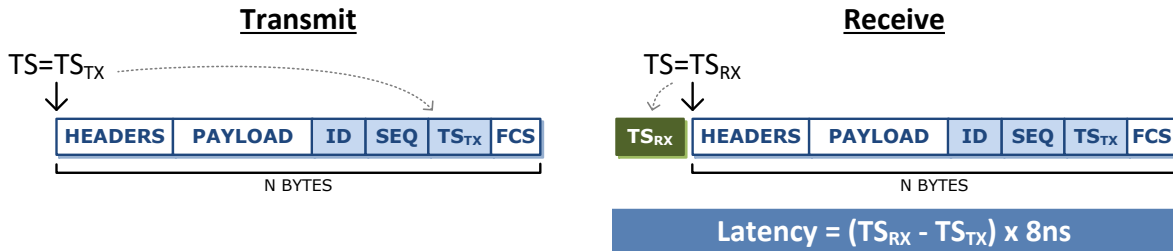


### 3.9 PACKET ANALYSIS

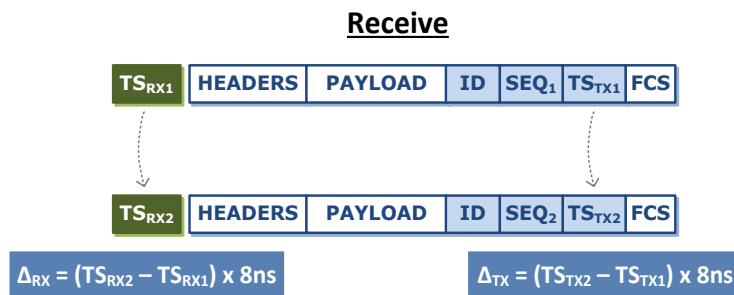
The Packet Analysis function provides basic analysis of the packet timestamp fields from the captured packets, and analysed with the **apgLoadPort ANALYSIS** command.

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 V1.3.1
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 V1.3.1
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.



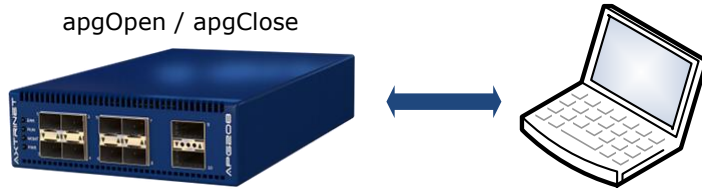
A **LOAD** command is not required before the **apgGetApiVersion**

#### 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    |
|--------------------|---------|--|------------|
| <b>COMPANY</b>     | 21 char | Xentech Solutions Ltd                      |            |
| <b>BRAND</b>       | 8 char  | Axtrinet                                   |            |
| <b>DESCRIPTION</b> | 11 char | APG TCL API                                |            |
| <b>VERSION</b>     | 10 char | Axtrinet TCL API Version currently running | APGV1.3.1  |
| <b>BUILD_DATE</b>  | 32 bits | Build date of the TCL API                  | 1675361105 |
| <b>API_VERSION</b> | 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

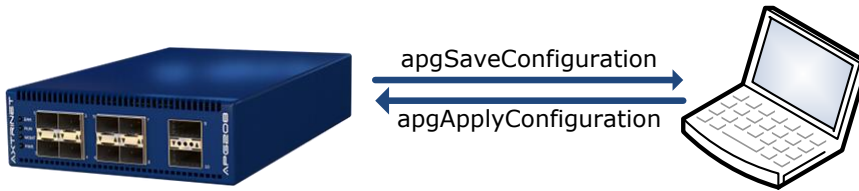
### 5.2.1 apgClose [UNITID]

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: | <pre># Instantiate APG TCL API (Windows) source "C:/Program Files/Axtrinet/APG/axtrinetApi.tcl"</pre>  | ← Load API                      |
|     | <pre># 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] puts "Opened connection to Unit \$UNITID at \$IPADDRESS"</pre> | Open Connection<br>→ UNITID = 1 |
|     | <pre>puts "Saving configuration" set FILENAME APGexample.apg apgSaveConfiguration \$UNITID ALL \$FILENAME</pre>  | Save Configuration              |
|     | <pre>puts "Closing Connection" apgClose \$UNITID</pre>   | Close Connection                |
|     | <pre>#####</pre>   |                                 |
|     | <pre>set UNITID [apgOpen \$IPADDRESS] puts "Opened connection to Unit \$UNITID at \$IPADDRESS"</pre>   | Open Connection<br>→ UNITID = 2 |
|     | <pre>puts "Loading saved configuration \$FILENAME" apgApplyConfiguration \$UNITID \$FILENAME</pre>   | Load Configuration              |
|     | <pre>puts "Closing Connection" apgClose \$UNITID</pre>   | 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.

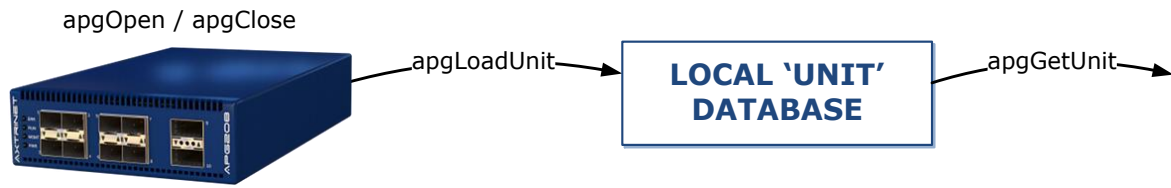
## 6.2 APPLY CONFIGURATION

### 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



|                       |   |
|-----------------------|---|
| <b>apgLoadUnit</b>    | <b>apgLoadUnit [UNITID] INFO</b><br><b>apgLoadUnit [UNITID] IPCONFIG</b><br><b>apgLoadUnit [UNITID] STATUS</b><br><b>apgLoadUnit [UNITID] CLOCKSTATUS</b>                     |
|                       | <b>apgLoadUnit [UNITID] PORTSTATUS</b><br><b>apgLoadUnit [UNITID] COUNTERS</b><br><b>apgLoadUnit [UNITID] RATES</b>   |
| <b>apgGetUnit</b>     | <b>apgGetUnit [UNITID] INFO [VAR]</b><br><b>apgGetUnit [UNITID] IPCONFIG [VAR]</b><br><b>apgGetUnit [UNITID] STATUS [VAR]</b><br><b>apgGetUnit [UNITID] CLOCKSTATUS [VAR]</b> |
| <b>apgSetUnit</b>     | <b>apgSetUnit [UNITID] [VAR] [VAL]</b>  |
| <b>apgApplyUnit</b>   | <b>apgApplyUnit [UNITID] CLOCKMODE</b>  |
| <b>apgControlUnit</b> | <b>apgControlUnit [UNITID] [COMMAND]</b>  |

```

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"

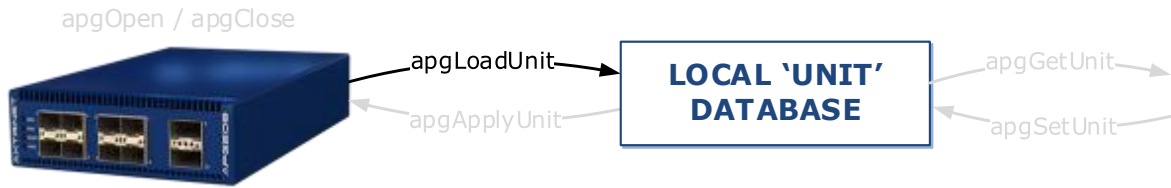
     # Load unit status into local database
     apgLoadUnit $UNITID STATUS → 1

     # 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.

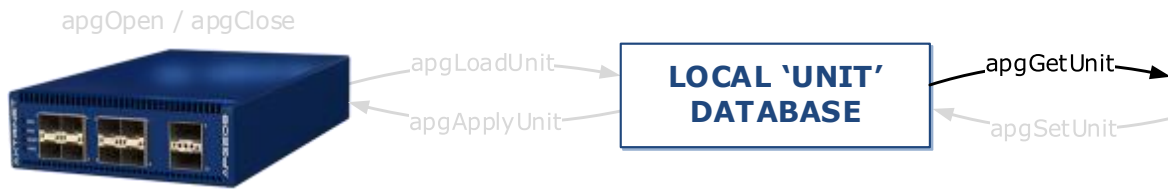
The rate calculator takes two counter readings, separated by {DURATION} milliseconds (eg 2000 = 2sec), to calculate the transmit and receive frame, byte, bit and error rates. If {DURATION} is omitted, a duration of 1000ms is assumed.

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**.

## 7.2 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  |
|-----------------------|----------|--|
| <b>API_VERSION</b>    | 32 bits  | API Version of the Firmware eg 161016  |
| <b>PORT_COUNT</b>     | 8 bits   | Number of physical ports on the unit   |
| <b>API_MINIMUM</b>    | 32 bits  | Minimum API supported by the unit<br>[ <b>apgGetApiVersion API_VERSION</b> ] must be greater than <b>API_MINIMUM</b> , and should ideally match. |
| <b>SERIAL</b>         | 16 chars | Serial Number eg APG000006   |
| <b>PRODUCT</b>        | 12 chars | Product Code eg APG208   |
| <b>FW_VERSION</b>     | 8 chars  | Firmware Version loaded onto the UNIT eg 1.2-1   |
| <b>FW_BUILDDATE</b>   | 32 bits  | Build timestamp of the Firmware Version loaded onto the UNIT eg 1457534868   |
| <b>FPGA_VERSION</b>   | 8 chars  | Firmware Version loaded onto the UNIT eg 6.010E  |
| <b>FPGA_BUILDDATE</b> | 32 bits  | Build timestamp of the Firmware Version loaded onto the UNIT eg 1455890890   |
| <b>HW_VERSION</b>     | 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                                |
|------------------|---------|--|
| <b>IPADDRESS</b> | 32 bits | IP Address eg 0xC0A8C8A7 = 192.168.200.167 |
| <b>IPMASK</b>    | 32 bits | IP Mask eg 0xFFFFFFFF00 = 255.255.255.0    |
| <b>IPGATEWAY</b> | 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   |
|--|---------|---|
| <b>UPTIME</b>  | 64 bits | Time since the unit was powered up (in ms)  |
| <b>TEMP</b>  | 2 bits  | Temperature Flags, calculated from the preset $T_{OPERATE}$ , $T_{WARNING}$ and $T_{SHUTDOWN}$ temperature thresholds:<br>0 = OK $T \leq T_{OPERATE} + 3$<br>1 = HIGH $T_{OPERATE} + 3 \leq T < T_{WARNING}$<br>2 = WARNING $T_{WARNING} \leq T < T_{SHUTDOWN}$<br>3 = ALARM $T > T_{WARN}$ |
| <b>FAN</b>   | 1 bit   | 0 = Fan Failed<br>1 = Fan OK  |
| <b>SELFTEST</b>  | 1 bit   | 0 = Self-Test Failed<br>1 = Self-Test Passed  |
| <div data-bbox="603 1585 678 1659" data-label="Image"> </div> <div data-bbox="699 1581 1441 1680" data-label="Text"> <p>Self-Test verifies that the internal power supplies are within tolerance and that the CPU interfaces are operational.</p> </div> |         |   |
| <b>READY</b>   | 1 bit   | 0 = Unit not ready (FPGA Booting)<br>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   |
|-------------------|--------|---|
| <b>CLKINMODE</b>  | 3 bits | 0 = Not Set (use default)<br>1 = Internal (default)<br>3 = External 1PPS (1Hz)<br>4 = External 1MHz       |
| <b>CLKOUTMODE</b> | 3 bits | 0 = Not Set (use default)<br>1 = 1PPS (1Hz) output (default)<br>2 = 1MHz output                           |
| <b>PLLSTATE</b>   | 4 bits | 0 = Invalid<br>1 = Unknown<br>2 = Internal<br>3 = Holdover<br>4 = Acquiring<br>5 = Tracking<br>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.

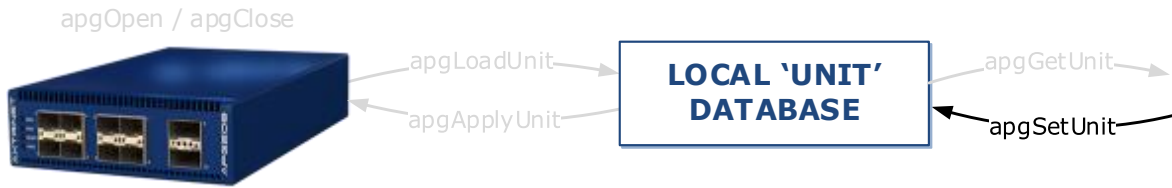
### 7.2.5 apgGetUnit [UNITID] TSRESETMODE (APGV2 only)

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".

### 7.3 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
after 1000
apgLoadUnit $UNITID2 CLOCKSTATUS ← Wait
set CLOCKSTATE [apgGetUnit $UNITID2 CLOCKSTATUS PLLSTATE] ← Load Clock Status
if { $CLOCKSTATE != 6 } { error "Not locked after 1sec" } ← Get Clock Status

# 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 (APGV2 only)

Sets the clock input and output modes (Section 3.6.1)

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

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 (APGV2 only)

Sets the timestamp reset mode (see Section 3.6.2)

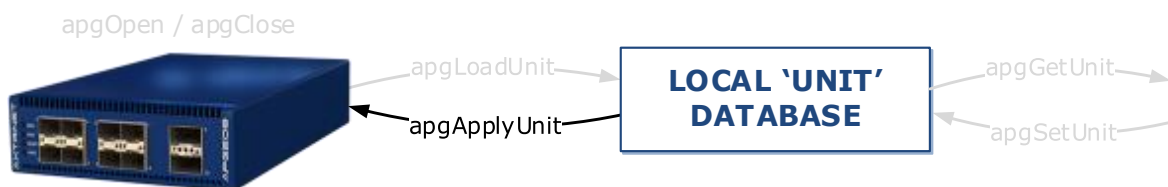
| [VAL]              | DESCRIPTION   | Default     |
|--------------------|---|-------------|
| <b>TSRESETMODE</b> | <b>ZERO</b> = Zero the timestamp on reset<br><b>TIME</b> = Set the timestamp to the current time on reset | <b>ZERO</b> |

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

### 7.5.1 apgControlUnit [COMMAND] [UNITID] (APGV2 only)

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  |
|----------------|--|
| <b>TSCLEAR</b> | Immediately reset the timestamp on [UNITID].<br><br>The timestamp is reset to ZERO or TIME with the <b>apgSetUnit TSRESETMODE</b> command. |

### 7.5.2 apgControlUnit [SYNC COMMAND] [UNITID] (APGV2 only)

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   |
|--------------------|---|
| <b>TSSYNCCLEAR</b> | Arm synchronous timestamp reset on [UNITID].<br><br>To synchronously reset the timestamps on multiple units, the SYNC interfaces on all units must be connected together with co-axial cables.<br><br>Synchronous timestamp reset is started on the rising edge of a pulse on the SYNC interface, either from a connected unit with the <b>apgControlUnit SYNCGO</b> command; or from an external synchronisation source.<br><br>The timestamp is reset to ZERO or TIME with the <b>apgSetUnit TSRESETMODE</b> command.<br><br>If the timestamp reset mode is TIME, it is recommended that the <b>TSSYNCCLEAR</b> is immediately followed by a locally generated synchronisation pulse from the <b>apgControlUnit SYNCGO</b> command to synchronise the unit timestamp close to the host PC time.<br><br>The armed state can be cleared with the <b>apgControlUnit SYNCCLEARUP</b> command.<br><br>See Section 3.6.2 and the APG User Guide [1] for more information. |

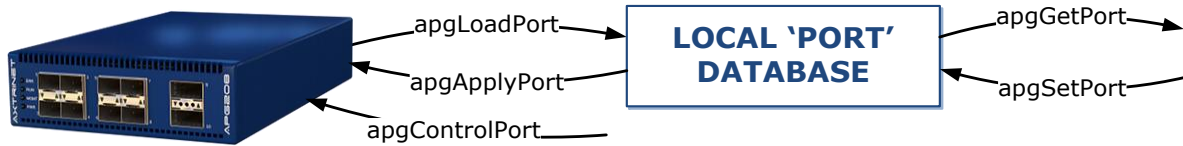


**A unit will stop responding to unit commands until the synchronise event (SYNCGO or external) occurs. During script development, this may leave the unit in a state where the script aborts with a "Unit Busy" error message. During script development, it is recommended that an apgControlUnit SYNCCLEARUP is used at the start of the script to clear any armed states.**

| [COMMAND]          | Description  |
|--------------------|--|
| <b>SYNCGO</b>      | <p>Generates a synchronisation pulse from [UNITID].</p> <p><b>SYNCGO</b> generates a single pulse on the rear SYNC interface to trigger the synchronous action on all connected units with primed ports.</p> <p>Synchronous transmission is armed with the <b>apgControlPort STARTSYNCTX</b> command.</p> <p>Synchronous timestamp reset is armed with the <b>apgControlUnit TSSYNCCLEAR</b> command.</p> <p>See Sections 3.6.2, 3.6.33.6.3 and the APG User Guide [1] for more information.</p> |
| <b>SYNCCLEARUP</b> | Clear the armed state.   |



## 8. PORT COMMANDS



|                       |  |
|-----------------------|--|
| <b>apgLoadPort</b>    | <b>apgLoadPort [PORTID] INFO</b><br><b>apgLoadPort [PORTID] MODULE</b><br><b>apgLoadPort [PORTID] CAPTURE</b><br><b>apgLoadPort [PORTID] ANALYSIS</b>  |
| <b>apgGetPort</b>     | <b>apgGetPort [PORTID] INFO [VAR]</b><br><b>apgGetPort [PORTID] STATUS [VAR]</b><br><b>apgGetPort [PORTID] MODULE [VAR]</b><br><b>apgGetPort [PORTID] TXSTATS [VAR]</b><br><b>apgGetPort [PORTID] RXSTATS [VAR]</b><br><b>apgGetPort [PORTID] RATES [VAR]</b><br><b>apgGetPort [PORTID] CAPTURE CONFIG [VAR]</b><br><b>apgGetPort [PORTID] CAPTURE PACKET TOTALPACKETS</b><br><b>apgGetPort [PORTID] CAPTURE PACKET [VAR] [PKTNUM]</b><br><b>apgGetPort [PORTID] ANALYSIS [TYPE] [VAR]</b> |
| <b>apgSetPort</b>     | <b>apgSetPort [PORTID] [VAR] [VAL]</b>   |
| <b>apgApplyPort</b>   | <b>apgApplyPort [PORTID] STATE</b><br><b>apgApplyPort [PORTID] COPPERMODULE</b>  |
| <b>apgControlPort</b> | <b>apgControlPort [COMMAND] [PORTLIST]</b><br><b>apgControlPort [CAPTURE] [PORTLIST]</b><br><b>apgControlPort CLEARANALYSIS [PORTLIST]</b>   |

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

```

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

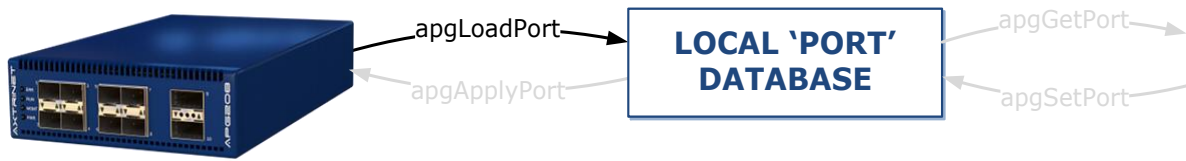
# Open Connection
set IPADDRESS 192.168.1.100
set UNITID [apgOpen $IPADDRESS]
set PORTLIST ""
Open Connection

# 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

# Configure Streams
puts "\nDEMO: Configure Streams"
set STREAM 0
foreach PORTID $PORTLIST {
    Configure Streams
}
    
```

|  |  |
|--|--|
| <pre> scan \$PORTID "%d %d %d" UN PO SP puts "      -&gt; Port \$PO" for { set ST 0 } { \$ST &lt; 8 } { incr ST } {   set STREAMID [linsert \$PORTID end \$ST]   if { \$ST == 0 } {     apgSetStream \$STREAMID DEFAULT     apgSetStream \$STREAMID CONFIG ENABLE 1     apgSetStream \$STREAMID CONFIG SIZE_MODE RANDOM     apgSetStream \$STREAMID CONFIG PACKET_SIZE 64     apgSetStream \$STREAMID CONFIG PACKET_SIZE_MAX 5000     apgSetStream \$STREAMID HEADER HEADER_LIST "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     if { \$PO &gt;= 9 } {       apgSetStream \$STREAMID PAYLOAD DATA_CONTROL INCREMENT       apgSetStream \$STREAMID PAYLOAD PAYLOAD_DATA "01 02 03 04 05 06 07 08"     }   } else {     apgSetStream \$STREAMID CONFIG ENABLE 0   }   apgApplyStream \$STREAMID } } </pre> | <p>Stream 0</p> <p>Enable</p> <p>Random Size</p> <p>64 -</p> <p>5000 bytes</p> <p>MACHEADER</p> <p>Random Data</p> <p>If 40G</p> <p>Increment</p> <p>Data</p> <p>Disable 1-7</p> <p>Apply Config</p> |
| <pre> # Port Transmit &amp; Capture Control puts "\nDEMO: Generating Packets for 3 seconds" apgControlPort STOPTH \$PORTLIST apgControlPort CLEARCOUNTERS \$PORTLIST apgControlPort STARTTX \$PORTLIST after 1000 puts "DEMO: Calculating data rates over 1000ms" apgLoadUnit \$UNITID RATES 1000 puts "DEMO: Clearing Capture Buffers and enabling Capture on all ports " apgControlPort PORTCAPTURE \$PORTLIST after 1000 puts "DEMO: Stopping Traffic" apgControlPort STOPTH \$PORTLIST </pre>  | <p>TX Control</p> <p>Clear counts</p> <p>Start TX</p> <p>Calculate</p> <p>Rates</p> <p>Enable Capture</p> <p>Stop TX</p>   |
| <pre> # Port Counters puts "\nDEMO: Display Port Counters" apgLoadUnit \$UNITID COUNTERS foreach PORTID \$PORTLIST {   scan \$PORTID "%d %d %d" UN PO SP   set TXPKT [apgGetPort \$PORTID TXSTATS GOOD_PACKETS]   set TXRATE [expr [apgGetPort \$PORTID RATES TXBITRATE] / 1000000.0]   set RXPKT [apgGetPort \$PORTID RXSTATS GOOD_PACKETS]   if { \$TXPKT &gt; 0 } {     puts [format "      Port %2d TX %7d at %8.2f Mbps, RX %7d packets" \       \$PO \$TXPKT \$TXRATE \$RXPKT]   } } </pre>  | <p>Counters</p> <p>Load</p> <p>Get counters</p>  |
| <pre> # Port Capture puts "\nDEMO: Display Captured Packets" foreach PORTID \$PORTLIST {   scan \$PORTID "%d %d %d" UN PO SP   apgLoadPort \$PORTID CAPTURE CONFIG   set AVAILABLE [apgGetPort \$PORTID CAPTURE CONFIG AVAILABLE ]   if { \$AVAILABLE &gt; 0 } {     apgLoadPort \$PORTID CAPTURE BUFFER     puts "\n      Port \$PO:\n"     set PKTCOUNT [ apgGetPort \$PORTID CAPTURE PACKET TOTALPACKETS ]     for { set PKT 1 } { \$PKT &lt;= \$PKTCOUNT } { incr PKT } {       set LENGTH [apgGetPort \$PORTID CAPTURE PACKET LENGTH \$PKT]       puts -nonewline [format "      %3d: %5d bytes -&gt; " \$PKT \$LENGTH]       set DATA [apgGetPort \$PORTID CAPTURE PACKET DATA \$PKT]       foreach BYTE \$DATA { puts -nonewline [format "%02X " \$BYTE] }       puts ""     }   } } </pre>   | <p>Capture</p> <p>Load config</p> <p>If data ...</p> <p>Load buffer</p> <p>Display data</p>  |
| <pre> # Close connection apgClose \$UNITID </pre>  | <p>Close</p> <p>connection</p>   |

## 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  |
|---------------|--|
| <b>CONFIG</b> | Load the port capture configuration into the local port database.  |
| <b>BUFFER</b> | Load the port capture buffer into the local port database<br>{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, <b>all</b> 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.

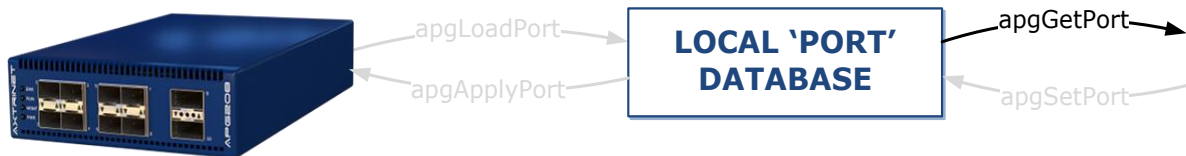
If {NUMPKTS} is excluded, all packets between {FIRSTPKT} and the end of the capture buffer are be analysed.

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  |
|---------------------|---------|--|
| <b>TOPOLOGY</b>     | 16 bits | <u>Current</u> port topology:<br>Bit 0-7 = Reserved<br>Bit 8 = Blocked (0x0100)<br>Bit 9 = 1Gbps (0x0200)<br>Bit 10 = Reserved<br>Bit 11 = 10Gbps (0x0800)<br>Bit 12 = 4x10Gbps (0x1000)<br>Bit 13 = 40Gbps (0x2000)<br>Bit 14-15 = Reserved |
| <b>TOPOLOGY_CAP</b> | 16 bits | Port topology capabilities:<br>Bit 0-7 = Reserved<br>Bit 8 = Blocked (0x0100)<br>Bit 9 = 1Gbps (0x0200)<br>Bit 10 = Reserved<br>Bit 11 = 10Gbps (0x0800)<br>Bit 12 = 4x10Gbps (0x1000)<br>Bit 13 = 40Gbps (0x2000)<br>Bit 14-15 = Reserved   |
| <b>NSUBPORTS</b>    | 8 bits  | Number of subports associated with the port  |
| <b>NSTREAMS</b>     | 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   |
|------------------------|---------|---|
| <b>LINK</b>            | 8 bits  | 0 = No Link<br>1 = Link OK  |
| <b>MODULE_STATUS</b>   | 8 bits  | Bit 0 = Not Present<br>Bit 1 = Module Present<br>Bit 2 = Module Fault<br>Bit 3 = LOS<br>Bit 4 = Transmit Enable<br>Bit 5 = Rate Select 0<br>Bit 6 = Rate Select 1<br>Bit 7 = Reserved   |
| <b>SPEED</b>           | 16 bits | Bit 0 = 10Mbps<br>Bit 1 = 100Mbps<br>Bit 2 = 1Gbps<br>Bit 3 = 10Gbps<br>Bit 4 = 40Gbps<br>Bit 5 = 100Gbps<br>Bit 6 = 4x10Gbps<br>Bit 7-15 = Reserved<br>Note that SPEED returns the expected port speed if the link is down, and the actual port speed if the link is up. |
| <b>SPEED_ABILITY</b>   | 16 bits | Bit 0 = 10Mbps<br>Bit 1 = 100Mbps<br>Bit 2 = 1Gbps<br>Bit 3 = 10Gbps<br>Bit 4 = 40Gbps<br>Bit 5 = 100Gbps<br>Bit 6 = 4x10Gbps<br>Bit 7-15 = Reserved  |
| <b>MODULE_SEQUENCE</b> | 16 bits | Module Sequence ID → Number of times a module has been inserted. Used to determine whether the module has changed.  |
| <b>MODULE_TYPE</b>     | 8 bits  | 0 = No Module<br>1 = SFP/SFP+<br>2 = QSFP+  |
| <b>MODULE_CLASS</b>    | 8 bits  | 0 = None<br>1 = Fibre<br>2 = Copper<br>3 = Passive Direct Attach<br>4 = Active Direct Attach  |
| <b>MODULE_TEMP</b>     | 16 bits | Module Temperature (°C)<br>Measured in 1/256ths of a degree. (ie divide by 256 to get measured temperature)   |

| [VAR]               | LENGTH  | DESCRIPTION   |
|---------------------|---------|---|
| <b>LINKMODE</b>     | 8 bits  | <p>Current MAC-to-Module PMA Link Mode:</p> <p>Bit 0 = 1000Base-X (SFP Port only)</p> <p>Bit 1 = SGMII (SFP+ Port only)</p> <p>Bits 3-4 = Reserved</p> <p>Bit 5 = SRLR (QSFP+ only)</p> <p>Note that the unit software will automatically choose the preferred link mode based on the transceiver, but can be over-written by the user.</p> |
| <b>LINKMODE_CAP</b> | 8 bits  | <p>MAC-to-Module PMA Link Mode Capabilities:</p> <p>Bit 0 = 1000Base-X (SFP+ Port only)</p> <p>Bit 1 = SGMII (SFP+ Port only)</p> <p>Bits 3-4 = Reserved</p> <p>Bit 5 = SRLR (QSFP+ only)</p>   |
| <b>TIMESTAMPID</b>  | 32 bits | Timestamp ID field  |
| <b>PORTNAME</b>     | 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   |
|--------------------|----------|---|
| <b>MODULE_TYPE</b> | 8 bits   | <p>0 = No Module</p> <p>1 = SFP/SFP+</p> <p>2 = QSFP+</p> |
| <b>ID</b>          | 96 chars | Serial ID Fields  |
| <b>VENDOR_ID</b>   | 32 chars | Vendor-Specific ID field                                  |

The following variables are extracted from the ID Fields:

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

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

| [VAR]        | LENGTH | DESCRIPTION   |
|--------------|--------|---|
| <b>VALID</b> | 8 bits | 0 = Copper Module Not Fitted<br>1 = Data Valid (Copper module fitted) |

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

| [VAR]        | LENGTH | DESCRIPTION   |
|--------------|--------|---|
| <b>LINK</b>  | 8 bits | 0 = No Link<br>1 = Port Link OK<br>2 = Port Module Link Ok (internal) |
| <b>FAULT</b> | 8 bits | 0 = OK<br>1 = Fault   |

#### Resolved settings:

|                        |        |   |
|------------------------|--------|---|
| <b>RESOLVED</b>        | 8 bits | 0 = Not Resolved (auto-negotiating)<br>1 = Resolved (auto-negotiation complete) |
| <b>RESOLVED_SPEED</b>  | 8 bits | 1 = 10Mbps<br>2 = 100Mbps<br>4 = 1Gbps<br>8 = 10Gbps                            |
| <b>RESOLVED_DUPLEX</b> | 8 bits | 1 = Half Duplex<br>2 = Full Duplex  |
| <b>RESOLVED_MDI</b>    | 8 bits | 2 = MDI<br>4 = MDIX   |
| <b>RESOLVED_MS</b>     | 8 bits | 4 = Master<br>8 = Slave   |
| <b>PAIRSWAP</b>        | 8 bits | 0 = Not Swapped<br>1 = Swapped  |

#### Configuration Settings:

|                            |        |   |
|----------------------------|--------|---|
| <b>CONFIG_SPEED_DUPLEX</b> | 8 bits | 1 = Auto-Negotiation Enabled<br>2 = Fixed Mode, Auto-Negotiation Disabled<br>Note: If 0, ignore configuration   |
| <b>CONFIG_FORCED_SPEED</b> | 8 bits | 1 = 10Mbps Half Duplex<br>2 = 10Mbps Full Duplex<br>4 = 100Mbps Half Duplex<br>8 = 100Mbps Full Duplex<br>16 = 1000Mbps Half Duplex<br>32 = 1000Mbps Full Duplex<br>Note: If 0, forced speeds not set |



| [VAR]                   | LENGTH  | DESCRIPTION  |
|-------------------------|---------|--|
| <b>CONFIG_ADV_SPEED</b> | 16 bits | Bit 0 = Advertise Valid (otherwise settings ignored)<br>Bit 1 = Advertise 10Mbps Half Duplex<br>Bit 2 = Advertise 10Mbps Full Duplex<br>Bit 3 = Advertise 100Mbps Half Duplex<br>Bit 4 = Advertise 100Mbps Full Duplex<br>Bit 5 = Advertise 1000Mbps Half Duplex<br>Bit 6 = Advertise 1000Mbps Full Duplex<br>Bit 7 = Advertise 10Gbps Full Duplex<br>Bit 8 = Advertise 2.5Gbps Full Duplex<br>Bit 9 = Advertise 5Gbps Full Duplex<br>Note: Advertised capabilities only valid if Bit 0 is set |
| <b>CONFIG_ADV_PAUSE</b> | 8 bits  | Bit 0 = Not used<br>Bit 1 = Advertise Symmetric Pause<br>Bit 2 = Advertise Asymmetric Pause  |
| <b>CONFIG_MDI</b>       | 8 bits  | 0 = Not Known<br>1 = Automatic<br>2 = Force MDI<br>4 = Force MDIX (Crossover)  |
| <b>CONFIG_MS</b>        | 8 bits  | 0 = Ignore<br>1 = Automatic, prefer MASTER<br>2 = Automatic, prefer SLAVE<br>4 = Force MASTER<br>8 = Force SLAVE   |
| <b>Capabilities:</b>    |         |  |
| <b>CAP_SPEED_DUPLEX</b> | 8 bits  | Bit 0 = Auto-Negotiation Mode Available<br>Bit 1 = Fixed Mode Available<br>Note: If 0, ignore capabilities.  |
| <b>CAP_FORCED_SPEED</b> | 8 bits  | Bit 0 = Capable of Fixed 10Mbps Half Duplex<br>Bit 1 = Capable of Fixed 10Mbps Full Duplex<br>Bit 2 = Capable of Fixed 100Mbps Half Duplex<br>Bit 3 = Capable of Fixed 100Mbps Full Duplex<br>Bit 4 = Capable of Fixed 1000Mbps Half Duplex<br>Bit 5 = Capable of Fixed 1000Mbps Full Duplex   |
| <b>CAP_ADV_SPEED</b>    | 16 bits | Bit 0 = Valid (1)<br>Bit 1 = 10Mbps Half Duplex<br>Bit 2 = 10Mbps Full Duplex<br>Bit 3 = 100Mbps Half Duplex<br>Bit 4 = 100Mbps Full Duplex<br>Bit 5 = 1000Mbps Half Duplex<br>Bit 6 = 1000Mbps Full Duplex<br>Bit 7 = 10Gbps Full Duplex<br>Bit 8 = 2.5Gbps Full Duplex<br>Bit 9 = 5Gbps Full Duplex  |
| <b>CAP_ADV_PAUSE</b>    | 8 bits  | Bit 0 = Not used<br>Bit 1 = Advertise Symmetric Pause<br>Bit 2 = Advertise Asymmetric Pause  |
| <b>CAP_MDI</b>          | 8 bits  | 0 = Not Known<br>1 = Automatic<br>2 = Force MDI<br>4 = Force MDIX (Crossover)  |

| [VAR]                      | LENGTH  | DESCRIPTION  |
|----------------------------|---------|--|
| <b>CAP_MS</b>              | 8 bits  | Bit 0 = Ignore<br>Bit 1 = Automatic, prefer MASTER<br>Bit 2 = Automatic, prefer SLAVE<br>Bit 3 = Force MASTER<br>Bit 4 = Force SLAVE   |
| <b>Remote Transceiver:</b> |         |  |
| <b>REMOTE_ADV_SPEED</b>    | 16 bits | Bit 1 = Advertising 10Mbps Half Duplex<br>Bit 2 = Advertising 10Mbps Full Duplex<br>Bit 3 = Advertising 100Mbps Half Duplex<br>Bit 4 = Advertising 100Mbps Full Duplex<br>Bit 5 = Advertising 1000Mbps Half Duplex<br>Bit 6 = Advertising 1000Mbps Full Duplex<br>Bit 7 = Advertising 10Gbps Full Duplex<br>Bit 8 = Advertising 2.5Gbps Full Duplex<br>Bit 9 = Advertising 5Gbps Full Duplex |
| <b>REMOTE_ADV_PAUSE</b>    | 8 bits  | Bit 0 = Not used<br>Bit 1 = Advertising Symmetric Pause<br>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
set IPADDRESS 192.168.1.100
set UNITID [apgOpen $IPADDRESS]
set PORTLIST {{ $UNITID 1 0} { $UNITID 2 0}}

# Port Counters
puts "\nDEMO: Display Port Counters"
apgLoadUnit $UNITID COUNTERS
foreach PORTID $PORTLIST {
  scan $PORTID "%d %d %d" UN PO SP
  set TXPKT [apgGetPort $PORTID TXSTATS GOOD_PACKETS]
  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
apgClose $UNITID
```

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   |
|----------------------|---------|---|
| <b>TIMESTAMP</b>     | 32 bits | Time that the transmit counters are read            |
| <b>BYTES</b>         | 64 bits | Number of transmitted bytes                         |
| <b>GOOD_PACKETS</b>  | 64 bits | Number of good packets transmitted                  |
| <b>PKT_64</b>        | 32 bits | Count of 64 byte packets transmitted                |
| <b>PKT_65_128</b>    | 32 bits | Count of 65-128 byte packets transmitted            |
| <b>PKT_129_256</b>   | 32 bits | Count of 129-256 byte packets transmitted           |
| <b>PKT_257_512</b>   | 32 bits | Count of 257-512 byte packets transmitted           |
| <b>PKT_513_1024</b>  | 32 bits | Count of 513-1024 byte packets transmitted          |
| <b>PKT_1025_1536</b> | 32 bits | Count of 1025-1536 byte packets transmitted         |
| <b>PKT_1537_9000</b> | 32 bits | Count of 1537-9000 byte packets transmitted         |
| <b>PKT_9001_MAX</b>  | 32 bits | Count of >9001 byte packets transmitted             |
| <b>TX_RUNTIME</b>    | 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   |
|----------------------|---------|---|
| <b>TIMESTAMP</b>     | 32 bits | Time that the receive counters are read                                   |
| <b>BYTES</b>         | 64 bits | Number of received bytes  |
| <b>GOOD_PACKETS</b>  | 64 bits | Number of good packets received   |
| <b>PKT_UNDERSIZE</b> | 32 bits | Number of under-sized (<64 Bytes) packets received                        |
| <b>PKT_FRAGMENTS</b> | 32 bits | Number of packets fragments received                                      |
| <b>PKT_FCSERROR</b>  | 32 bits | Number of packets received with Frame Checksum (FCS) Errors               |
| <b>PKT_NOSFD</b>     | 32 bits | Number of packets received without a valid Start-of-Frame Delimiter (SFD) |
| <b>PKT_64</b>        | 32 bits | Count of 64 byte packets received   |
| <b>PKT_65_128</b>    | 32 bits | Count of 65-128 byte packets received                                     |
| <b>PKT_129_256</b>   | 32 bits | Count of 129-256 byte packets received                                    |
| <b>PKT_257_512</b>   | 32 bits | Count of 257-512 byte packets received                                    |
| <b>PKT_513_1024</b>  | 32 bits | Count of 513-1024 byte packets received                                   |
| <b>PKT_1025_1536</b> | 32 bits | Count of 1025-1536 byte packets received                                  |
| <b>PKT_1537_9000</b> | 32 bits | Count of 1537-9000 byte packets received                                  |
| <b>PKT_9001_MAX</b>  | 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   |
|--------------------|---------|---|
| <b>TXPKTRATE</b>   | 32 bits | Transmit Packet Rate (packets/sec)  |
| <b>TXBYTERATE</b>  | 64 bits | Transmit Byte Rate (Bytes/sec)<br>Derived from the Transmit BYTE count  |
| <b>TXBITRATE</b>   | 64 bits | Transmit Bit Rate (Bits/sec)<br>Bits on the wire, including Inter-Frame Gap (IFG) and Start-of-Frame (SFD) delimiter.           |
| <b>RXPKTRATE</b>   | 32 bits | Receive Packet Rate (packets/sec)   |
| <b>RXERRORRATE</b> | 32 bits | Receive FCS Errored Packet Rate (packets/sec).<br>Divide the returned value by 16384 to get the FCS errored packets per second. |
| <b>RXBYTERATE</b>  | 64 bits | Receive Byte Rate (Bytes/sec)<br>Derived from the Receive BYTE count  |
| <b>RXBITRATE</b>   | 64 bits | Receive Bit Rate (Bits/sec)<br>Bits on the wire, including Inter-Frame Gap (IFG) and Start-of-Frame (SFD) delimiter.            |

## 8.2.7 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)<br>source "C:/Program Files/Axtrinet/APG/axtrinetApi.tcl"  |  |
| # Open Connection<br>set IPADDRESS 192.168.1.100<br>set UNITID [apgOpen \$IPADDRESS]<br>set PORTLIST {{ \$UNITID 1 0 } { \$UNITID 2 0 } }  | Open Connection  |
| # Capture Control<br>puts "\nDEMO: Generating Packets"<br>apgControlPort STOPTHX \$PORTLIST<br>apgControlPort STARTTX \$PORTLIST<br>puts "DEMO: Clearing Capture Buffers and enabling Capture on all ports "<br>apgControlPort PORTCAPTURE \$PORTLIST<br>after 100<br>puts "DEMO: Stopping Traffic"<br>apgControlPort STOPTHX \$PORTLIST   | TX Control<br><br>Start TX<br><br>Enable Capture<br><br>Stop TX                  |
| # Port Capture<br>puts "\nDEMO: Display Captured Packets"<br>foreach PORTID \$PORTLIST {<br>scan \$PORTID "%d %d %d" UN PO SP<br>apgLoadPort \$PORTID CAPTURE CONFIG<br>set AVAILABLE [apgGetPort \$PORTID CAPTURE CONFIG AVAILABLE ]<br>if { \$AVAILABLE > 0 } {<br>apgLoadPort \$PORTID CAPTURE BUFFER<br>puts "\n  Port \$PO:\n"<br>set PKTCOUNT [ apgGetPort \$PORTID CAPTURE PACKET TOTALPACKETS ]<br>for { set PKT 1 } { \$PKT <= \$PKTCOUNT } { incr PKT } {<br>set LENGTH [apgGetPort \$PORTID CAPTURE PACKET LENGTH \$PKT]<br>puts -nonewline [format "  %3d: %5d bytes -> " \$PKT \$LENGTH]<br>set DATA [apgGetPort \$PORTID CAPTURE PACKET DATA \$PKT]<br>foreach BYTE \$DATA { puts -nonewline [format "%02X " \$BYTE] }<br>puts ""<br>}<br>}<br>} | Capture<br><br>Load config<br><br>If data ...<br>Load buffer<br><br>Display data |
| # Close connection<br>apgClose \$UNITID  | Close 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   |
|-------------------|---------|---|
| <b>AVAILABLE</b>  | 32 bits | Filled capture data available for upload (bytes)  |
| <b>BUFFERSIZE</b> | 32 bits | Total memory available for packet capture (bytes)   |
| <b>FLAGS</b>      | 2 bits  | 0 = Capture Disabled<br>1 = Local Port Buffer Enabled & Capturing Packets<br>2 = Capture Disabled<br>3 = Extended Packet Buffer Enabled & Capturing Packets |



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**.

### 8.2.8 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**.

### 8.2.9 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  |
|-----------------|----------|--|
| <b>RXTIME</b>   | 64 bits  | Receive timestamp of packet [PKTNUM] measured in 8ns intervals since the unit was turned on.                   |
| <b>TSMODE</b>   | 4 bits   | Receive Timestamp Flag of packet [PKTNUM]<br>0xA = Approximated timestamp<br>Anything else = Actual timestamp. |
| <b>LENGTH</b>   | 16 bits  | Length of packet [PKTNUM]  |
| <b>PKTERROR</b> | 1 bit    | Packet Error Flag of packet [PKTNUM]:<br>0 = OK<br>1 = FCS Error   |
| <b>DATA</b>     | [String] | Captured data of packet [PKTNUM]<br>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   |
|----------------|---------|---|
| <b>IDFIELD</b> | 32 bits | ID Field extracted from packet [PKTNUM]<br>Bytes [LENGTH]-19 to [LENGTH]-16   |
| <b>SEQNUM</b>  | 32 bits | Sequence Number extracted from packet [PKTNUM]<br>Bytes [LENGTH]-15 to [LENGTH]-12  |
| <b>TXTIME</b>  | 64 bits | Transmit timestamp extracted from packet [PKTNUM]<br>Bytes [LENGTH]-11 to [LENGTH]-4, measured in 8ns intervals since the unit was turned on. |
| <b>DELTA</b>   | 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.

### 8.2.10 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                                 |
|-----------------|---|
| <b>TXTX</b>     | Transmit-to-transmit timestamp calculations |
| <b>TXRX</b>     | Transmit-to-receive timestamp calculations  |
| <b>RXRX</b>     | Receive-to-receive timestamp calculations   |
| <b>SEQUENCE</b> | Sequence Number status                      |

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

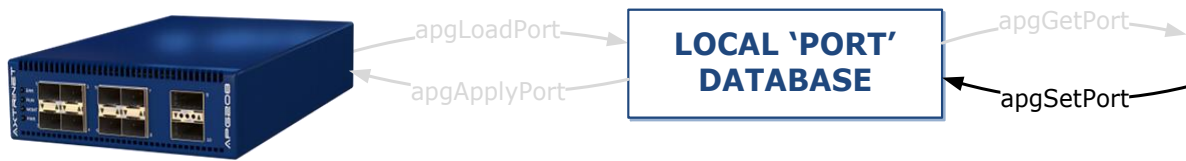
| [VAR]          | DESCRIPTION                                 |
|----------------|---|
| <b>CYCLES</b>  | Number of capture cycles to collect samples |
| <b>SAMPLES</b> | Number of processed latency samples         |
| <b>LIST</b>    | List of processed timing deltas             |
| <b>MIN</b>     |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

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

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



### 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 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                                       |
|-----------------|--|---|
| <b>TXENABLE</b> | 0 = Module Disabled<br>1 = Module Enabled  | 0x1   |
| <b>RATE</b>     | [RATE SELECT 1][RATE SELECT 0]<br>0 = RS1 & RS0 Not Set (low speed operation)<br>1 = RS1 Not Set, RS0 Set<br>2 = RS1 Set, RS0 Not Set<br>3 = RS1 & RS0 Set (high speed operation)<br>Note: Both bits should normally be set high for 10Gbps operation, although some vendor's transceivers do not meet the SFP MSA guidelines. Check vendor's recommended settings in the datasheet. | 0x3   |
| <b>TOPOLOGY</b> | BLOCKED = Port Disabled<br><b>SFP Ports only:</b><br>1G = Fixed 1Gbps<br><b>SFP+ Ports only:</b><br>10G = Fixed 10Gbps<br><b>QSFP+ Ports only:</b><br>40G = Fixed 40Gbps<br>4x10G = Fixed 4x10Gbps   | SFP+ Ports:<br>10G<br><br>QSFP+ Ports:<br>40G |
| <b>LINKMODE</b> | <b>SFP Ports only:</b><br>1000X = 1000Base-X (1Gbps only)<br><b>SFP+ Ports only:</b><br>SGMII = SGMII (10Gbps or 1Gbps)<br><b>QSFP+ Ports only:</b><br>SRLR = 40GBase-SR/LR  | SFP+ Ports:<br>SGMII<br>QSFP+ Ports:<br>SRLR  |

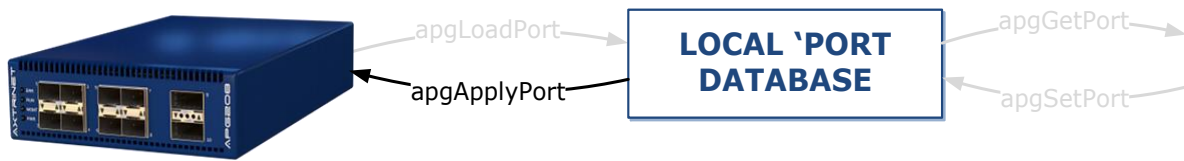
### 8.3.1.2 Copper Module

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

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

| [VAR]              | Description   | Default   |
|--------------------|---|---|
| <b>AUTONEG</b>     | <b>Auto-Negotiation/Fixed Mode</b><br>0 = Ignore<br>1 = Auto-Negotiate Speed/Duplex<br>2 = Fixed Speed/Duplex   | 1 (auto)  |
| <b>FORCE_SPEED</b> | <b>Force Speed/Duplex</b><br>1 = Force 10Mbps, Half Duplex<br>2 = Force 10Mbps, Full Duplex<br>4 = Force 100Mbps, Half Duplex<br>8 = Force 100Mbps, Full Duplex<br>16 = Force 1000Mbps, Half Duplex<br>32 = Force 1000Mbps, Full Duplex<br>Note: If 0, ignore settings.   | 0x00 (ignore)   |
| <b>ADV_SPEED</b>   | <b>Advertise Speed/Duplex</b><br>Bit 0 = Advertise Valid<br>Bit 1 = Advertise 10Mbps, Half Duplex<br>Bit 2 = Advertise 10Mbps, Full Duplex<br>Bit 3 = Advertise 100Mbps, Half Duplex<br>Bit 4 = Advertise 100Mbps, Full Duplex<br>Bit 5 = Advertise 1000Mbps, Half Duplex<br>Bit 6 = Advertise 1000Mbps, Full Duplex<br>Bit 7 = Advertise 10Gbps, Full Duplex<br>Note: Advertise Valid bit must be set otherwise the advertised capabilities are ignored. | 0x55 (1Gbps)<br>[10/100/1000M]<br><br>0xC1 (10Gbps)<br>[10G/1G] |
| <b>ADV_PAUSE</b>   | <b>Advertise Pause Frames</b><br>Bit 0 = Not Used<br>Bit 1 = Advertise Symmetrical Pause<br>Bit 2 = Advertise Asymmetrical Pause  | 0 (disabled)  |
| <b>MDI</b>         | <b>MDI Crossover</b><br>1 = Automatic<br>2 = MDI<br>4 = MDIX (Crossover)  | 1 (auto)  |
| <b>MS</b>          | <b>MASTER/SLAVE Mode: [1000Base-T only]</b><br>1 = Automatic, prefer MASTER<br>2 = Automatic, prefer SLAVE<br>4 = Force MASTER<br>8 = Force SLAVE   | 1 (auto, master)  |

## 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.

## 8.5 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   |
|---------------------|---|
| <b>STOPPED</b>      | <p>Traffic generation stopped.</p> <p>Stream configuration is loaded into the port traffic generator when a <b>STARTTX</b> or <b>STEPTX</b> is issued.</p> <p>Port transmit counters are stopped, although the port receive counters may still be counting if the port is receiving traffic.</p>  |
| <b>ARMED</b>        | <p>The port is armed and remains in this state until a <b>apgControlUnit SYNCGO</b> or <b>TSSYNCCLEAR</b> command is issued to synchronously start transmission on all armed ports; or a <b>SYNCCLEARUP</b> command is issued to disarm the port.</p> <p>See Section 3.6 for more information.</p>  |
| <b>TRANSMITTING</b> | <p>The port is generating traffic, and remains in this state until a <b>STOPTX</b>, <b>PAUSETX</b> or <b>STEPTX</b> command is issued.</p> <div style="border: 1px solid #ccc; background-color: #f0f8ff; padding: 10px; margin: 10px 0;"> <p> If all of the enabled streams are generating fixed burst length traffic, the port will stop transmitting when all streams have transmitted the <b>BURST LENGTH</b> number of packets and move to the <b>STOPPED</b> state.</p> </div> <p>A <b>STEPTX</b> command transmits the next packet, then PAUSES. Issuing a second <b>STARTTX</b> has no effect.</p> |
| <b>PAUSED</b>       | <p>Traffic generation is stopped.</p> <p>Transmission continues when a <b>STARTTX</b> or <b>STEPTX</b> command is issued, without loading a new stream configuration.</p> <div style="border: 1px solid #ccc; background-color: #f0f8ff; padding: 10px; margin: 10px 0;"> <p> The PAUSE state is useful for interrupting a continuous stream to allow the test data to settle and read the counters.</p> </div>  |
| <b>STEP</b>         | <p>Transmit the next packet from the stream queue. The port immediately returns to the PAUSED state.</p>  |

```

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 {{ $UNITID 1 0 } { $UNITID 2 0 }}
Open Connection

# Port Transmit & Capture Control
puts "\nDEMO: Generating Packets for 3 seconds"
apgControlPort STOPTX $PORTLIST
apgControlPort CLEARCOUNTERS $PORTLIST
apgControlPort STARTTX $PORTLIST
TX Control
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 "
Enable Capture
apgControlPort PORTCAPTURE $PORTLIST
after 1000
puts "DEMO: Stopping Traffic"
Stop TX
apgControlPort STOPTX $PORTLIST

# Close connection
apgClose $UNITID
Close 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   |
|----------------------|---|
| <b>STARTTX</b>       | Start packet transmission on all [PORTLIST] ports.<br>From the STOPPED state, <b>STARTTX</b> loads the new stream configuration on all [PORTLIST] ports and start transmitting.<br>From the PAUSED state, packet transmission will start without loading the latest configuration.            |
| <b>STEPTX</b>        | Transmit the next packet on all [PORTLIST] ports.<br>From the STOPPED state, <b>STEPTX</b> loads the new stream configuration on all [PORTLIST] ports and transmits a single packet.<br>Where multiple streams are enabled, the port will transmit the next packet from the stream sequencer. |
| <b>PAUSETX</b>       | Pause the packet transmission on all [PORTLIST] ports   |
| <b>STOPTX</b>        | Stop the packet transmission on all [PORTLIST] ports  |
| <b>CLEARCOUNTERS</b> | Clear the transmit and receive packet counters on all [PORTLIST] ports  |

### 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  |
|--------------------|--|
| <b>STARTSYNCTX</b> | Arm synchronous packet transmission on all [PORTLIST] ports.<br>From the STOPPED state, <b>STARTSYNCTX</b> loads the new stream configuration on all [PORTLIST] ports but does not start transmitting.<br>Synchronous transmission is started on the rising edge of a pulse on the SYNC interface, either from a connected unit with the <b>apgControlUnit SYNCGO</b> command; or from an external synchronisation source.<br>The armed state can be cleared with the <b>apgControlUnit SYNCCLEARUP</b> 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 3.6.3 and the APG User Guide [1] for more information.

### 8.5.3 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   |
|-----------------------|---|
| <b>PORTCAPTURE</b>    | <p>Enable the local capture buffers on all [PORTLIST] ports and start capturing packets.</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;">  Enabling the local capture from a deep capture will clear the deep capture buffer.                 </div>   |
| <b>DEEPCAPTURE</b>    | <p>Enable the extended capture buffer on all [PORTLIST] ports and start capturing packets.</p> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;">  Enabling the deep capture from a local capture will clear the local capture buffer.                 </div>  |
| <b>DISABLECAPTURE</b> | <p>Disable packet capture on all [PORTLIST] ports. The capture buffer is not cleared.<br/>The selected capture buffer is still enabled, and can be read using <b>apgLoadPort CAPTURE BUFFER</b>.<br/>Capture can be restarted with <b>apgControlPort PORTCAPTURE</b> (local) or <b>apgControlPort DEEPCAPTURE</b> (deep). If the capture buffer has not been cleared, the buffer will contain the already-captured data after restarting.</p> |
| <b>CLEARCAPTURE</b>   | <p>Clear the capture buffers on all [PORTLIST] ports</p>  |

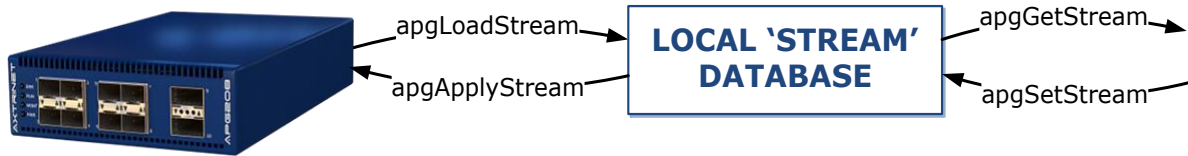
For example, to capture and display active traffic on connected ports 1 & 5:

|     |  |   |
|-----|--|---|
| Eg: | <pre>set PORTLIST {{1 2}{1 5}} apgControlPort PORTCAPTURE \$PORTLIST  apgControlPort CLEARCAPTURE \$PORTLIST apgControlPort STARTTX \$PORTLIST  foreach PORTID \$PORTLIST {apgLoadPort \$PORTID CAPTURE BUFFER 100}  foreach SUBPORTID \$PORTLIST {   set PKTCOUNT [ apgGetPort \$SUBPORTID CAPTURE PACKET TOTALPACKETS ]    for { set PKT 1 } { \$PKT &lt;= \$PKTCOUNT } { incr PKT } {     puts [apgGetPort \$SUBPORTID CAPTURE PACKET DATA \$PKT]   } }</pre> | <p>Enable local port capture on both ports.</p> <p>Clear capture buffers and start transmitting</p> <p>Load 100 packets from each port buffer</p> <p>For each port, get the number of downloaded packets</p> <p>Then display each packet.</p> |
|-----|--|---|

### 8.5.4 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]**

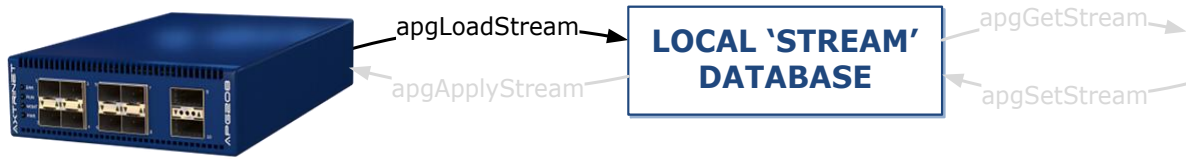
**apgApplyStream**    **apgApplyStream [STREAMID]**

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

```

Eg: # Instantiate APG TCL API (Windows)
     source "C:/Program Files/Axtrinet/APG/axtrinetApi.tcl"
     # Open Connection
     set IPADDRESS 192.168.1.100
     set UNITID [apgOpen $IPADDRESS]
     set PORTLIST {{ $UNITID 1 0 } { $UNITID 2 0 }}
     # Configure Streams
     puts "\nDEMO: Configure 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 } {
                 apgSetStream $STREAMID DEFAULT
                 apgSetStream $STREAMID CONFIG ENABLE 1
                 apgSetStream $STREAMID CONFIG SIZE_MODE RANDOM
                 apgSetStream $STREAMID CONFIG PACKET_SIZE 64
                 apgSetStream $STREAMID CONFIG PACKET_SIZE_MAX 5000
                 apgSetStream $STREAMID HEADER HEADER_LIST "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
                 if { $PO >= 9 } {
                     apgSetStream $STREAMID PAYLOAD DATA_CONTROL INCREMENT
                     apgSetStream $STREAMID PAYLOAD PAYLOAD_DATA "01 02 03 04 05 06 07 08"
                 }
             } else {
                 apgSetStream $STREAMID CONFIG ENABLE 0
             }
             apgApplyStream $STREAMID
         }
     }
     # Close connection
     apgClose $UNITID
    
```

## 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.

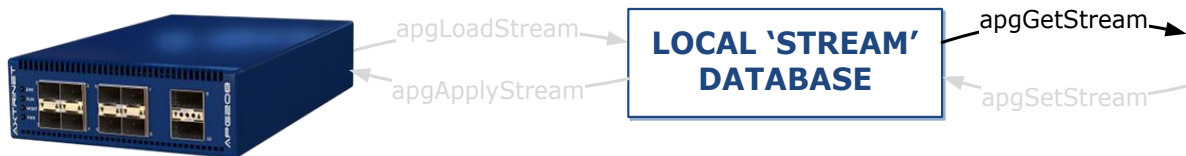
### 9.1.1 apgLoadStream [STREAMID]

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:

|                |  |
|----------------|--|
| <b>CONFIG</b>  | Defines the stream state, packet rate and packet size          |
| <b>HEADER</b>  | Defines the Ethernet headers (eg MACHEADER) and fields (eg DA) |
| <b>PAYLOAD</b> | 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  |
|------------------------|---------|--|
| <b>ENABLE</b>          | 8 bits  | 0 = Stream Disabled<br>1 = Stream Enabled  |
| <b>TX_MODE</b>         | 8 bits  | 0 = Continuous<br>1 = Burst  |
| <b>RATE_MODE</b>       | 8 bits  | 0 = TXGAP (Inter Frame Gap)<br>1 = PPS (Frames Per Second)<br>2 = Percent  |
| <b>SIZE_MODE</b>       | 8 bits  | 0 = Fixed<br>3 = Random  |
| <b>TX_CONTROL</b>      | 8 bits  | 0 = Transmit good frames   |
| <b>RATE_VALUE</b>      | 64 bits | If RATE_MODE = 0 (TXGAP)<br>→ TXGAP = RATE_VALUE * 8 bytes * link speed<br><br>If RATE_MODE = 1 (Packets per Second)<br>→ PPS = RATE_VALUE pps<br><br>If RATE_MODE = 2 (PERCENT)<br>→ PERCENT = RATE_VALUE * 1000<br>Eg RATE_VALUE = 90500 → 90.5% |
| <b>TX_BURST_SIZE</b>   | 32 bits | Number of frames in the burst  |
| <b>TX_BURST_COUNT</b>  | 32 bits | Number of bursts to transmit   |
| <b>TX_IBG</b>          | 32 bits | Inter-burst gap (TX_IBG x 5.5ns)   |
| <b>PACKET_SIZE</b>     | 32 bits | Total frame length (including FCS) in bytes.<br>If SIZE_MODE is not 0 (FIXED), FRAME_SIZE defines the minimum frame length for the increment, decrement or random size.  |
| <b>PACKET_SIZE_MAX</b> | 32 bits | Maximum frame length (including FCS) in bytes for the increment, decrement or random size modes.<br>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: <table border="0" style="margin-left: 20px;"> <tr> <td><b>FIX</b></td> <td>Fixed [FLD] value [VAL]</td> </tr> <tr> <td><b>INC</b></td> <td>Incrementing [FLD] value [VAL], starting at [VAL] with each byte incrementing to [MAX], wrapping to [MIN] to continue incrementing</td> </tr> </table> | <b>FIX</b> | Fixed [FLD] value [VAL] | <b>INC</b> | Incrementing [FLD] value [VAL], starting at [VAL] with each byte incrementing to [MAX], wrapping to [MIN] to continue incrementing |
| <b>FIX</b> | Fixed [FLD] value [VAL]  |            |                         |            |  |
| <b>INC</b> | 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  |
|--------------------|-------------------|-------------------|---|
| <b>MAC-ADDRESS</b> | 00:00:00:00:00:00 | FF:FF:FF:FF:FF:FF |   |
| <b>HEX4</b>        | 0x0               | 0xFFFF            |   |
| <b>HEX2ARRAY</b>   | 0x0               | 0xFF              | Variable length HEX2 values separated by spaces |

### 9.2.3.1 MACHEADER

| [FLD]     | DESC                    | SIZE    | FORMAT      | MODE    |
|-----------|-------------------------|---------|-------------|---------|
| <b>DA</b> | MAC Destination Address | 6 bytes | MAC-ADDRESS | FIX INC |
| <b>SA</b> | MAC Source Address      | 6 bytes | MAC-ADDRESS | FIX INC |

### 9.2.3.2 ETHERNET\_II

| [FLD]            | DESC             | SIZE    | FORMAT | MODE    |
|------------------|------------------|---------|--------|---------|
| <b>ETHERTYPE</b> | 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 |
|-------------|-------------|----------|-----------|------|
| <b>DATA</b> | Data Values | Variable | HEX2ARRAY | --   |

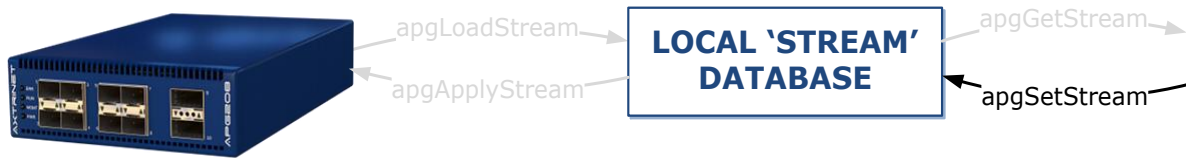
## 9.2.4 apgGetStream [STREAMID] PAYLOAD [VAR]

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                         |
|---------------------|---------|-------------------------------------|
| <b>PAYLOAD_DATA</b> | 12 Char | Payload values                      |
| <b>DATA_CONTROL</b> | 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"

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

# Configure Streams
puts "\nDEMO: Configure 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 } {
            apgSetStream $STREAMID DEFAULT
            apgSetStream $STREAMID CONFIG ENABLE 1
            apgSetStream $STREAMID CONFIG SIZE_RANDOM
            apgSetStream $STREAMID CONFIG PACKET_SIZE 64
            apgSetStream $STREAMID CONFIG PACKET_SIZE_MAX 5000
            apgSetStream $STREAMID HEADER HEADER_LIST "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
            if { $PO >= 9 } {
                apgSetStream $STREAMID PAYLOAD DATA_CONTROL INCREMENT
                apgSetStream $STREAMID PAYLOAD PAYLOAD_DATA "01 02 03 04 05 06 07 08"
            }
        } else {
            apgSetStream $STREAMID CONFIG ENABLE 0
        }
        apgApplyStream $STREAMID
    }
}

# Close connection
apgClose $UNITID
    
```

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

| [VAR]                  | Description   | Default | FORMAT |
|------------------------|---|---------|--------|
| <b>PACKET_SIZE</b>     | <p>Total packet length (including FCS) in bytes.<br/>If SIZE_MODE is not 0 (FIXED), FRAME_SIZE defined the minimum frame length for the increment, decrement or random size.<br/><math>32 \leq \text{PACKET\_SIZE} \leq 16000</math> bytes<br/>If <math>\text{PACKET\_SIZE} &lt; 32 \rightarrow 32</math><br/>If <math>\text{PACKET\_SIZE} &gt; 16000 \rightarrow 16000</math><br/>Packet sizes <math>&lt; 64</math> bytes will be counted as UNDERSIZE packets.</p> <div style="border: 1px solid orange; padding: 5px; background-color: #fff9e6;">  <b>At 10Mbps, the packet size must be less than 1600 bytes</b> </div> | 100     | INT    |
| <b>PACKET_SIZE_MAX</b> | <p>Maximum packet length (including FCS) in bytes.<br/><br/>If SIZE_MODE is 0 (FIXED) this value is ignored.<br/><br/><b>PACKET_SIZE_MAX</b> <math>\leq 16000</math> bytes<br/><br/>Larger values will be set to 16000.</p>   | 100     | INT    |

Where variable FORMAT has the follow settings:

| FORMAT      | Min | Max        |                     |
|-------------|-----|------------|---------------------|
| <b>INT</b>  | 0   | N/A        |                     |
| <b>HEX2</b> | 0x0 | 0xFF       | (256 dec)           |
| <b>HEX4</b> | 0x0 | 0xFFFF     | (65,535 dec)        |
| <b>HEX8</b> | 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: <table border="0" style="margin-left: 20px;"> <tr> <td><b>FIX</b></td> <td>Fixed [FLD] value [VAL]</td> </tr> <tr> <td><b>INC</b></td> <td>Incrementing [FLD] value [VAL], starting at [VAL] with each byte incrementing to [MAX], wrapping to [MIN] to continue incrementing</td> </tr> </table> | <b>FIX</b> | Fixed [FLD] value [VAL] | <b>INC</b> | Incrementing [FLD] value [VAL], starting at [VAL] with each byte incrementing to [MAX], wrapping to [MIN] to continue incrementing |
| <b>FIX</b> | Fixed [FLD] value [VAL]  |            |                         |            |  |
| <b>INC</b> | 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 <div style="background-color: #ffe4c4; padding: 5px; margin-top: 5px;">  <b>Always zero in APG TCL API APG V1.3.1 Software.</b> </div> [FLD] value wraps to [MAX] in decrementing mode.   |            |                         |            |  |
| [MAX]      | Maximum range for INC .<br>[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.

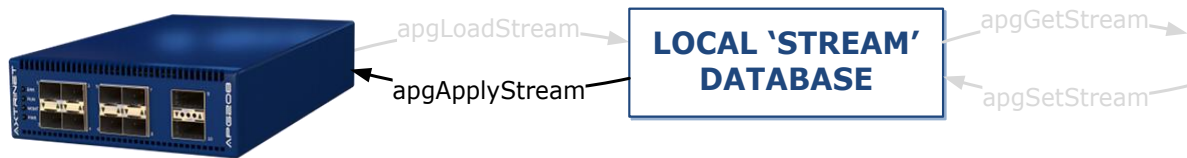
| [VAR]               | Description                                      | Default | RANGE     |
|---------------------|--|---------|-----------|
| <b>DATA_CONTROL</b> | FIXED, RANDOM, INCREMENT, DECREMENT              | FIXED   |           |
| <b>PAYLOAD_DATA</b> | Payload values                                   | 00      | HEX2ARRAY |
| <b>TS_ENABLE</b>    | Timestamp Control<br>0 = Disabled<br>1 = Enabled | 0       | INT       |

Where variable FORMAT has the follow settings:

| FORMAT           | Min | Max  |   |
|------------------|-----|------|---|
| <b>HEX2ARRAY</b> | 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   |
|----------------|--|
| <b>VERSION</b> | --   |
| <b>UNIT</b>    | INFO<br>STATUS<br>CLOCKSTATUS<br>IPCONFIG                                      |
| <b>PORT</b>    | INFO<br>STATUS<br>TXSTATS<br>RXSTATS<br>RATES<br>SETPORTCONFIG<br>COPPERMODULE |
| <b>STREAM</b>  | CONFIG<br>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**

|                    |   |
|--------------------|---|
| <b>LABEL</b>       | Header field name                                   |
| <b>DESCRIPTION</b> | Field description                                   |
| <b>LENGTH</b>      | Length (integer)                                    |
| <b>SCALE</b>       | Scale (eg bits, bytes)                              |
| <b>FORMAT</b>      | See <a href="#">apgSetStream</a> (Section 9.3)      |
| <b>EDITABLE</b>    | 0 = Fixed (value cannot be changed)<br>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

|            |                       |
|------------|-----------------------|
| <b>PS</b>  | Picoseconds           |
| <b>NS</b>  | Nanoseconds (default) |
| <b>US</b>  | Microseconds          |
| <b>MS</b>  | Milliseconds          |
| <b>SEC</b> | Seconds               |
| <b>MIN</b> | Minutes               |

## APPENDIX A - QUICK REFERENCE GUIDE

| TCL API VERSION                                      | [VAR]  |
|--|--|
| apgGetApiVersion [VAR]                               | COMPANY<br>BRAND<br>DESCRIPTION<br>VERSION<br>BUILD_DATE<br>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<br>PORT_COUNT<br>API_MINIMUM<br>SERIAL<br>PRODUCT<br>HWARE_VERSION<br>FPGA_VERSION<br>FPGA_BUILDDATE<br>FIRMWARE_VERSION<br>FIRMWARE_BUILDDATE |
| apgLoadUnit [UNITID] IPCONFIG                        |  |
| → apgGetUnit [UNITID] IPCONFIG [VAR]                 | IPADDRESS<br>IPMASK<br>IPGATEWAY   |
| apgLoadUnit [UNITID] STATUS                          |  |
| → apgGetUnit [UNITID] STATUS [VAR]                   | UPTIME<br>TEMP<br>FAN<br>SELFTEST<br>READY   |
| apgLoadUnit [UNITID] CLOCKSTATUS (APGV2 only)        |  |
| → apgGetUnit [UNITID] CLOCKSTATUS [VAR] (APGV2 only) | CLOCKMODE<br>PLLSTATE  |
| apgSetUnit [UNITID]                                  | CLKINMODE<br>CLKOUTMODE<br>TSRESETMODE   |
| → apgApplyUnit [UNITID]                              | CLOCKMODE  |
| apgControlUnit [COMMAND] [UNITID]                    | TSCLEAR  |
| apgControlUnit [SYNC COMMAND] [UNITID]               | SYNCGO<br>SYNCCLEARUP<br>TSSYNCCLEAR   |

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

---

**apgControlPort [COMMAND] [PORTLIST]**

STARTTX  
STEPTX  
PAUSETX  
STOPTX  
CLEARCOUNTERS

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

---

**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] | CYCLES  |
| → apgGetPort [PORTID] ANALYSIS TXRX [VAR] | SAMPLES |
| → apgGetPort [PORTID] ANALYSIS RXXR [VAR] | 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]<br>Eg for MACHEADER:<br>MACHEADER DA<br>MACHEADER SA |
|--|---|

|   |                              |
|---|------------------------------|
| → apgGetStream [STREAMID] PAYLOAD [VAR] | PAYLOAD_DATA<br>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]<br>Eg for MACHEADER:<br>MACHEADER DA 08:00:00:11:22:33<br>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
#
# PORT 1 #####
set PORTID {UNITID 1}
apgSetPort $PORTID UPDATE 6
apgSetPort $PORTID MODULE_STATUS 113
apgSetPort $PORTID SPEED 8
apgApplyPort $PORTID STATE

# STREAM 1.0.0 #####
set STREAMID {UNITID 1 0 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

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

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
```

|   |   |
|---|---|
| <pre> 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         </pre>       | VLAN #1   |
| <pre> 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         </pre> | VLAN #2   |
| <pre> 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         </pre>            | VLAN #3   |
| <pre> 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         </pre> | VLAN #4   |
| <pre> apgSetStream \$STREAMID HEADER ETHERNET_II ETHERTYPE 0x0812         </pre>  | ETHERNET_II                                     |
| <pre> apgSetStream \$STREAMID HEADER USERDEFINED16 DATA 0x0000 INC 0x0010 0x0000 0x2000         </pre>  | USERDEFINED16                                   |
| <pre> 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         </pre>  | Payload configuration                           |
| <pre> apgApplyStream \$STREAMID         </pre>  | Apply stream config                             |
| <pre> # STREAM 1.0.1 ##### set STREAMID {UNITID 1 0 1} apgSetStream \$STREAMID CONFIG ENABLE 0 apgApplyStream \$STREAMID         </pre>   | Stream 1<br>... Disabled<br>Apply stream config |
| <pre> # STREAM 1.0.2 ##### set STREAMID {UNITID 1 0 2} apgSetStream \$STREAMID CONFIG ENABLE 0 apgApplyStream \$STREAMID         </pre>   | Stream 2<br>... Disabled<br>Apply stream config |
| <pre> # STREAM 1.0.3 ##### set STREAMID {UNITID 1 0 3} apgSetStream \$STREAMID CONFIG ENABLE 0 apgApplyStream \$STREAMID         </pre>   | Stream 3<br>... Disabled<br>Apply stream config |
| <pre> # STREAM 1.0.4 ##### set STREAMID {UNITID 1 0 4} apgSetStream \$STREAMID CONFIG ENABLE 0 apgApplyStream \$STREAMID         </pre>   | Stream 4<br>... Disabled<br>Apply stream config |
| <pre> # STREAM 1.0.5 ##### set STREAMID {UNITID 1 0 5} apgSetStream \$STREAMID CONFIG ENABLE 0 apgApplyStream \$STREAMID         </pre>   | Stream 5<br>... Disabled<br>Apply stream config |
| <pre> # STREAM 1.0.6 ##### set STREAMID {UNITID 1 0 6} apgSetStream \$STREAMID CONFIG ENABLE 0 apgApplyStream \$STREAMID         </pre>   | Stream 6<br>... Disabled<br>Apply stream config |
| <pre> # STREAM 1.0.7 ##### set STREAMID {UNITID 1 0 7} apgSetStream \$STREAMID CONFIG ENABLE 0 apgApplyStream \$STREAMID         </pre>   | Stream 7<br>... Disabled<br>Apply stream config |
| <pre> # END OF FILE #####         </pre>  |   |

The logo for AXTRINET, featuring a stylized 'A' with a dot and 'X' followed by the word 'TRINET' in a clean, sans-serif font.

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](mailto:support@axtrinet.com)