

COMPUTING

ViewCheck™ on SharpStreamer™ PCIE-7207 for MaxCore™

User Guide

P/N: 6806800U30B

December 2016

ARTESYN[™]
EMBEDDED TECHNOLOGIES

© Copyright 2016 Artesyn Embedded Technologies, Inc.
All rights reserved.

Trademarks

Artesyn Embedded Technologies, Artesyn and the Artesyn Embedded Technologies logo are trademarks and service marks of Artesyn Embedded Technologies, Inc. All other names and logos referred to are trade names, trademarks, or registered trademarks of their respective owners. © 2016 Artesyn Embedded Technologies, Inc. All rights reserved. For full legal terms and conditions, please visit www.artesyn.com/legal.

Notice

While reasonable efforts have been made to assure the accuracy of this document, Artesyn assumes no liability resulting from any omissions in this document, or from the use of the information obtained therein. Artesyn reserves the right to revise this document and to make changes from time to time in the content hereof without obligation of Artesyn to notify any person of such revision or changes.

Electronic versions of this material may be read online, downloaded for personal use, or referenced in another document as a URL to an Artesyn website. The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of Artesyn.

It is possible that this publication may contain reference to or information about Artesyn products (machines and programs), programming, or services that are not available in your country. Such references or information must not be construed to mean that Artesyn intends to announce such Artesyn products, programming, or services in your country.

Limited and Restricted Rights Legend

If the documentation contained herein is supplied, directly or indirectly, to the U.S. Government, the following notice shall apply unless otherwise agreed to in writing by Artesyn.

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data clause at DFARS 252.227-7013 (Nov. 1995) and of the Rights in Noncommercial Computer Software and Documentation clause at DFARS 252.227-7014 (Jun. 1995).

Contact Address

Artesyn Embedded Technologies
Marketing Communications
2900 S. Diablo Way, Suite 190
Tempe, Arizona 85282

Artesyn Embedded Technologies
Lilienthalstr. 17-19
85579 Neubiberg/Munich
Germany

Contents

About this Manual	9
1 Introduction	13
1.1 Terminology	13
1.1.1 Test Identification	13
1.1.2 Monitor ID	15
1.1.3 Error ID	15
1.2 Components of ViewCheck	16
2 Validating ViewCheck	19
2.1 Access and Execution of a Test Using CLI	19
2.2 Access and Execution of a Test Using XML	23
2.2.1 Authenticate	24
2.2.2 Configure	25
2.2.3 GetClassList	26
2.2.4 DescribeClass	28
2.2.5 InvokeMethod	33
2.3 Access and Execution of a Test Using GUI	35
2.4 ViewCheck Service Log Information	35
2.5 ViewCheck TestLog Information	35
3 Commands Execution	37
3.1 Generic Commands	37
3.1.1 config-watch-dog	37
3.1.2 configure-error-strings	38
3.1.3 configure-log-path	39
3.1.4 mcpDeviceCategory	40
3.1.5 hw-inventory-list	40
3.1.6 list-all-tests	41
3.1.7 purge-all-log	41
3.1.8 set-log-level	42
3.1.9 show-all-monitors	43
3.1.10 show-all-test-status	43
3.1.11 show-diag-scan-result	44

3.1.12	show-log-content	44
3.1.13	show-heart-beat	45
3.1.14	show-watch-dog	45
3.1.15	start-watch-dog	46
3.1.16	show-log-info	46
3.1.17	show-log-level	47
3.1.18	show-running-mode	47
3.1.19	show-system-error-log	48
3.1.20	start-all-monitors	48
3.1.21	start-diag-scan	49
3.1.22	stop-all-monitors	51
3.1.23	stop-diag-scan	51
3.1.24	stop-watch-dog	52
3.1.25	switch-mode	52
3.1.26	show-version	53
3.2	Test Management and Control Commands	54
3.2.1	start-test	54
3.2.2	show-test-help	56
3.2.3	list-tests	56
3.2.4	show-test-status	57
3.2.5	show-test-result	58
3.2.6	stop-test	59
3.3	Monitoring Commands	60
3.3.1	show-monitor-id	60
3.3.2	show-poll-interval	61
3.3.3	show-lower-threshold-info	62
3.3.4	show-upper-threshold-info	63
3.3.5	start-monitor	64
3.3.6	set-lower-threshold-info	65
3.3.7	set-upper-threshold-info	66
3.3.8	set-threshold-default	68
3.3.9	set-poll-interval	69
3.3.10	set-poll-interval-default	70
3.3.11	stop-monitor	71
3.3.12	set-rate	72
3.3.13	show-rate	73

3.3.14	show-network-counter-log	74
3.3.15	show-network-error-log	74
3.3.16	exit	75
3.4	In-service Monitoring Specifics	75
3.4.1	Monitoring Hardware Device Status	75
3.4.2	Monitoring Device Critical Errors	77
3.4.3	Software Watchdog	77
3.4.3.1	Watchdog Related Commands	78
3.4.4	Monitoring Heartbeat	78
3.4.4.1	show-heartbeat-info	79
3.5	Out of Service Diagnostics Tests	79
3.5.1	CPU	79
3.5.2	Memory	80
3.5.3	OS	80
3.5.4	PCI	81
3.5.5	Network	81
3.5.6	RTC	81
4	Related Documentation	83
4.1	Artesyn Embedded Technologies - Embedded Computing Documentation	83

List of Tables

Table 3-1	Possible Actions	37
Table 3-2	Parameter of configure-error-strings	38
Table 3-3	Parameter of configure-log-path	39
Table 3-4	Parameters of set-log-level Values	42
Table 3-5	Parameters of set-log-level Arguments	42
Table 3-6	Parameters of show-log-content	44
Table 3-7	Parameters of start-all-monitors	49
Table 3-8	Parameters of start-diag-scan	50
Table 3-9	Parameters of stop-all-monitors	51
Table 3-10	switch-mode Arguments	53
Table 3-11	Parameters of start-test	54
Table 3-12	Parameters of show-test-help	56
Table 3-13	Parameters of show-test-status	58
Table 3-14	Parameters of show-test-result	59
Table 3-15	Parameters of stop-test	60
Table 3-16	Parameters of show-poll-interval	61
Table 3-17	Parameters of show-lower-threshold-info	62
Table 3-18	Parameters of show-upper-threshold-info	63
Table 3-19	Parameters of start-monitor	64
Table 3-20	Parameters of set-lower-threshold-info	65
Table 3-21	Parameters of set-upper-threshold-info	67
Table 3-22	Parameters of set-threshold-default	68
Table 3-23	Parameters of set-poll-interval	69
Table 3-24	Parameters of set-poll-interval-default	70
Table 3-25	Parameters of stop-monitor	71
Table 3-26	Parameters of set-rate	72
Table 3-27	Parameters of show-rate	73
Table 3-28	Monitors	76
Table 3-29	Watchdog Related Commands	78
Table 4-1	Artesyn Embedded Technologies - Embedded Computing Publications	83

About this Manual

Overview of Contents

This guide provides detailed information about installation, configuration, and usage of ViewCheck™. This guide is divided into the following chapters and appendices.

- [About this Manual](#) lists all the conventions and abbreviations used in this manual and outlines the revision history of the manual.
- [Introduction](#) describes the components and features of ViewCheck.
- [Validating ViewCheck](#) describes the procedures to access and execute tests using CLI, GUI, and XML interface.
- [Commands Execution](#) describes the generic, monitoring, test management and control commands.
- [Additional Information](#) describes the integration of ViewCheck with Nagios^(R).
- [Related Documentation](#) lists the relevant manuals and provides additional information.

Abbreviations

The following table lists the abbreviations used throughout the document.

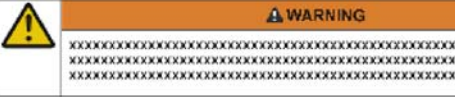

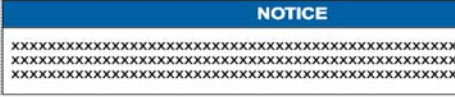
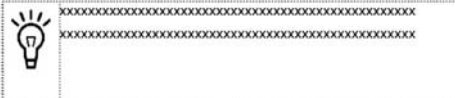
Abbreviation	Definition
BBS	Basic Blade Services
Client	The applications used to access ViewCheck via various Interfaces (CLI and XML).
INSM	In Service Monitoring. Functional module in ViewCheck framework handling the monitoring functionality of various critical parameters in the blade.
NRPE	Nagios Remote Plug-in Executor)
NSCA	Nagios Service Check Acceptor
OOSD	Online Out of Service Diagnostics. Functional module in ViewCheck framework that manages Test Management requests related to Out of Service Diagnostics.

Abbreviation	Definition
PCIe	Peripheral Component Interconnect Express
SSF	System Services Framework
TCL	Tool Command Language
XML	Extensible Markup Language

Conventions

The following table describes the conventions used throughout this manual.

Notation	Description
0x00000000	Typical notation for hexadecimal numbers (digits are 0 through F), for example used for addresses and offsets
0b0000	Same for binary numbers (digits are 0 and 1)
bold	Used to emphasize a word
Screen	Used for on-screen output and code related elements or commands in body text
Courier + Bold	Used to characterize user input and to separate it from system output
<i>Reference</i>	Used for references and for table and figure descriptions
File > Exit	Notation for selecting a submenu
<text>	Notation for variables and keys
[text]	Notation for software buttons to click on the screen and parameter description
...	Repeated item for example node 1, node 2, ..., node 12
.	Omission of information from example/command that is not necessary at the time being
..	Ranges, for example: 0..4 means one of the integers 0,1,2,3, and 4 (used in registers)

Notation	Description
	Logical OR
	Indicates a hazardous situation which, if not avoided, could result in death or serious injury
	Indicates a hazardous situation which, if not avoided, may result in minor or moderate injury
	Indicates a property damage message
	No danger encountered. Pay attention to important information

Summary of Changes

Part Number	Date	Description
6806800U30B	December 2016	Updated Table 3-28 .
6806800U30A	September 2016	Initial Version

ViewCheck™ is a comprehensive software service used to diagnose, manage and monitor Artesyn products. The diagnostic utilities of ViewCheck helps in identifying, detecting, and locating hardware issues on a SharpStreamer PCIe-7207 card. ViewCheck also provides mechanisms to monitor status of CPU temperature, CPU core usage, memory usage, power sensor value, temperature sensors value, Ethernet counters, and errors.

ViewCheck also supports user-triggered test cases and event notification mechanisms.

Note: In this manual, 'ViewCheck' and 'Diagnostics' are used interchangeably, as Diagnostics is the core part of ViewCheck.

ViewCheck can be accessed locally using Command Line Interface (CLI), Graphical User Interface (GUI), and Extensible Markup Language (XML) interfaces provided via System Services Framework (SSF) service. ViewCheck runs in any of the following modes:

- **In-service Diagnostics**

In this mode, the diagnostics service can run even while the customer applications are running on the PCIe-7207 card.

ViewCheck can monitor key hardware parameters like CPU temperature, network counters, and network errors. It can also be used for watching kernel critical errors logged by various hardware devices and device drivers.

- **Out-of-service Diagnostics**

In this mode, ViewCheck can execute all Out-of-Service Diagnostics (OOSD) tests along with the in-service diagnostics activities. These tests can only be executed when the PCIe-7207 card is offline and is not providing any service.

For more information on commands supported for in-service and OOSD, see [Commands Execution on page 37](#).

1.1 Terminology

This section explains terminology and keywords used in ViewCheck services.

1.1.1 Test Identification

The tests performed by ViewCheck have a unique test identification that is based on the following triple key.

< Device Category, Test ID, Device Instance >

Device Category

The Device Category is an enumerated value, reused from similar enumeration already defined in the HPI-B Standard specification.

It is used to express commonly in use devices, such as storage, network, serial, CPU, and memory, available on all the PCIE-7207 cards, irrespective of their function and architecture. This category is used in commands; as one of the key fields to uniquely identify a particular test.

Device category allows to:

- Group test cases, as per the category; for display and statistical purpose.
- Reuse test IDs across the device categories.

Test ID

Test ID is an integer value that uniquely identifies the actual test that can be invoked or executed on a hardware device instance, which belongs to a specific device category available on the PCIE-7207 card. The following are the examples of tests that can be executed on the devices.

- Ping Flood test
- Network connectivity test in case of Network Device Category
- Bad Blocks test in case of Storage Device category
- Temperature tests in case of CPU categories

Each of these tests would be associated with a unique Test ID. These Test IDs start with value '0' and increase linearly for various sub-tests in a device category.

Some tests may be applicable to all the device instances in a particular device category. The combination of <Device category, Test ID, Device Instance> would be unique and provides capability to control, execute, and manage the test on a device instance in a device category. With this mechanism, the same test can be simultaneously initiated or triggered on multiple device instances under that device category, thus providing parallel execution of tests.

Device Instance

Hardware devices uniquely identified and recognized by the drivers and OS on the PCIE-7207 card are treated as device instances. A device instance can belong to a particular device category. Tests can be invoked and executed on this device instance. OS and driver support to access the device is assumed to be readily available.

For example, device instances eth0, eth1, eth2 or Base 0, Base 1 are used to identify unique devices in the networking device category. Similarly, hda1, hda2, and so on can identify unique instances of devices in the storage category. Device instances use the standard nomenclature already defined by the OS (for instance Linux) on the PCIE-7207 card.

A diagnostics test identified with "`< Device Category, Test ID, device Instance >`" is executed on the specified device instance.

1.1.2 Monitor ID

ViewCheck service monitors pre-identified parameters for hardware devices. These parameters include CPU temperature, CPU core usage, memory usage, power sensor value, temperature sensors value, Ethernet counters, and errors. To periodically poll and check these parameters, the ViewCheck service uses CLI and XML configuration. For each parameter of interest, a Monitor ID is an enumerated constant that uniquely represents the monitoring entity. ViewCheck uses this value to control monitoring and to report events via XML on these monitors.

1.1.3 Error ID

Error ID is to provide identification for pre-determined errors/warnings of hardware devices generated by the device driver or the kernel on the PCIE-7207 card. These critical error messages are indications of abnormal behavior on part of the kernel or the hardware device on the PCIE-7207 cards. ViewCheck functionality attempts to detect all such errors and provides suitable information to external high-level software intelligence to act upon.

The list of messages that constitute these errors is not standardized by the hardware device vendor nor the Linux Kernel Community. Error ID attempts to standardize all such messages on Artesyn products. These messages would be OS and driver specific. Mostly, the same Error ID would be associated with the same category of error across PCIE-7207 card and OS. For more information on commands, refer to [Commands Execution on page 37](#).

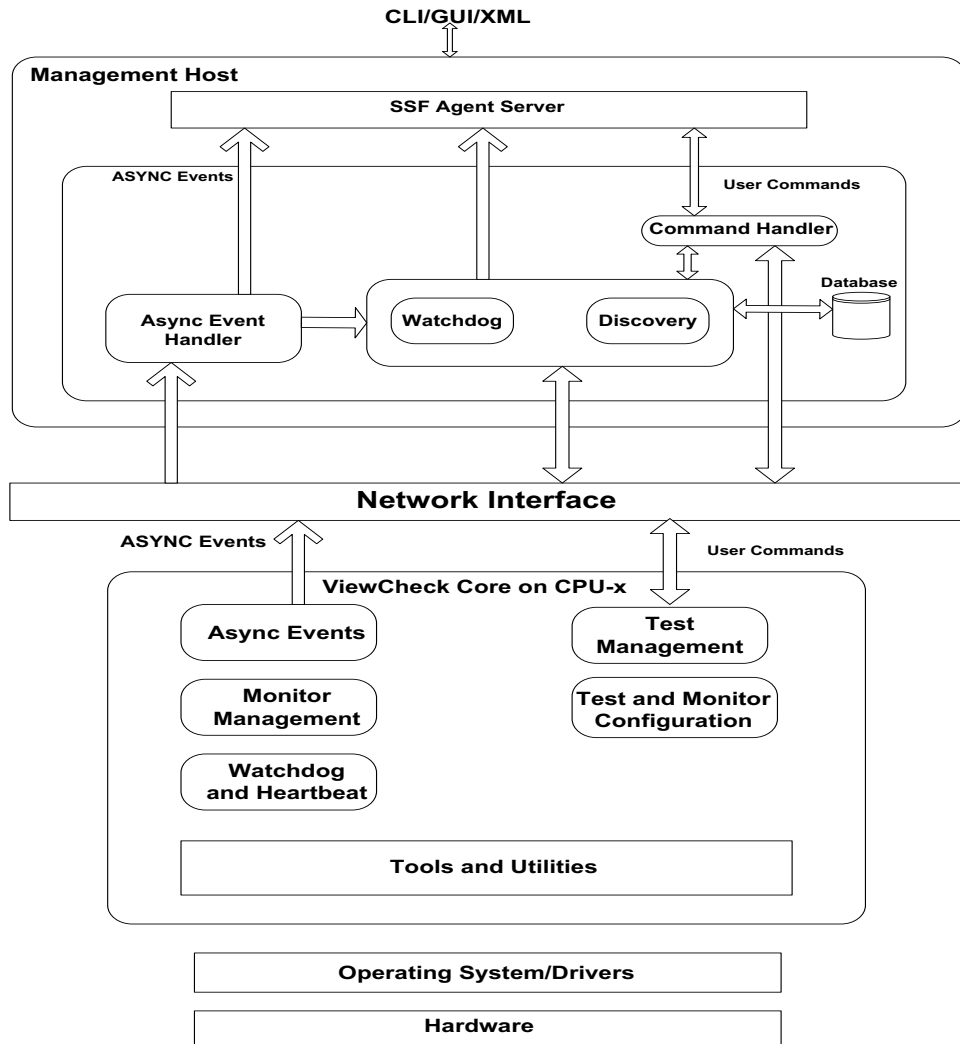
Note: Monitor ID and Error ID are used for in-service monitoring.

1.2 Components of ViewCheck

ViewCheck has the following components:

- **Management Host:** The Management Host refers to the Physical Execution Environment (PEE) on a particular server. PEE is used to indicate the components that are used collectively with the applications code to make the complete system. For example, the processors, networks, operating systems and so on.
- **Watchdog:** Watchdog feature allows you to check the health of MCP. If MCP becomes unresponsive, the watchdog timer will expire and a configured action will be taken by the watchdog timer.
- **Heartbeat:** The heartbeat feature allows user applications or any other services running on MCP to register for regular health check with ViewCheck core.

Figure 1-1 ViewCheck Block Diagram



Validating ViewCheck

This chapter explains various methods for accessing ViewCheck services on Artesyn products. It describes the procedures to access and execute tests using CLI, GUI, and XML interface.

2.1 Access and Execution of a Test Using CLI

Using CLI, you can start, stop, and query the status of a particular test. Also, you can configure the monitoring parameters. The ViewCheck CLI can be accessed via a console using Secure Shell (SSH).

You can access ViewCheck CLI using the following procedure:

1. Establish the secure shell using SSH or Putty.
2. Start the **Telnet** connection from an already established secure shell.


```
[root@pcie7410-s1-cl ~]# telnet localhost 11001
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^'.
Welcome to SSF CLI
```
3. Type user name and password.


```
Username: Admin
Password:
Access granted
>enable
#configure terminal
MaxCore(config)#system 1
MaxCore(system-1)#shelf 1
MaxCore(shelf-1-1)#PCIEslot 3
MaxCore(PCIEslot-1-1-3)#PCIECard 1
MaxCore(PCIECard-1-1-3-1)#PCIECardPort 1
MaxCore(PCIECardPort-1-1-3-1-1)#mcp 1
MaxCore(mcp-1-1-3-1-1-1)#mcpVirExecEnv vee0
```

```
MaxCore(mcpVirExecEnv-1-1-3-1-1-1-vee0)#mcpDiagnostic
```

NOTICE

By default, the administrator user name and password are "Admin".

After logging into the ViewCheck CLI, you can list all the supported commands by typing ' ? ' on the CLI console. Following is an example.

```
MaxCore(mcpDiagnostic-1-1-6-1-1-1-vee0)#?
```

Command	Description
config-watch-dog	Configures Watchdog.
configure-error-strings	User configurable error strings, notified by ViewCheck when reported by the device.
configure-log-path	Configures the log path where the results are stored.
device-category	Configures the device category.
diag-service	Diagnostics operation (status/start/stop/restart).
exit	Exit from diagnostic configuration mode.
get-registered	To get the registered flag stored in diagnostics.
hw-Inventory-list	Provides the Inventory of Hardware (Type, Vendor ID, Major Number, Minor Number and any associated Details) as detected by the Diagnostics Software.
no	Negate a command or set its defaults.
purge-all-log	All log files are Zipped and stored away.
purge-log	Purge specific log file.
reload	Indicates there are updates to the XML configuration file provided for User edits. The Configuration file is re-read and test database is re created.
set-log-level	Sets the Log Level of diagnostic application.
set-registered	To set the registered flag stored in diagnostics.
show-heart-beat	Displays status of Heartbeat for all registered services.

Command	Description
<code>show-log-content</code>	Displays the content of log files related to Rawlogs and ResultsLog of ViewCheck.
<code>show-log-info</code>	List all Log files of diagnostics application.
<code>show-log-level</code>	Displays list of log level of diagnostics.
<code>show-monitor-value</code>	Displays current value of the requested monitor.
<code>show-running-mode</code>	Displays the current running mode of ViewCheck.
<code>show</code>	Show running system information.
<code>show-system-error-log</code>	Show system error log.
<code>show-version</code>	Displays version of diagnostics.
<code>show-watch-dog</code>	Displays current configuration of watchdog.
<code>start-all-monitors</code>	Starts all the supported Monitors available on the board or related to a device category.
<code>start-diag-scan</code>	Executes the diagnostics tests supported on the Board.
<code>start-watch-dog</code>	Starts the watchdog.
<code>stop-all-monitors</code>	Stops all the supported monitors available on the board or related to a device category.
<code>stop-diag-scan</code>	Stop the diagnostics scan tests.
<code>stop-watch-dog</code>	Stops the watchdog.

You can enter into device category mode by executing the command `device-category` on CLI and view the list of commands supported only at the device category level.

```
MaxCore(mcpDiagnostic-1-1-6-1-1-1-vee0)#mcpDeviceCategory
processor
```

Command	Description
exit	Exit from deviceCategory configuration mode.
list-instances	List all the possible device instances in present deviceCategory.
set-lower-threshold-info	Set the Lower Threshold value for the Monitor.
set-poll-interval-default	Set Poll Interval to default value.
set-poll-interval	Set the Poll Interval.
set-rate	Set the rate of change value for network monitors.
set-threshold-default	Set Threshold to default value.
set--upper-threshold-info	Set the Upper Threshold value for the Monitor.
show-lower-threshold-info	Show Lower Threshold info.
show-monitor-id	List all monitors for the deviceCategory.
show-network-counter-log	Show network counters log.
show-network-error-log	Show network errors log.
show-poll-interval	Show Poll Interval.
show-rate	Show the rate of change value for network monitors.
show	Show running system information.
show-test-help	Brief help on the usage of the Specific Test referred by Test ID.
show-test-list	User can use this command to get information on the available Diagnostic tests with details like tests and sub tests associated, along with Test IDs.
show-test-result	Show test result.
show-test-status	Show test status.

Command	Description
<code>show-upper-threshold-info</code>	Show upper threshold information.
<code>start-monitor</code>	Start a monitor.
<code>start-test</code>	Start a test.
<code>stop-monitor</code>	Stop a monitor.
<code>stop-test</code>	Stop a test.

After logging into the CLI, you can start, stop, and query a test from the CLI. You can view the details of the test by executing the `show-test-help` command.

To start a test, you can run the `start-test` command with `testId`, `deviceInstance`, and (optional) arguments as input to the command.

After test execution, the results can be viewed by the `show-test-result` command, which displays the test result and a raw log generated by that test.

Using the CLI, you can list all the in-service diagnostics monitors in a specific device category. By default, all the monitors start when ViewCheck is initialized. You can start and stop any monitor using the `start-monitor` and `stop-monitor` commands.

To exit from the ViewCheck CLI, use the following command.

```
MaxCore(mcpDiagnostic-1-1-6-1-1-1-vee0)#exit
```

2.2 Access and Execution of a Test Using XML

XML interface support methods, classes, and event notification mechanisms. Using XML, you can start, stop, query, and configure the parameters related to tests and monitors. Similar to CLI, the XML interface can be accessed using SSF. XML notifications are generated with the following details:

- State changes about the diagnostic test under execution
- Pre-determined monitor; exceeding the set threshold value
- Occurrence of any pre-determined hardware device error/warning generated by the device driver or the kernel.

For more information on executing ViewCheck tests using XML commands, refer to SSF for MaxCore™ Platform XML Interface Guide.

You can access ViewCheck XML interface similar to the ViewCheck CLI.

1. Establish the secure shell using SSH or Putty.
2. Start the Telnet connection from an already established secure shell.

```
telnet localhost 15550
```

```
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
```

2.2.1 Authenticate

After a connection is established with XML Agent, the only command that should be executed is `Authenticate`. This command is for evaluating user credentials.

The `Authenticate` command contains user name and password for verification. Successful authentication is signified by the success response, otherwise an error message is returned.

You need to enter the user credentials to XML Agent using the command `Authenticate`. The XML Agent forwards the authentication request to the SSF system, which validates the user credentials and allows the XML client to access it.

- Request:

In the below request, the user credentials “Admin” and “Admin” are created by default in the SSF. But you need to provide credentials by creating them using the “CreateObject” command as shown in the section [GetClassList](#).

```
<Script><Authenticate><User>Admin</User><Password>Admin</Password></Authenticate></Script>]]]]>
```

- Response:

The below response is received if the authentication is successful.

```
<?xml version="1.0"?><Response><Success/></Response>]]]]>
```

The following response is received if authentication fails.

```
<?xml
version="1.0"?><Response><Error><ErrorCode>1</ErrorCode><Type>O
peration not allowed</Type><Description>Authentication
error</Description><CustomError/></Error></Response>]]]]>
```


2.2.2 Configure

This command configures the XML protocol for the indentation and the events to enable/disable state. There are two configuration options available:

IndentOutput option controls the indentation of the XML response produced by the MINDAgent. Its default value is '0', which means, by default, output indentation is off.

- Request:
The following request command sets the indentation to 4:

```
<Script><Configure><IndentOutput>4</IndentOutput></Configure></Script>]]>]]>
```
- Response:

```
<?xml version="1.0"?>
<Response>
  <Success/>
</Response>]]>]]>
```

EnableEvents option enables or disables events that reach XML interface from the SSF framework. Successful execution is signified by the success response. Otherwise, an error message is returned.

The values for the command is true/false. The "True" value enables the events and the "false" value disables the events.

- Request:

```
<?xmlversion ="1.0"?>
<Script>
<Configure>
<EnableEvents>true</EnableEvents>
</Configure>
</Script>]]>]]>
```
- Response:

```
<?xml version="1.0"?><Response><Success/></Response>]]>]]>
```

2.2.3 GetClassList

This command is used to retrieve all the SSF classes.

- Request:

```
<Script ><GetClassList /></Script >]]]]>
```
- Response:
The response message received from the XML Agent is shown below. The response contains all the SSF classes including the SSF framework classes along with the application defined classes.

```
<Response>  
  <Classes>  
    <Class>HardwarePlatformManager</Class>  
    <Class>session</Class>  
    <Class>shell</Class>  
    <Class>mode</Class>  
    <Class>command</Class>  
    <Class>user</Class>  
    <Class>group</Class>  
    <Class>membership</Class>  
    <Class>ACManager</Class>  
    <Class>shutdown</Class>  
    <Class>usmuser</Class>  
    <Class>traphost</Class>  
    <Class>cppscript</Class>  
    <Class>classlock</Class>  
    <Class>globallock</Class>  
    <Class>CommunityMO</Class>  
    <Class>SystemMO</Class>  
    <Class>Connection</Class>  
    <Class>Schema</Class>  
    <Class>SNMPAgentConfig</Class>  
    <Class>agent</Class>  
    <Class>logsink</Class>  
    <Class>logfilter</Class>  
    <Class>acl</Class>  
    <Class>aclclass</Class>  
    <Class>aclmember</Class>  
    <Class>aclmoid</Class>
```

```
<Class>aclclassrange</Class>
<Class>aceclass</Class>
<Class>acemember</Class>
<Class>acemoid</Class>
<Class>aceclassrange</Class>
<Class>system</Class>
<Class>shelf</Class>
<Class>PCIEslot</Class>
<Class>PCIECard</Class>
<Class>CPU</Class>
<Class>virExecEnv</Class>
<Class>endPoint_NIC</Class>
<Class>endPoint_nic_ND</Class>
<Class>linuxApplication</Class>
<Class>fan</Class>
<Class>psu</Class>
<Class>chassis</Class>
<Class>airFilter</Class>
<Class>backPlaneEC</Class>
<Class>endPoint_hdd_ND</Class>
<Class>event</Class>
<Class>bay</Class>
<Class>MCPMU</Class>
<Class>endPoint_HDD</Class>
<Class>SSD</Class>
<Class>alarm</Class>
<Class>backPlaneECPort</Class>
<Class>battery</Class>
<Class>sol</Class>
<Class>backPlaneECvNIC</Class>
<Class>PCIECardPort</Class>
<Class>PCIECardvNIC</Class>
<Class>CardConfiguration</Class>
<Class>BCMSwitch</Class>
<Class>interface</Class>
<Class>BCMPort</Class>
<Class>DSPBay</Class>
<Class>DSP</Class>
<Class>mcp</Class>
```

```
<Class>mcpVirExecEnv</Class>
<Class>mcpVirExecInterface</Class>
<Class>mcpVirExecService</Class>
<Class>mcpDiagnostic</Class>
<Class>mcpDeviceCategory</Class>
<Class>diagnostic</Class>
<Class>deviceCategory</Class>
<Class>switch</Class>
<Class>vlan</Class>
<Class>port</Class>
<Class>lb</Class>
<Class>pool</Class>
<Class>table</Class>
<Class>match_list</Class>
</Classes>
</Response>]]>]]>
```

NOTICE

ViewCheck software uses classes, diagnostics, and device category to start, stop, and query the status of tests and monitors.

2.2.4 DescribeClass

This command retrieves the description and properties of a SSF class, such as if the class can be created, deleted, and written. It also retrieves, attributes, attributes description, and attribute type. It returns class description on success or an error on failure.

- Request:
The below XML request gets the description of class, "shelf" in the SSF system.

```
<?xml version = "1.0" ?>
<Script version = "2" >
<DescribeClass>
<Class>mcpDiagnostic</Class>
</DescribeClass>
</Script>]]>]]>
<?xml version="1.0" ?>
```

- Response:

The below XML response shows various details of the class, "shelf" in the SSF system.

```

<Response>
  <ClassDesc>
    <Class>mcpDiagnostic</Class>
    <Id>1201</Id>
    <Description>In Diagnostic mode, user can configure and perform
    Diagnostic related tests, monitoring &
    updating.</Description>
    <IsSingleton>>false</IsSingleton>
    <IsCreatable>>true</IsCreatable>
    <IsDeletable>>true</IsDeletable>
    <IsWritable>>false</IsWritable>
    <MOIDDesc>
      <Name>diagnosticMoid</Name>
      <Id>1201</Id>
      <Description/>
      <MOIDItemDesc>
        <Ref>
          <ClassId>1102</ClassId>
          <MOIDId>1102</MOIDId>
        </Ref>
      </MOIDItemDesc>
    </MOIDDesc>
    <MethodDesc>
      <Name>hwInventoryList</Name>
      <Id>10</Id>
      <Description>Provides the Inventory of Hardware (Type, Vendor
      ID, Major Number, Minor Number and any associated Details) as
      detected by the Diagnostics Application. </Description>
      <IsStatic>>false</IsStatic>
      <ArgDesc>
        <Name>outputResult</Name>
        <Id>0</Id>
        <Description/>
      </ArgDesc>
    </MethodDesc>
  </ClassDesc>
</Response>

```

```
<IsInput>>false</IsInput>
<IsOutput>>true</IsOutput>
<IsOptional>>false</IsOptional>
  </ArgDesc>
</MethodDesc>
<MethodDesc>
  <Name>reload</Name>
  <Id>20</Id>
  <Description>User can issue this command when there are
updates to the User XML .</Description>
  <IsStatic>>false</IsStatic>
  <ArgDesc>
<Name>commandStatus</Name>
<Id>0</Id>
<Description/>
<TypeDesc>
  <Name/>
  <Description/>
</TypeDesc>
<IsInput>>false</IsInput>
<IsOutput>>true</IsOutput>
<IsOptional>>true</IsOptional>
  </ArgDesc>
</MethodDesc>
<MethodDesc>
  <Name>setLogLevel</Name>
  <Id>30</Id>
  <Description>Sets the Log Level of Diagnostic
Application.</Description>
  <IsStatic>>false</IsStatic>
  <ArgDesc>
<Name>logLevelValue</Name>
<Id>0</Id>
<Description>Valid log levels (3-Critical, 2-Normal, 1-
Info)</Description>
<TypeDesc>
  <Name>Integer</Name>
  <Description>Integer number in range  $(-2^{32})/2$  to  $(2^{32})/2 - 1$ </Description>
```

```

</TypeDesc>
<IsInput>>true</IsInput>
<IsOutput>>false</IsOutput>
<IsOptional>>false</IsOptional>
  </ArgDesc>
  <ArgDesc>
<Name>commandStatus</Name>
<Id>1</Id>
<Description/>
<TypeDesc>
  <Name/>
  <Description/>
</TypeDesc>
<IsInput>>false</IsInput>
<IsOutput>>true</IsOutput>
<IsOptional>>true</IsOptional>
  </ArgDesc>
</MethodDesc>
<MethodDesc>
  <Name>showLogInfo</Name>
  <Id>40</Id>
  <Description>List all Log files of Diagnostics
Application.</Description>
  <IsStatic>>false</IsStatic>
  <ArgDesc>
<Name>outputResult</Name>
<Id>0</Id>
<Description/>
<TypeDesc>
  <Name/>
  <Description/>
</TypeDesc>
<IsInput>>false</IsInput>
<IsOutput>>true</IsOutput>
<IsOptional>>false</IsOptional>
  </ArgDesc>
</MethodDesc>
<MethodDesc>
  <Name>purgeLog</Name>

```

```
        <Id>50</Id>
        <Description>Purge specific log file.</Description>
        <IsStatic>>false</IsStatic>
        <ArgDesc>
<Name>logFileName</Name>
<Id>0</Id>
<Description>log file name</Description>
<TypeDesc>
    <Name/>
    <Description>file name</Description>
</TypeDesc>
<IsInput>>true</IsInput>
<IsOutput>>false</IsOutput>
<IsOptional>>false</IsOptional>
    </ArgDesc>
    <ArgDesc>
<Name>commandStatus</Name>
<Id>1</Id>
<Description/>
<TypeDesc>
    <Name/>
    <Description/>
</TypeDesc>
<IsInput>>false</IsInput>
<IsOutput>>true</IsOutput>
<IsOptional>>true</IsOptional>
    </ArgDesc>
</MethodDesc>
<MethodDesc>
    <Name>purgeAllLog</Name>
    <Id>60</Id>
    <Description>All log files are Zipped and stored
away.</Description>
    <IsStatic>>false</IsStatic>
    <ArgDesc>
<Name>commandStatus</Name>
<Id>0</Id>
<Description/>
<TypeDesc>
```



```
<Name/>
    <Description/>
</TypeDesc>
<IsInput>>false</IsInput>
<IsOutput>>true</IsOutput>
<IsOptional>>true</IsOptional>
    </ArgDesc>
</MethodDesc>
<MethodDesc>
    <Name>showSystemErrorLog</Name>
    <Id>70</Id>
    <Description>show system error log</Description>
    <IsStatic>>false</IsStatic>
    <ArgDesc>
<Name>outputResult</Name>
<Id>0</Id>
<Description/>
<TypeDesc>
    <Name/>
    <Description/>
</TypeDesc>
<IsInput>>false</IsInput>
<IsOutput>>true</IsOutput>
<IsOptional>>false</IsOptional>
    </ArgDesc>
</MethodDesc>
    </ClassDesc>
</Response>]]>]]>
```

2.2.5 InvokeMethod

This command calls the method of a SSF Object. SSF methods are defined with method parameters such as input, output, and input-output.

SSF Object method can be invoked with a list of input or input-output arguments. This command returns a list of output or input-output arguments. The input argument means it is only an input and will not be displayed in the output. But in the case of input-output arguments, both input and output will be displayed in the output.

- **Request:**
The below example invokes the method class, "mcpDiagnostic" with instance "vee0". Upon execution of this method by the class instance, response will be sent in the output/input-output arguments.

```
Request
=====
<Script>
  <InvokeMethod>
    <Object>
      <Class>mcpdiagnostic</Class>
      <Name>vee0</Name>
    </Object>
    <Method>setLogLevel</Method>
    <Argument>
      <Name>logLevelValue</Name>
      <Value>1</Value>
    </Argument>
  </InvokeMethod>
</Script>]]>]]>
```

- **Response:**
InvokeMethod command returns the below response after executing the above example command.

```
<?xml version="1.0"?>
<Response>
  <Arguments>
    <Argument>
      <Name>commandStatus</Name>
      <Value>Set Log Level Success</Value>
    </Argument>
  </Arguments>
</Response>]]>]]>
```

To exit from the ViewCheck XML:

```
<Script>  
<Command name="Exit" />  
<Script>]]>]]>
```

2.3 Access and Execution of a Test Using GUI

ViewCheck is a web based in-service and out-of-service fault detection software for PCIe-7207. The ViewCheck helps in identifying, detecting, and locating failures on a PCIe-7207.

The ViewCheck software can be used locally as standalone ViewCheck software using CLI and XML interfaces or it can be accessed remotely as part of the SSF.

ViewCheck enables you to do the following through its GUI:

- Running ViewCheck Test
- Health Check Configuration
- General Configuration

For more details, see the SSF for MaxCore GUI Online Help.

2.4 ViewCheck Service Log Information

ViewCheck service logs are generated in `EmrDiag_Debug.log` file and is located at `/opt/diagnostics/var/log/diag/service/`

The Test result logs and raw logs generated by various tests are available at `/opt/diagnostics/var/log/diag/testlog/`

2.5 ViewCheck TestLog Information

ViewCheck internally retains data related to tests invoked by the user. If the number of tests invoked by the user exceed 1000, all this information is saved in the `Emr_TestResultsMib.txt` file and the internal storage is erased.

Commands Execution

CLI and XML are the primary ways to access ViewCheck capabilities on the PCIE-7207 card. These mechanisms allow you to perform activities such as start, stop, and query on the ViewCheck software. Using CLI and XML, you can also set the parameters for monitoring.

ViewCheck commands are classified as:

- Generic commands
- Test management and control commands
- Monitoring commands

3.1 Generic Commands

This section describes the generic commands of ViewCheck service.

3.1.1 config-watch-dog

`config-watch-dog` commands is used to configure action for the software watchdog.

Syntax for CLI

```
config-watch-dog action <string>
```

Table 3-1 Possible Actions

Action Name	Description
reset	Resets the MCP.
powercycle	Power cycle the MCP.
powerdown	Power down the CPU.
none	No action.

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDiagnostic</Class>
```

```

<Name>{mcpVirExecEnv MOID}</Name>
</Object>
<Method>configWatchDog</Method>
<Argument>
<Name>action</Name>
<Value>{action String}</Value>
</Argument>
</InvokeMethod>
</Script>]]]]>

```

3.1.2 configure-error-strings

configure-error-strings command allows adding user-defined kernel error strings to the diagnostics database.

Syntax for CLI

```
configure-error-strings errorStrings <string>
```

The following table provides the configure-error-strings command arguments.

Table 3-2 Parameter of configure-error-strings

Argument	Data Type	Description
string	String	Kernel error string to be added to the diagnostics database.

NOTICE

Make sure that the string that you enter must be more than one word.

Syntax for XML

```

<Script>
<InvokeMethod>
<Object>
<Class>mcpDiagnostic</Class>
<Name>{mcpVirExecEnv MOID}</Name>
</Object>
<Method>configureErrStrings</Method>

```

```
<Argument>
<Name>errorStrings</Name>
<Value>{error String}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>
```

3.1.3 configure-log-path

configure-log-path command allows you to configure the location of diagnostics logs. You can also specify the maximum limit for log size. Once the log size reaches the user-defined limit, a trap is sent to you. Specifying log size is optional and by default, its value is 1GB.

Syntax for CLI

```
configure-log-path logpath <PATH> logSize <size>
```

The following table provides the deviceCategory command arguments.

Table 3-3 Parameter of configure-log-path

Argument	Data Type	Description
logpath	String	Location of the log files where diagnostics should place.
size	String	Optional parameter. Size of the log. For example, 10M, 2G, 100K, 1048576.

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDiagnostic</Class>
<Name>{mcpVirExecEnv MOID}</Name>
</Object>
<Method>configureLogPath</Method>
<Argument>
<Name>logpath</Name>
```

```
<Value>{logPath String}</Value>  
</Argument>  
</InvokeMethod>  
</Script>]]>]]>
```

3.1.4 mcpDeviceCategory

mcpDeviceCategory command allows to configure the available device categories.

Syntax

```
mcpDeviceCategory <Dev category>
```

Expected Output

The CLI prompt will show the device category that you have selected.

NOTICE

This command is valid only in CLI.

3.1.5 hw-inventory-list

hw-inventory-list command provides the detailed information of hardware components available on the PCIE-7207 card. The command displays the Hardware Type, Vendor ID, Major Number, Minor Number and any other associated details that are identified by the ViewCheck.

Syntax for CLI

```
hw-inventory-list
```

Syntax for XML

```
<Script>  
<InvokeMethod>  
<Object>  
<Class>mcpDiagnostic</Class>  
<Name>{mcpVirExecEnv MOID}</Name>
```



```

</Object>
<Method>hwInventoryList</Method>
</InvokeMethod>
</Script>]]>]]>

```

3.1.6 list-all-tests

`list-all-tests` command lists all the tests available on the MCP.

Syntax for CLI

```
list-all-tests
```

Syntax for XML

```

<Script>
<InvokeMethod>
<Object>
<Class>mcpDiagnostic</Class>
<Name>{mcpVirExecEnv MOID}</Name>
</Object>
<Method>showListAllTests</Method>
</InvokeMethod>
</Script>]]>]]>

```

3.1.7 purge-all-log

`purge-all-log` command allows to zip all log files and store away.

Syntax for CLI

```
purge-all-log
```

Syntax for XML

```

<Script>
<InvokeMethod>
<Object>
<Class>mcpDiagnostic</Class>
<Name>{mcpVirExecEnv MOID}</Name>

```

```
</Object>
<Method>purgeAllLog</Method>
</InvokeMethod>
</Script>]]>]]>
```

3.1.8 set-log-level

`set-log-level` command sets the current log level of ViewCheck to value 'X'. This is an internal debug command, used mainly for generating detailed debug log information. The valid log level values are listed in the following table.

Table 3-4 Parameters of set-log-level Values

Values	Description
1-Info	All logs are logged. Even functions like entry and exit are also logged.
2- Normal	Details of function flows are logged.
3- Critical	High level errors are logged.

Syntax for CLI

```
set-log-level logLevelValue <x>
```

The following table lists the `set-log-level` command argument.

Table 3-5 Parameters of set-log-level Arguments

Argument	Data Type	Description
x	Integer	Possible values are 3, 2, 1 (3-Critical, 2-Normal, 1-Info).

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDiagnostic</Class>
<Name>{mcpVirExecEnv MOID}</Name>
</Object>
```

```

<Method>setLogLevel</Method>
<Argument>
<Name>logLevelValue</Name>
<Value>{Log Level Value}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>

```

3.1.9 show-all-monitors

`show-all-monitors` command lists all the monitors available on the PCIE-7207 card.

Syntax for CLI

```
show-all-monitors
```

Syntax for XML

```

<Script>
<InvokeMethod>
<Object>
<Class>mcpDiagnostic</Class>
<Name>{mcpVirExecEnv MOID}</Name>
</Object>
<Method>showAllMonitors</Method>
</InvokeMethod>
</Script>]]>]]>

```

3.1.10 show-all-test-status

`show-all-test-status` command allows you to view the current status of all tests available on the PCIE-7207 card.

Syntax for CLI

```
show-all-test-status
```

Syntax for XML

```

<Script>
<InvokeMethod>
<Object>
<Class>mcpDiagnostic</Class>

```

```
<Name>{mcpVirExecEnv MOID}</Name>  
</Object>  
<Method>showAllTestStatus</Method>  
</InvokeMethod>  
</Script>]]>]]>
```

3.1.11 show-diag-scan-result

show-diag-scan-result command shows result of the last diag-scan command.

Syntax for CLI

```
show-diag-scan-result
```

Syntax for XML

```
<Script>  
<InvokeMethod>  
<Object>  
<Class>mcpDiagnostic</Class>  
<Name>{mcpVirExecEnv MOID}</Name>  
</Object>  
<Method>showDiagScanResult</Method>  
</InvokeMethod>  
</Script>]]>]]>
```

3.1.12 show-log-content

show-log-content command displays the content of the mentioned log file.

Syntax for CLI

```
show-log-content logfileName <logfile Name>
```

The following table provides the show-log-content command arguments.

Table 3-6 Parameters of show-log-content

Argument	Data Type	Description
logfile Name	String	Name of the log file to be displayed.

Syntax for XML

```

<Script>
<InvokeMethod>
<Object>
<Class>mcpDiagnostic</Class>
<Name>{mcpVirExecEnv MOID}</Name>
</Object>
<Method>showLogContent</Method>
<Argument>
<Name>logFileName</Name>
<Value>{logfilename String}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>

```

3.1.13 show-heart-beat

show-heart-beat command displays the status of Heartbeat for all the registered services and applications.

Syntax for CLI

```
show-heart-beat
```

Syntax for XML

```

<Script>
<InvokeMethod>
<Object>
<Class>mcpDiagnostic</Class>
<Name>{mcpVirExecEnv MOID}</Name>
</Object>
<Method>showHeartBeat</Method>
</InvokeMethod>
</Script>]]>]]>

```

3.1.14 show-watch-dog

show-watch-dog command displays the current configuration of watchdog.

Syntax for CLI

```
show-watch-dog
```

Syntax for XML

```
<Script>  
<InvokeMethod>  
<Object>  
<Class>mcpDiagnostic</Class>  
<Name>{mcpVirExecEnv MOID}</Name>  
</Object>  
<Method>showWatchDog</Method>  
</InvokeMethod>  
</Script>]]>]]>
```

3.1.15 start-watch-dog

start-watch-dog command starts the watchdog.

Syntax for CLI

```
start-watch-dog
```

Syntax for XML

```
<Script>  
<InvokeMethod>  
<Object>  
<Class>mcpDiagnostic</Class>  
<Name>{mcpVirExecEnv MOID}</Name>  
</Object>  
<Method>startWatchDog</Method>  
</InvokeMethod>  
</Script>]]>]]>
```

3.1.16 show-log-info

show-log-info command provides the details of the various log files along with the diagnostics data.

Syntax for CLI

```
show-log-info
```

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDiagnostic</Class>
<Name>{mcpVirExecEnv MOID}</Name>
</Object>
<Method>showLogInfo</Method>
</InvokeMethod>
</Script>]]>]]>
```

3.1.17 show-log-level

show-log-level command displays the current logging level of ViewCheck.

Syntax for CLI

```
show-log-level
```

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDiagnostic</Class>
<Name>{mcpVirExecEnv MOID}</Name>
</Object>
<Method>showLogLevel</Method>
</InvokeMethod>
</Script>]]>]]>]
```

3.1.18 show-running-mode

show-running-mode command displays the running mode of the ViewCheck (INSM or OOSD).

Syntax for CLI

```
show-running-mode
```

Syntax for XML

```
<Script>  
<InvokeMethod>  
<Object>  
<Class>mcpDiagnostic</Class>  
<Name>{mcpVirExecEnv MOID}</Name>  
</Object>  
<Method>showRunningMode</Method>  
</InvokeMethod>  
</Script>]]>]]>
```

3.1.19 show-system-error-log

show-system-error-log command displays the kernel critical and error messages captured by ViewCheck application.

Syntax for CLI

```
show-system-error-log
```

Syntax for XML

```
<Script>  
<InvokeMethod>  
<Object>  
<Class>mcpDiagnostic</Class>  
<Name>{mcpVirExecEnv MOID}</Name>  
</Object>  
<Method>showSystemErrorLog</Method>  
</InvokeMethod>  
</Script>]]>]]>
```

3.1.20 start-all-monitors

start-all-monitors command starts all the monitors of the mentioned device category.

Syntax for CLI

```
start-all-monitors device-category <Dev category>
```


The following table provides the `start-all-monitors` command arguments.

Table 3-7 Parameters of start-all-monitors

Argument	Data Type	Description
Dev category	String	Name of the device category.

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDiagnostic</Class>
<Name>{mcpVirExecEnv MOID}</Name>
</Object>
<Method>startAllMonitors</Method>
<Argument>
<Name>deviceCategory</Name>
<Value>{Dev category String}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>
```

3.1.21 start-diag-scan

`start-diag-scan` starts the diag-scan on the specified device categories.

Syntax for CLI

```
start-diag-scan deviceCategory <Dev category-1>, ..., <Dev category-N>
Iterations <Itr-num> haltOnError <halt-string> timeout <timeout-
val>
```

Iterations, haltOnError, and timeout are optional arguments.

The following table provides the `startDiagScan` command arguments.

Table 3-8 Parameters of start-diag-scan

Argument	Data Type	Description
Dev category-N	String	Name of the device category. You can specify multiple device category using comma ',' in between.
Itr-num	Integer	Enter the number of times that diag-scan has to run. By default, value is "1". Maximum number of iterations that user can specify is 1000.
halt-string	String	Type "Yes" or "No". <code>haltOnError</code> specifies whether to continue or stop with test case execution on the occurrence of any error. By default, value is "No".
timeout-val	Integer	Enter the maximum time period to be taken by each test to execute. Note: If the value is '0', it will overwrite the factory configured default time-out value of all tests that are invoked as part of diagnostics scan. If no value is specified, each test runs for default time-out value. The defined time-out value varies from test to test and also depends on hardware.

Syntax for XML

```

<Script>
<InvokeMethod>
<Object>
<Class>mcpDiagnostic</Class>
<Name>{mcpVirExecEnv MOID}</Name>
</Object>
<Method>startDiagScan</Method>
<Argument>
<Name>deviceCategory</Name>
<Value>{Dev category String}</Value>
</Argument>
</InvokeMethod>
</Script>]]]]>
    
```

3.1.22 stop-all-monitors

stop-all-monitors command stops all the monitors of the mentioned device category.

Syntax for CLI

```
stop-all-monitors device-category <Dev category>
```

The following table provides the stopAllMonitors command arguments.

Table 3-9 Parameters of stop-all-monitors

Argument	Data Type	Description
Dev category	String	Name of the device category.

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDiagnostic</Class>
<Name>{mcpVirExecEnv MOID}</Name>
</Object>
<Method>stopAllMonitors</Method>
<Argument>
<Name>deviceCategory</Name>
<Value>{Dev category String}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>
```

3.1.23 stop-diag-scan

stop-diag-scan stops the currently running diag-scan command.

Syntax for CLI

```
stop-diag-scan
```

Syntax for XML

```
<Script>  
<InvokeMethod>  
<Object>  
<Class>mcpDiagnostic</Class>  
<Name>{mcpVirExecEnv MOID}</Name>  
</Object>  
<Method>stopDiagScan</Method>  
</InvokeMethod>  
</Script>]]>]]>
```

3.1.24 stop-watch-dog

stop-watch-dog stops the currently running watchdog command.

Syntax for CLI

```
stop-watch-dog
```

Syntax for XML

```
<Script>  
<InvokeMethod>  
<Object>  
<Class>mcpDiagnostic</Class>  
<Name>{mcpVirExecEnv MOID}</Name>  
</Object>  
<Method>stopWatchDog</Method>  
</InvokeMethod>  
</Script>]]>]]>
```

3.1.25 switch-mode

switch-mode command allows to switch ViewCheck from OOS mode to InService mode and vice versa.

NOTICE

Do not execute OOSD tests under OOS mode when user applications are still running.

Syntax for CLI

```
switch-mode modeVal <x>
```

The following table lists the `switch-mode` command arguments.

Table 3-10 switch-mode Arguments

Argument	Data Type	Description
X	string	Possible values are <code>inasm</code> and <code>oosd</code> . Parameter given for the command is case-insensitive.

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDiagnostic</Class>
<Name>{mcpVirExecEnv MOID}</Name>
</Object>
<Method>switchMode</Method>
<Argument>
<Name>modeVal</Name>
<Value>{Mode Value}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>
```

3.1.26 show-version

`show-version` command displays the RPM versions of all the ViewCheck packages installed.

Syntax for CLI

```
showVersion
```

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDiagnostic</Class>
```

```
<Name>{mcpVirExecEnv MOID}</Name>
</Object>
<Method>showVersion</Method>
</InvokeMethod>
</Script>]]>]]>
```

3.2 Test Management and Control Commands

This section describes the CLI commands used for test management and control of diagnostics tests.

NOTICE

Execute all test management and control commands only after entering a specific device category.

3.2.1 start-test

`start-test` command allows the user to start a particular diagnostics test.

Syntax for CLI

```
start-test testId <Test ID> deviceInstance <Dev Instance> arguments
-t <timeout-val> -Iterations <Itr-num> -Halt-onerror <halt-string>
```

Iterations, haltOnError, and timeout are optional arguments.

The following table lists the `start-test` command arguments.

Table 3-11 Parameters of start-test

Argument	Data Type	Description
Test ID	Integer	Type the unique ID of a particular test that you want to start.
Dev Instance	Enum	Run <code>show-test-help</code> command to get the list of device instances applicable for the test.

Table 3-11 Parameters of start-test (continued)

Argument	Data Type	Description
timeout-val	Integer	Time out value for the test. Note: If the value is '0', it will overwrite the factory configured default time-out value of the specified test ID. That is, the test runs until the test execution is complete. If no value is specified, each test runs for default time-out value. The defined time-out value varies from test to test and also depends on hardware.
ltr-num	Integer	Number of iterations of the test.
halt-string	String	"Yes" or "No" value. If "Yes", the test halts on error. If "No", the test does not halts.

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{Dev Cat}</Name>
</Object>
<Method>startTest</Method>
<Argument>
<Name>deviceInstance</Name>
<Value>{Device Instance}</Value>
</Argument>
<Argument>
<Name>testId</Name>
<Value>{Test ID}</Value>
</Argument>
<Argument>
<Name>arguments</Name>
<Value>{Parameters}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>
```

3.2.2 show-test-help

`show-test-help` command provides brief information on how to use a particular test. This command provides information on how to start, stop, and query the specified test ID.

Syntax for CLI

```
show-test-help testId <Test ID>
```

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{dev Cat}</Name>
</Object>
<Method>showTestHelp</Method>
<Argument>
<Name>testId</Name>
<Value>{Test ID}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>
```

The following table lists the `show-test-help` command arguments.

Table 3-12 Parameters of show-test-help

Argument	Data Type	Description
Test ID	Integer	Type the unique ID of a particular test to get the details of it.

3.2.3 list-tests

`list-tests` command provides a supported list of diagnostics tests on the PCIE-7207 card with associated test IDs.

Syntax for CLI

```
list-tests
```


Syntax for XML

```

<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{dev Cat}</Name>
</Object>
<Method>showTestList</Method>
</InvokeMethod>
</Script>]]>]]>

```

3.2.4 show-test-status

`show-test-status` command allows you to view the status of a particular test. The status of a test can be In Progress, Test Execution Completed, Test Stopped, or Test Timed Out.

Syntax for CLI

```
show-test-status testId <Test ID> deviceInstance <Dev Instance>
```

Syntax for XML

```

<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{Dev Cat}</Name>
</Object>
<Method>showTestStatus</Method>
<Argument>
<Name>deviceInstance</Name>
<Value>{Device Instance}</Value>
</Argument>
<Argument>
<Name>testId</Name>
<Value>{Test ID}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>

```

The following table lists the `show-test-status` command arguments.

Table 3-13 Parameters of show-test-status

Argument	Data Type	Description
Test ID	Integer	Type the unique ID of a particular test that you want to view the status.
Dev Instance	Enum	Use <code>list-device-instances</code> CLI command to get the equivalent enumerated value of supported Device Instance.

3.2.5 show-test-result

`show-test-result` command allows you to view the latest result of a particular test. This command displays the start and end time of the test, the test status such as Passed, Failed, Aborted, Timed Out, and additional test arguments.

Syntax for CLI

```
show-test-result testId <Test ID> deviceInstance <Dev Instance>
```

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{Dev Cat}</Name>
</Object>
<Method>showTestResult</Method>
<Argument>
<Name>deviceInstance</Name>
<Value>{Device Instance}</Value>
</Argument>
<Argument>
<Name>testId</Name>
```

```

<Value>{Test ID}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>

```

Table 3-14 Parameters of show-test-result

Argument	Data Type	Description
Test ID	Integer	Type the unique ID of a particular test that you want to view the latest result.
Dev Instance	Enum	Use <code>list-device-instances</code> command to get the equivalent enumerated value of supported device instance.

3.2.6 stop-test

`stop-test` command allows you to stop or cancel any running diagnostics test.

Syntax for CLI

```
stop-test testId <Test ID> deviceInstance <Dev Instance>
```

Syntax for XML

```

<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{Dev Cat}</Name>
</Object>
<Method>stopTest</Method>
<Argument>
<Name>deviceInstance</Name>
<Value>{Device Instance}</Value>
</Argument>
<Argument>
<Name>testId</Name>

```

```
<Value>{Test ID}</Value>  
</Argument>  
</InvokeMethod>  
</Script>]]>]]>
```

The following table provides the `stop-test` command arguments.

Table 3-15 Parameters of stop-test

Argument	Data Type	Description
Test ID	Integer	Type the unique ID of the Test that you want to stop
Dev Instance	Enum	Use <code>list-device-instances</code> CLI command to get the equivalent enumerated value of supported Device Instance

3.3 Monitoring Commands

The following list of commands are used for management of the In-service Monitoring functionality of ViewCheck.

NOTICE

Execute all commands related to monitors only after entering the specific device category mode.

3.3.1 show-monitor-id

`show-monitor-id` command displays the list of parameters that are monitored using the In-service Diagnostic. This command also displays default monitor ID values of the parameters.

Syntax for CLI

```
show-monitor-id
```

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{Dev Cat}</Name>
</Object>
<Method>showMonitorId</Method>
</InvokeMethod>
</Script>]]]]>
```

NOTICE
<p>This command displays a list of monitors, if supported, only in that particular device category.</p>

3.3.2 show-poll-interval

show-poll-interval command displays the list of default Poll intervals that is associated with the monitors. The Poll interval values are in seconds.

Syntax for CLI

```
show-poll-interval monitorId <Monitor ID> deviceInstance <Dev Instance>
```

The following table provides the show-poll-interval command arguments.

Table 3-16 Parameters of show-poll-interval

Argument	Data Type	Description
Monitor ID	Integer	Type the unique ID of the monitor of which you want to view its default Poll interval values, if any.
Dev Instance	Enum	Use list-device-instances command to get the equivalent enumerated value of supported device instance.

Syntax for XML

```

<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{Dev Cat}</Name>
</Object>
<Method>showPollInterval</Method>
<Argument>
<Name>deviceInstance</Name>
<Value>{Device Instance}</Value>
</Argument>
<Argument>
<Name>monitorId</Name>
<Value>{Monitor ID}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>

```

3.3.3 show-lower-threshold-info

show-lower-threshold-info command displays lower threshold information that is applicable to a monitor.

Syntax for CLI

```

show-lower-threshold-info monitorId <Monitor ID> deviceInstance
<Dev Instance>

```

The following table provides the show-lower-threshold-info command arguments.

Table 3-17 Parameters of show-lower-threshold-info

Argument	Data Type	Description
Monitor ID	Integer	Type the unique ID of the monitor of which you want to view its lower threshold value.
Dev Instance	Enum	Use list-device-instances CLI command to get the equivalent enumerated value of supported device instance.

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{Dev Cat}</Name>
</Object>
<Method>showLowerThresholdInfo</Method>
<Argument>
<Name>deviceInstance</Name>
<Value>{Device Instance}</Value>
</Argument>
<Argument>
<Name>monitorId</Name>
<Value>{Monitor ID}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>
```

3.3.4 show-upper-threshold-info

show-upper-threshold-info command displays upper threshold information that is applicable to a monitor.

Syntax for CLI

```
show-upper-threshold-info monitorId <Monitor ID> deviceInstance
<Dev Instance>
```

The following table provides the show-upper-threshold-info command arguments.

Table 3-18 Parameters of show-upper-threshold-info

Argument	Data Type	Description
Monitor ID	Integer	Type the unique ID of the monitor of which you want to view its upper threshold value.
Dev Instance	Enum	Use list-device-instances command to get the equivalent enumerated value of supported device instance.

Syntax for XML

```

<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{Dev Cat}</Name>
</Object>
<Method>showUpperThresholdInfo</Method>
<Argument>
<Name>deviceInstance</Name>
<Value>{Device Instance}</Value>
</Argument>
<Argument>
<Name>monitorId</Name>
<Value>{Monitor ID}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>

```

3.3.5 start-monitor

`start-monitor` command allows you to trigger a specific monitor to start monitoring, if it is not already initiated by default.

Syntax for CLI

```
start-monitor monitorId <Monitor ID> deviceInstance <Dev Instance>
```

The following table provides the `start-monitor` command arguments.

Table 3-19 Parameters of start-monitor

Argument	Data Type	Description
Monitor ID	Integer	Type the unique ID of the monitor that you want to start.
Dev Instance	Enum	Use <code>list-device-instances</code> command to get the equivalent enumerated value of supported device instance.

Syntax for XML

```

<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{Dev Cat}</Name>
</Object>
<Method>startMonitor</Method>
<Argument>
<Name>deviceInstance</Name>
<Value>{Device Instance}</Value>
</Argument>
<Argument>
<Name>monitorId</Name>
<Value>{Monitor ID}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>

```

3.3.6 set-lower-threshold-info

`set-lower-threshold-info` command is used to set the lower threshold value for a particular monitor based on which the monitor performs.

Syntax for CLI

```

set-lower-threshold-info monitorId < Monitor ID> deviceInstance
<Dev Instance> lowerThreshold <Threshold Value>

```

The following table provides the `set-lower-threshold-info` command arguments.

Table 3-20 Parameters of set-lower-threshold-info

Argument	Data Type	Description
Monitor ID	Integer	Type the unique ID of the monitor for which you want to set the lower threshold value.
Dev Instance	Enum	Use <code>list-device-instances</code> command to get the equivalent enumerated value of supported device instance.

Table 3-20 Parameters of set-lower-threshold-info (continued)

Argument	Data Type	Description
Threshold Value	Integer	Type a threshold value for the specified monitor ID.

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{Dev Cat}</Name>
</Object>
<Method>setLowerThresholdInfo</Method>
<Argument>
<Name>deviceInstance</Name>
<Value>{Device Instance}</Value>
</Argument>
<Argument>
<Name>monitorId</Name>
<Value>{Monitor ID}</Value>
</Argument>
<Argument>
<Name>lowerThreshold</Name>
<Value>{Threshold Value}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>
```

3.3.7 set-upper-threshold-info

set-upper-threshold-info command is used to set the upper threshold value for a particular monitor based on which the monitor performs.

Syntax for CLI

```
set-upper-threshold-info monitorId < Monitor ID> deviceInstance
<Dev Instance> upperThreshold <Threshold Value>
```

The following table provides the `set-upper-threshold-info` command arguments.

Table 3-21 Parameters of set-upper-threshold-info

Argument	Data Type	Description
Monitor ID	Integer	Type the unique ID of the monitor for which you want to set the upper threshold value.
Dev Instance	Enum	Use <code>list-device-instances</code> command to get the equivalent enumerated value of supported device instance.
Threshold Value	Integer	Type a threshold value for the specified monitor ID.

Syntax for XML

```

<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{Dev Cat}</Name>
</Object>
<Method>setUpperThresholdInfo</Method>
<Argument>
<Name>deviceInstance</Name>
<Value>{Device Instance}</Value>
</Argument>
<Argument>
<Name>monitorId</Name>
<Value>{Monitor ID}</Value>
</Argument>
<Argument>
<Name>upperThreshold</Name>
<Value>{Threshold Value}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>
    
```

3.3.8 set-threshold-default

set-threshold-default command is used to reset the threshold value to default value of a particular monitor.

Syntax for CLI

```
set-threshold-default monitorId <Monitor ID> deviceInstance <Dev Instance>
```

The following table provides the set-threshold-default command arguments.

Table 3-22 Parameters of set-threshold-default

Argument	Data Type	Description
Monitor ID	Integer	Type the unique ID of the monitor for which you want to reset its threshold value to default value.
Dev Instance	Enum	Use list-device-instances command to get the equivalent enumerated value of supported device instance.

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{Dev Cat}</Name>
</Object>
<Method>setThresholdDefault</Method>
<Argument>
<Name>deviceInstance</Name>
<Value>{Device Instance}</Value>
</Argument>
<Argument>
<Name>monitorId</Name>
<Value>{Monitor ID}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>
```

3.3.9 set-poll-interval

set-poll-interval command is used to set the Poll interval value for monitors.

Syntax for CLI

```
set-poll-interval monitorId <Monitor ID> deviceInstance <Dev Instance> pollInterval <PollIntervalValue>
```

The following table provides the set-poll-interval command arguments.

Table 3-23 Parameters of set-poll-interval

Argument	Data Type	Description
Monitor ID	Integer	Type the unique ID of the monitor for which you want to set the Poll interval value.
Dev Instance	Enum	Use list-device-instances command to get the equivalent enumerated value of supported device instance.
PollInterval Value	Integer	Type the Poll interval value for the specified monitor ID.

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{Dev Cat}</Name>
</Object>
<Method>setPollInterval</Method>
<Argument>
<Name>deviceInstance</Name>
<Value>{Device Instance}</Value>
</Argument>
<Argument>
<Name>monitorId</Name>
<Value>{Monitor ID}</Value>
</Argument>
<Argument>
<Name>pollInterval</Name>
```

```
<Value>{Poll Interval Value}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>
```

3.3.10 set-poll-interval-default

set-poll-interval-default command is used to reset the Poll interval value to default value of a particular monitor.

Syntax for CLI

```
set-poll-interval-default monitorId <Monitor ID> deviceInstance
<Dev Instance>
```

The following table provides the set-poll-interval-default command arguments.

Table 3-24 Parameters of set-poll-interval-default

Argument	Data Type	Description
Monitor ID	Integer	Type the unique ID of the monitor for which you want to reset the Poll interval value to default value.
Dev Instance	Enum	Use list-device-instances command to get the equivalent enumerated value of supported device instance.

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{Dev Cat}</Name>
</Object>
<Method>setPollIntervalDefault</Method>
<Argument>
<Name>deviceInstance</Name>
<Value>{Device Instance}</Value>
</Argument>
<Argument>
<Name>monitorId</Name>
```

```
<Value>{Monitor ID}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>
```

3.3.11 stop-monitor

stop-monitor command allows you to trigger a specific monitor to stop monitoring, which is already in service.

Syntax for CLI

```
stop-monitor monitorId <Monitor ID> deviceInstance <Dev Instance>
```

The following table provides the stop-monitor command arguments.

Table 3-25 Parameters of stop-monitor

Argument	Data Type	Description
Monitor ID	Integer	Type the unique ID of the monitor that you want to stop monitoring.
Dev Instance	Enum	Use list-device-instances command to get the equivalent enumerated value of supported device instance.

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{Dev Cat}</Name>
</Object>
<Method>stopMonitor</Method>
<Argument>
<Name>deviceInstance</Name>
<Value>{Device Instance}</Value>
</Argument>
<Argument>
<Name>monitorId</Name>
```

```
<Value>{Monitor ID}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>
```

3.3.12 set-rate

set-rate command sets the rate of change value for network monitors.

Syntax for CLI

```
set-rate monitorId <Monitor Id> deviceInstance <Dev Instance> rate
<Rate>
```

The following table provides the set-rate command arguments.

Table 3-26 Parameters of set-rate

Argument	Data Type	Description
Monitor ID	Integer	Type the unique ID of the monitor for which rate of change value is to be set.
Dev Instance	Enum	Use list-device-instances command to get the equivalent enumerated value of supported device instance.
Rate	Integer	Rate of change value.

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}- {Dev Cat}</Name>
</Object>
<Method>setRate</Method>
<Argument>
<Name>deviceInstance</Name>
<Value>{Device Instance}</Value>
</Argument>
<Argument>
<Name>monitorId</Name>
```



```
<Value>{Monitor ID}</Value>
</Argument>
<Argument>
<Name>rate</Name>
<Value>{Rate}</Value>
</Argument>
</InvokeMethod>
</Script>]]>]]>
```

3.3.13 show-rate

show-rate command shows the rate of change value of network monitors.

Syntax for CLI

```
show-rate monitorId <Monitor Id> deviceInstance <Dev Instance>
```

The following table provides the show-rate command arguments.

Table 3-27 Parameters of show-rate

Argument	Data Type	Description
Monitor ID	Integer	Type the unique ID of the monitor of which rate of change value is to be shown.
Dev Instance	Enum	Use list-device-instances CLI command to get the equivalent enumerated value of supported Device Instance.

Syntax for XML

```
<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{Dev Cat}</Name>
</Object>
<Method>showRate</Method>
<Argument>
<Name>deviceInstance</Name>
<Value>{Device Instance}</Value>
</Argument>
```

```
<Argument>  
<Name>monitorId</Name>  
<Value>{Monitor ID}</Value>  
</Argument>  
</InvokeMethod>  
</Script>]]>]]>
```

3.3.14 show-network-counter-log

`show-network-counter-log` command displays the list of network counters on devices that have crossed the maximum rate value. For more information on command, rate, see [show-rate on page 73](#).

Syntax for CLI

```
show-network-counter-log
```

Syntax for XML

```
<Script>  
<InvokeMethod>  
<Object>  
<Class>mcpDeviceCategory</Class>  
<Name>{mcpVirExecEnv MOID}-{Dev Cat}</Name>  
</Object>  
<Method>showNetworkCounterLog</Method>  
</InvokeMethod>  
</Script>]]>]]>
```

3.3.15 show-network-error-log

`show-network-error-log` command displays the list of network errors on devices that have crossed the maximum rate value. For more information on command, rate, refer [show-rate on page 73](#).

Syntax for CLI

```
show-networkerror-log
```

Syntax for XML

```

<Script>
<InvokeMethod>
<Object>
<Class>mcpDeviceCategory</Class>
<Name>{mcpVirExecEnv MOID}-{Dev cat}</Name>
</Object>
<Method>showNetworkErrorLog</Method>
</InvokeMethod>
</Script>]]>]]>

```

3.3.16 exit

exit command allows you to exit from the ViewCheck CLI.

Syntax for CLI

```
exit
```

Syntax for XML

```

<Script>
<Command name="Exit" />
</Script>]]>]]>

```

3.4 In-service Monitoring Specifics

The In-service Monitoring of ViewCheck provides functionality to:

- Monitor Hardware Device Status
- Monitor Device Critical Errors

3.4.1 Monitoring Hardware Device Status

In-service monitoring also periodically polls for status of the various devices on the PCIE-7207 card. They are:

- CPU Core Temperature Status
- Network Devices Counter Statistics

- Network Devices Error Statistics
- I2C Sensors
- CPU and CPU Core Usage Monitor
- Memory Usage Monitor
- Software Watchdog
- Heartbeat

Each of the above is identified by a unique Monitor ID. For more information about Monitor ID, see [Monitor ID on page 15](#).

The default Poll Interval for monitoring each of the Monitor ID is set to 10 seconds. Commands are provided to edit the default settings.

For more information on the commands provided for In-service Monitoring, see [Monitoring Commands on page 60](#).

The following table provides list of monitor IDs and device instances for each of the monitors being monitored on the PCIE-7207 card.

Table 3-28 Monitors

Monitor Description	Monitor ID	Valid Device Instances	Remarks
CPU Core Temperatures	1000	Core 0 to Core 1 (0 to 1)	Temperature of the 2 cores available on the PCIE-7207 card are monitored individually and reported if crosses a set threshold value.
CPU and Core Usage	1001	CPU, Core0 to Core3	CPU and its core usage is monitored.
Network Errors	1020	eth0	Monitors the various error counters for each of the network device instances and provides an error counter exceeds the rate of change.
Network Counters	1021	eth0	Monitors the various counters for each of the network device instances

Table 3-28 Monitors (continued)

Monitor Description	Monitor ID	Valid Device Instances	Remarks
Memory Usage	1030	MEMCONTLR	Memory usage is monitored.
I2C Sensor	1040	i2c0	I2C sensor power is monitored.
	1041	i2c1 to i2c2	I2C sensors SPD#1 and SPD#2 are monitored.
Watchdog	NA	NA	Monitors the health of MCP.
Heartbeat	NA	NA	Monitors registered applications and services on MCP.

3.4.2 Monitoring Device Critical Errors

Under Linux OS, the device drivers log abnormal behavior and potential errors occurring in the hardware device with `KERN_ERR` or `KERN_CRIT` category. These notifications are considered as potential errors as they could manifest into latent faults in the live system.

As part of monitoring the device critical errors, all such kernel critical and kernel error notifications have been extracted from the CentOS7.1 driver sources and represented in the form of a database.

The In-service monitoring module of ViewCheck looks for the occurrence of these notifications and on detection sends a notification to XML.

The device errors are captured and are identified uniquely with their ERROR IDs. For definition of ERROR ID, see [Error ID on page 17](#).

3.4.3 Software Watchdog

Software Watchdog feature allows you to check the health of MCP. If MCP becomes unresponsive, the watchdog timer will expire and a configured action will be taken by the watchdog timer. The watchdog supports following actions.

- Reset (default action)
- Shutdown
- Power Cycle
- None

3.4.3.1 Watchdog Related Commands

ViewCheck provides commands to manage the watchdog for each MCP. For more information on watchdog related commands see the table below.

Table 3-29 Watchdog Related Commands

Commands	Description
<code>conf-watch-dog</code>	This command is used to configure action. These actions are taken if watchdog timer expires.
<code>show-watch-dog</code>	This command is used to see the current configuration and status of watchdog.
<code>start-watch-dog</code>	This command is used to start watchdog, if it is in stopped state.
<code>stop-watch-dog</code>	This command is used to stop watchdog, if it is in ongoing (start) state.

3.4.4 Monitoring Heartbeat

The heartbeat feature allows user applications or any other services running on MCP to register for regular health check with ViewCheck core. The hregister tool available at the following location is used to Register and De-register.

```
/opt/diagnostics/tools/diagtestutils/hregister <request_type>
<service_name> <health_check_script_path>
```

<request_type>:

The type of heartbeat request. Either '1' for register or '0' for de-register.

<service_name>:

The name of application or service requesting for heartbeat check. The length of the name should not be more than 16 characters.

<health_check_script_path>:

This script should be provided by the user for the application or service which is being registered. ViewCheck core calls this script at regular intervals to check if health check passes or fails. The script should perform all the necessary steps to check for health of the application or service and return the appropriate exit code as below.

Exit 0: If health check status passes

Exit 1: If health check status fails

Exit 2: If health check status is unknown

Registration Example:

```
/opt/diagnostics/tools/diagtestutils/hbregister "1" "MYAPP"
"/opt/bladeservices/scripts/healthcheck.sh"
```

De-registration Example:

```
/opt/diagnostics/tools/diagtestutils/hbregister "0" "MYAPP"
```

3.4.4.1 show-heartbeat-info

`show-heartbeat-info` command is used to list all the applications or services registered for health check. It also shows the current status of the health check such as Pass, Fail, Initialized, and Unknown status. A sample output of this command is provided below.

1. Service Name: MYAPP, Last heartbeat status: PASS
2. Service Name: MYSERVICE, Last heartbeat status: PASS

3.5 Out of Service Diagnostics Tests

Out of Service Diagnostics (OOSD) tests are used to monitor and manage the performance of the hardware components of a PCIe-7207 card. You can execute these tests only when PCIe-7207 card is offline (card is not providing any customer's applications and service).

3.5.1 CPU

- **CpuTempTest:** Reads the processor core(s) temperature levels and ensures that all cores are operating in normal levels.
- **CpuBurnTest:** This test constantly cycles FPU intensive functions. The resultant calculations are constantly checked for data integrity. If the test detects erroneous data, the test fails.

- **CacheSizeTest:** Verifies L1, L2, and L3 Cache sizes of CPU(s) on the board.
- **CpuBenchMark:** Tests different arithmetic operations and gives the results.

3.5.2 Memory

- **MemCntlrTest:** Randomly writes to areas of memory, then reads the memory back to ensure the written values remain unchanged.
- **RandomMemoryTest:** Performs stress testing on the memory subsystem. This test is effective in finding intermittent and non-deterministic faults. The problems in other hardware areas such as overheating CPU, out-of-specification power supply, and so on can cause memory faults.
- **MemBandwidth:** Measures the ability to copy, read, and write data over a varying set of sizes.
- **MemLatency:** Measures the time taken by the memory to respond with the data for read-request.

3.5.3 OS

- **MemSepTest:** Ensures that user space programs cannot read and write to areas of memory utilized by items such as Video RAM and kernel code.
- **SupervisorInstrTest:** Ensures that the enforcement of the property that privileged instructions should only be in supervisor mode is still in effect. The set of privileged instructions tested to confirm this is architecture dependent.
- **DmesgCheckTest:** Checks for user-given keywords in the kernel dmesg logs.

3.5.4 PCI

- **PCIScanTest:** Enumerates all active PCI devices in the PCIE-7207 card and ensures card's default configuration is active. This test need to be executed under supervisory mode.

3.5.5 Network

- **EthLinkTest:** Verifies Ethernet device (for example, link status (active/inactive) and also captures various statistics of Ethernet device.
- **FloodPingTest:** Uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP_ECHO_RESPONSE from a host or gateway.
- **EthStatsTest:** Tests basic network packet consistency, that is, checks network Rx, Tx errors, generates a warning if errors are less than or equal to 100 and marks the test as fail, if number of errors are more than 100.
- **NetworkCntlrTest:** Verifies random data transmitted and also the data received for each configured network device. It verifies only Ethernet and token ring devices that are configured and active. The asynchronous devices are not verified.
- **NetworkAdapterTest:** Executes adapter selftest (BIST) on the specified ethernet device.
- **NetworkThroughputServ:** Starts the server for network throughput testing.
- **NetworkThroughput:** Tests the network throughput.
- **NetworkTxBenchMark:** Checks the transmission rate of interface.

3.5.6 RTC

- **RTCTest:** Checks for proper functioning of real time clock.

Related Documentation

4.1 Artesyn Embedded Technologies - Embedded Computing Documentation

The publications listed below are referenced in this manual. You can obtain electronic copies of Artesyn Embedded Technologies - Embedded Computing publications by contacting your local Artesyn sales office. For released products, you can also visit our Web site for the latest copies of our product documentation.

1. Go to www.artesyn.com/computing/support/product/technical-documentation.php.
2. Under **FILTER OPTIONS**, click the Document types drop-down list box to select the type of document you are looking for.
3. In the **Sea87rch** text box, type the product name and click **GO**.

Table 4-1 Artesyn Embedded Technologies - Embedded Computing Publications

Document Title	Publication Number
SharpStreamer PCIE-7207 Installation and Use	6806800T60
SSF-on-SharpStreamer-PCIE-7207-for-MaxCore-CLI	6806800U38
SSF for MaxCore™ MC3000 Platform XML Interface Guide	6806800T71
SSF for MaxCore™ MC3000 Platform Installation and Use	6806800T81



Artesyn Embedded Technologies, Artesyn and the Artesyn Embedded Technologies logo are trademarks and service marks of Artesyn Embedded Technologies, Inc. All other product or service names are the property of their respective owners.

© 2016 Artesyn Embedded Technologies, Inc.