CONTENTS

What is ValkyrieCLI?
CLI API
  Basics
  Commands and Status messages
  First steps
Scripting...
  Port configuration
  Test Logic
  Login as example
Automating Test Suites
Valkyrie REST Server
What is ValkyrieCLI?

ValkyrieCLI is installed together with all the other software applications in every Valkyrie release.

Command Line Interface (CLI) to the Xena Server (chassis)
  • Covers all functionality
Supports interactive sessions
  • Command and Reply
  • Replay of command lists

Scripted or programmed sessions
  • Pre-programmed actions and check

Reuse of existing GUI setup
  • Jump start script writing
  • Use as template
ValkyrieCLI Advantages

Versatile
Can be used with ANY scripting and program language
Can run in ANY operating system
No need to install drivers or proprietary programs
Re-use your existing automation framework

Fast
Extremely low communication overhead
Very simple and efficient protocol
No requirements to the CLI client

Simple
Purely text based
No binary modules
Easy to learn
Well documented
Easy to debug
Reuse existing configurations
CLI API - BASICS

Xena Automation is based on:

1. A TCP connection from script/program to Valkyrie tester
2. Input/output via CLI-like API commands over the TCP socket

TCL Example

Creating a TCP Socket –

```
set s [socket -async $chassis_ip $chassis_port]
```

Communicating via Socket (s) –

```
puts $s "c_logon $chassis_pass"
gets $s response
```
CLI API - BASICS

Xena Automation is based on:

1. A TCP connection from script/program to Valkyrie tester
2. Input/output via CLI-like API commands over the TCP socket

Ruby Example:

Creating a TCP Socket –

```ruby
require 'socket'  # Sockets are in standard library
s = TCPSocket.open($hostname, $port)
```

Communicating via Socket (s) –

```ruby
socket.puts(tx_string)
response = socket.gets
```
CLI API - BASICS

Xena Automation is based on:

1. A TCP connection from script/program to Valkyrie tester
   +
2. Input/output via CLI-like API commands over the TCP socket

Python Example:

Creating a TCP Socket –

```python
from LabUtils.Drivers.SocketDrivers import SimpleSocket
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

Communicating via Socket (s) –

```python
sock.send(cmd + '\n')
response = sock.recv(1024)
```
CLI API - BASICS

Xena Automation is based on:

1. A TCP connection from script/program to Valkyrie tester
   +
2. Input/output via CLI-like API commands over the TCP socket

Bash Example:

Creating a TCP Socket –

```bash
exec 3<> /dev/tcp/${MACHINE}/${PORT}
```

Communicating via Socket (s) –

```bash
echo -en "C_LOGON ${PASSWORD}\n" >&3
read <&3
```
CLI API - BASICS

Remember:

<table>
<thead>
<tr>
<th>TCP Port used for connecting:</th>
<th>22611</th>
</tr>
</thead>
<tbody>
<tr>
<td>Once TCP Socket is open, ASCII text commands are used, must be terminated by:</td>
<td>CR/LF</td>
</tr>
</tbody>
</table>
CLI API - Commands

Command Structure

`module/port command [index] value value ...`

Example:

```
2/5 PS_RATEPPS [3] 500000
```

Meaning:

- Module is no. 2
- Port on module is no. 5
- Command is: `PS_RATEPPS` = Set Stream Rate as Packets Per Second
- Stream on port is no. 3 (`SID`)
- Packets Per Second is set to 500000

- All indices start at zero

* Use “?” as value to READ current setting (explained in next slide)
Most commands can be used both to:
query the current status/data using “?” or
modify/set data via the actual value you would like to set:

You would query for the current value this way:

```
0/5 PS_RATEPPS [3] ?
```

And the chassis would respond the same way that you set the value yourself:

```
0/5 PS_RATEPPS [3] 500000
```
C_.........CHASSIS PARAMETER
M_...... MODULE PARAMETER
P_...... PORT PARAMETER
PS_.....STREAM PARAMETER
PM_.... MATCH TERM PARAMETER
PL_..... LENGTH TERM PARAMETER
PF_......FILTER PARAMETERS
PC_.... CAPTURE PARAMETER
PT_.... TRANSMIT STATISTICS PARAMETER
PR_..... RECEIVE STATISTICS PARAMETER
PD_..... DATASET PARAMETER(HISTOGRAMS)
PP_.... 40/100G PARAMETER
CLI API - Commands

Special scripting commands

Commands for supporting the scripting process itself:

**sync**

Produces a reply of `<SYNC>`, helpful when parsing and delimiting returned lines.

**sync on**

Automatically set “sync” after each command.

**sync off**

Disables “sync on”.

**wait n**

Waits for the specified number of seconds, (max 60), then replies `<RESUME>`.

**help ?**

Gives an overview of the built-in help function.

**help ”cmd”**

Gives a brief overview of the syntax for “cmd”
The set/change commands themselves simply produce a reply from the chassis of:

<OK>

If something is unacceptable to the chassis, it may return one of the following:

- <NOTLOGGEDON> You have not issued a C_LOGON providing the chassis password.
- <NOTRESERVED> You have not issued a x_RESERVATION for the resource you want to change.
- <NOTWRITABLE> The parameter is read-only.
- <NOTREADABLE> The parameter is write-only.
- <NOTVALID> The operation is not valid in the current chassis state, e.g. because traffic is on.
- <BADMODULE> The module index value is out of bounds.
- <BADPORT> The port index value is out of bounds.
- <BADINDEX> A parameter sub-index value is wrong.
- <BADSIZE> The size of a data value is not appropriate.
- <BADVALUE> A value is not appropriate.
- <FAILED> An operation failed to produce a result.
Status messages

If there is a plain syntax error, misspelled parameter, or an inappropriate use of module/port/indices, the chassis will return a line pointing out the column where the error was detected, e.g.:

```
0/5 PS_RATEPPS [] 5q00
   ^
#Syntax error in column 24
```
CLI API – First Steps

First open connection via script/program or via Xena Script Client.

Then authenticate the connection to the chassis and provide a username for reservation:

C_LOGON "xena"
C_OWNER "example"
CLI API – First Steps
An automated test script has two main components:

1 - Port Configuration
2 - Test Logic
Scripting

Port Configuration

• Includes Port, Streams, Capture, Filters, Histograms Configuration commands
• Can be exported via the GUI
• Can be imported to a port via a small method
Scripting – Port Configuration
Scripting

Test Logic
• If/else/while/etc... logic provided by the scripting or program language
• Real-time interaction commands with Xena referred to as the SCRIPT CORE COMMANDS
• Real-time interaction commands with the DUT

Connect
   Connect / Login
Reservation
   Reserve / Release / +IsReserved? / +IsReservedByMe?
Port
   PortLinkUp / PortLinkDown / +HasLink?
Configuration
   Clear / Load
Traffic
   Start / Stop
Capture
   Start / Stop / Save
Results
   Clear / Get (Port / Stream / Filter) / Save (Port / Stream / Filter) to .CSV
# ------------------ Login ------------------
proc Login {s chassis_pass chassis_user console} {

    set pf_flag 1

    puts "$s "c_logon "$chassis_pass"
    gets $s response
    if {"$response" == ""} { gets $s response }

    if {"$response" != "<OK>"} { set pf_flag 0 }
    if {$console == 1} { puts "Logging | "$response" }

    puts "$s "c_owner "$chassis_user"
    gets $s response
    if {"$response" == ""} { gets $s response }

    if {$console == 1} { puts "Owner | "$response" }
    if {"$response" != "<OK>"} { set pf_flag 0 }

    return $pf_flag
}
#
# -------------------------------
Automating Test Suites

Use simple one-line shell command to automate a full RFC test based on prebuilt test configuration

TCL uses “EXEC” in order to execute a process
Valkyrie REST Server

The new Valkyrie REST server

- is a client-less, language-agnostic, out-of-the-box, chassis embedded REST server.
- lets you build automation scripts with your choice of language, tool and client environment.
- supports all Valkyrie CLI commands and adds many more abstract operations (like returning statistics as ready-to-consume JSON) that simplify and speed up automation development.

With the REST server Xena also releases a full Python object-oriented REST client

- enabling Python developers to simply pip install it and start building automation scripts without spending time on developing the traffic generator layer.

For more information see:

[Valkyrie REST Server User Manual](#)
Want more?

CHECK TECHNICAL DOCUMENTATION

TRY OUR LIVE DEMO SYSTEM

BOOK A GUIDED SW TOUR

CONTACT US:
support@xenanetworks.com