

Hawkeye

Release 6.0.7

Advanced Admin User Guide

Notices

Copyright Notice

© Keysight Technologies 2016–2025

No part of this document may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Keysight Technologies, Inc. as governed by United States and international copyright laws.

Warranty

The material contained in this document is provided “as is,” and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Keysight disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Keysight shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Keysight and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

U.S. Government Rights

The Software is “commercial computer software,” as defined by Federal Acquisition Regulation (“FAR”) 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement (“DFARS”) 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly,

Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at <http://www.keysight.com/find/sweula>. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data. 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Contacting us

Keysight headquarters

1400 Fountaingrove Parkway
 Santa Rosa, CA 95403-1738
 Email address: support@keysight.com

Support

Location	Phone number	Local time
<i>Americas</i>		
US, Canada	1-888-829-5558	8h00 – 17h00
Brazil	0800-892-0522	8h00 – 17h00
Mexico	001-888-829-5558	8h00 – 17h00
Other	+1-719-273-6516	8h00 – 17h00
<i>EMEA</i>		
Belgium	0800-18686	8h30 – 17h30
Finland	0800-913-352	8h30 – 17h30
France	0800-917228	8h30 – 17h30
Germany	0800-0824099	8h30 – 17h30
India	1800-18-02552	8h30 – 17h30
Ireland	1800-949245	8h30 – 17h30
Israel	1-809-454975	8h30 – 17h30
Italy	0800-790571	8h30 – 17h30
Luxembourg	0800-25112	8h30 – 17h30
Netherlands	0800-022-9086	8h30 – 17h30

Romania	0213 015 699	8h30 – 17h30
Spain	800-654386	8h30 – 17h30
Sweden	0201-202266	8h30 – 17h30
United Kingdom	0800-0293882	8h30 – 17h30
<i>Asia and Australia</i>		
Australia	1-800-370-558	8h30 – 17h00
China Mainland	800-810-0005	8h30 – 17h30
	400-810-0005	8h30 – 17h30
Hong Kong	800-931-613	9h00 – 18h00
Japan	0120-421-621	9h00 – 17h30
Malaysia	1800-819 092	8h30 – 17h30
South Korea	080-770-0800	8h30 – 17h30
Singapore	800-101-3797	8h30 – 17h30
Taiwan	0800-699-880	9h00 – 18h00
Other	+65 6215 7600	8h30 – 17h30 (Singapore)

Table of Contents

Contacting us	3
Chapter 1 Advanced Administration Guide	7
MariaDB Database Management	8
Hawkeye Configuration Database Structure	8
Elasticsearch Management	9
OSS (Operations Support System) Integration	10
Backup and restore Hawkeye databases	11
Probe Management - Probe Busy	14
Hawkeye Folder Structure	16
Set up e-mails with Hawkeye server	16
SNMP Configuration and Verification	18
SNMP Traps	18
How to use SNMP Traps	18
Verify Hawkeye Server MIBs file can be interpreted by the User's SNMP server	19
Configure Hawkeye Server to send SNMP trap to a specific SNMP server	19
Configure your SNMP server to receive Hawkeye server originated SNMP traps	19
Enable SNMP traps to be triggered for endpoint scheduled tests	20
Verify SNMP trap successfully received and decoded by SNMP server	20
Advanced Troubleshooting – Debug SNMP packet routing issues	20
Rest API	21
REST examples	25
Server upgrade	32
Configure Time Zone and NTP	37
Configure Time Zone	37
Configure NTP Client	37
Configure Hawkeye as NTP Server	38

Configure UI Time Zone	39
Index	41

CHAPTER 1

Advanced Administration Guide

This section covers the advanced administration steps of Hawkeye server. You should have administrator access to the server hosting Hawkeye to be able to use this.

The Hawkeye system is currently based on the Debian 12 OS. The main hawkeye software is installed in /opt/hawkeye . The status of the assorted services can be shown by issuing the hawkeyeStatus command. They can then be restarted by issuing the hawkeyeServices command.

The associated config files, which allow you to fine tune and better integrate the hawkeye server are outlined in this chapter.

In this chapter:

MariaDB Database Management	8
Hawkeye Configuration Database Structure	8
Elasticsearch Management	9
OSS (Operations Support System) Integration	10
Backup and restore Hawkeye databases	11
Probe Management - Probe Busy	14
Hawkeye Folder Structure	16
Set up e-mails with Hawkeye server	16
SNMP Configuration and Verification	18
SNMP Traps	18
How to use SNMP Traps	18
Verify Hawkeye Server MIBs file can be interpreted by the User's SNMP server	19
Configure Hawkeye Server to send SNMP trap to a specific SNMP server	19
Configure your SNMP server to receive Hawkeye server originated SNMP traps	19
Enable SNMP traps to be triggered for endpoint scheduled tests	20
Verify SNMP trap successfully received and decoded by SNMP server	20
Advanced Troubleshooting – Debug SNMP packet routing issues	20
Rest API	21
REST examples	25
Server upgrade	32
Configure Time Zone and NTP	37
Configure Time Zone	37
Configure NTP Client	37

Configure Hawkeye as NTP Server	38
Configure UI Time Zone	39

MariaDB Database Management

Hawkeye Database

The tuning file for the Hawkeye Database is `/etc/mysql/mariadb.conf.d/99_tuning.cnf`. This file contains the tuning parameters for the server as well as defining where the MySQL server stores its data files.

For larger installations, you should consult Keysight Support to determine if any of the values here should be modified to accommodate the requirements of the specific installation.

Example of configuration for a server with 16G RAM:

```
#####
# mariaDB System optimizations can be made utilizing this file
# Edit this file using your favorite text editor.
# vi /etc/mysql/mariadb.conf.d/99_tuning.cnf
#
# Uncomment the settings you wish to set and update the associated values.
#
#####
[mariadb]

#####
# innodb_buffer_pool_size=1G This should be set to be 25% of the
# total system RAM. Example for a system with 16GB of RAM you should
# allocate 4G to the innodb_buffer_pool_size
#####

innodb_buffer_pool_size=4G

#####
# Default max-connections=151 This defines the maximum number of
# concurrent connections opened against MySQL. In the case of numerous
# endpoints/tests per hour (above 1000 tests per hour), Keysight recommends
# to increase this number to 1000
#####
```

`max-connections=1000`

MariaDB runs as a service:

- Do not attempt to terminate mariadb service as this can corrupt the database.
- Use `hawkeyeServices` command to restart MariaDB service.

Hawkeye Configuration Database Structure

Hawkeye utilizes MariaDB to store:

- System information and administration management
- Test execution Queue Management

Elasticsearch Management

Elasticsearch advanced configuration for performance tuning

The tuning file for Elasticsearch is `/opt/hawkeye/conf/elastic/jvm.options.d/jvm_tuning.options`. This file contains the tuning parameters for Elasticsearch. For larger installations, it is recommended to consult Keysight support team to determine if any of the values must be modified to accommodate the requirements of the specific installation.

NOTE

Heap Size should never exceed 49% of total system RAM. This can lead to Memory Utilization Alarms and system instability. We recommend to implement 45% of total system RAM.

Example of increment heap size from default 1GB to 4GB:

```
[root@Hawkeye-Server ~]# cat /opt/hawkeye/conf/elastic/jvm.options.d/jvm_tuning.options
#####
-Xms4g
-Xmx4g
#####
```

To apply the new changes made, you must restart **HawkeyeElasticsearch.service**.

HawkeyeElasticsearch runs as a service on which other services depend. To properly restart the service, use the `hawkeyeServices` command.

OSS (Operations Support System) Integration

Hawkeye is designed to be an open framework for easy integration into service providers OSS.

It supports a range of northbound and southbound interfaces that are designed to :

- Export or collect key information and alarms generated by Hawkeye to third party system.
- Automate some tasks from third party tools (generate automatic tests for example).
- Generate customized actions.

Hawkeye SNMP Alarms

Based on test results Hawkeye can trigger an Hawkeye SNMP trap to a third party.

The mib is based on SNMPv2 and is easily imported into supervision systems.

Alarms are generated based on pass fail criteria generated by thresholds.

Hawkeye Email Notifications

Hawkeye can generate automatic emails to configurable recipients (configurable per alert) with alarms based on test results or thresholds.

Automatic emails can also be sent with content result of aggregated report – these are user defined and can be scheduled over time (every hour/day/week).

Hawkeye REST API

Hawkeye REST API allows you to work on the hawkeye server through scripts. This is especially useful for integration in OSS. REST API can be used to integrate the following:

- Manage user/group
- Create tests
- Access user trail
- Update status of probes connected to the system
- Manage Mesh
- Manage IxProbe/Tap/Bypass
- Create and execute test template
- View test results

Keysight also provides the Hawkeye Data Extraction Tool (HDET), which you can download from the support page as a sample set of scripts, which can be run outside of the Hawkeye server to export the test results and audit logs. HDET utilizes python to export data in .json or .csv format for ease of integration with other tools.

Backup and restore Hawkeye databases

Hawkeye Results Data

Hawkeye Results are stored in a NoSQL database for faster response times and to facilitate future expansion. Hawkeye Results data is backed up every night at midnight server time. System administrators are also presented with the option to create a backup of the Hawkeye Results data as part of a server update.

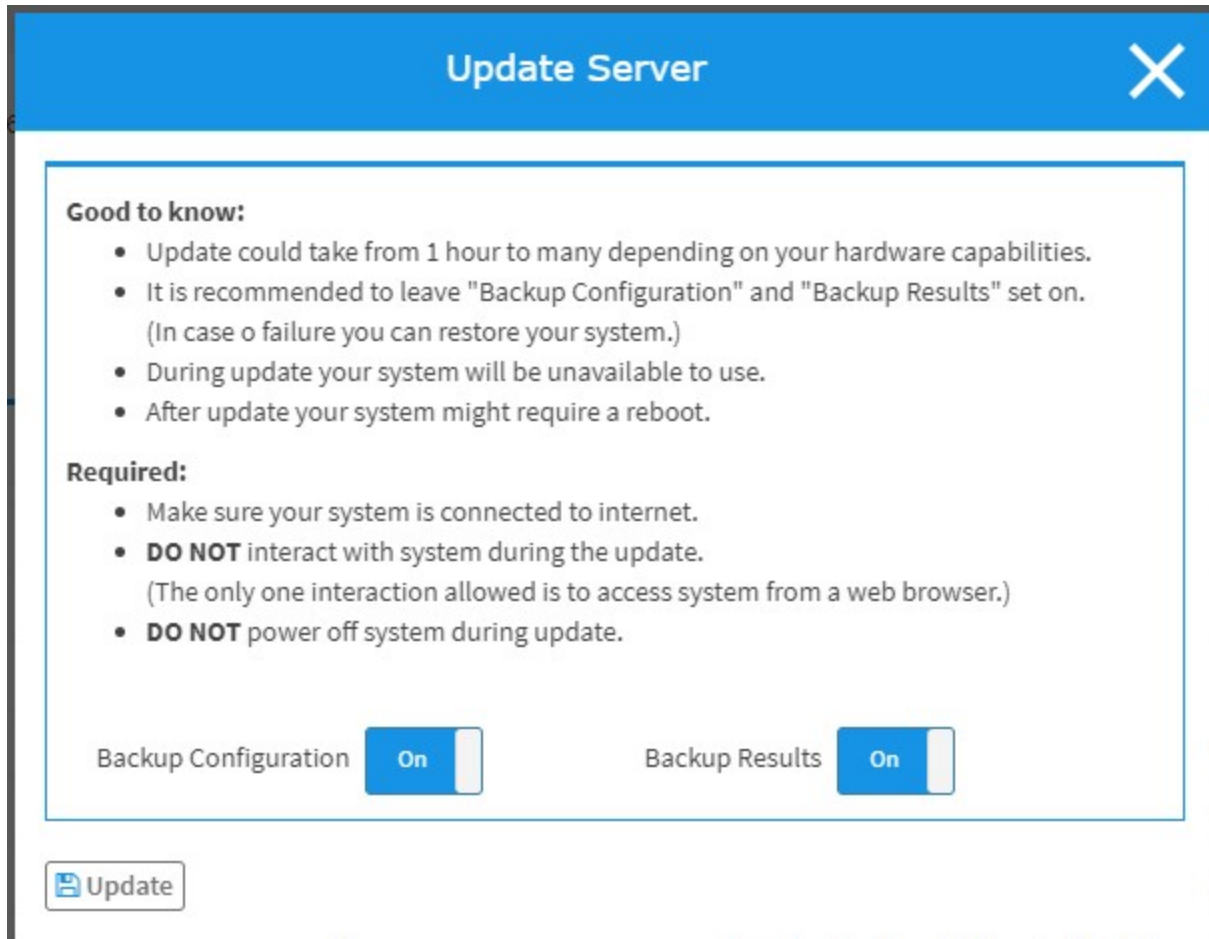
Hawkeye Configuration Data

Hawkeye Configuration data is stored in a SQL database. System Administrators are presented with the option to create a backup of the Hawkeye Configuration data as part of a server update. Otherwise a backup can be manually triggered from the Hawkeye UI.

Creating a Backup

Backing up during an update

The most common time for System administrators to want to create a back up is right before an update to the system. Knowing this information, during the upgrade dialog the System Administrator is presented with the option to create backups of Configuration data, Results data, or both. By setting the corresponding slider to "ON" a backup of the data will be created before any changes are made to the system.



Manual Backup

If for any reason you want to create a backup of the configuration data, results data, or both you can do so by navigating to the System page on the side bar and select the Backup button in the Data Pane. This will open the backup dialog:

Backup Data ✕

Good to know:

- Backups could take from few minutes to hours depending on your "Results" size(if selected) and hardware capabilities.
- Backups are executed in background, after process is finished you will receive a notification with status.

Required:

- **DO NOT** reboot / power off system during process.

Configuration ☐

Results ☐



Set the slider to **ON** next to the data you want to create a backup for and then select **Backup Now**.

NOTE

Hawkeye Server only stores one backup for the Configuration data and one backup for the Results data. Each subsequent backup will overwrite the existing backup.

Backup Storage locations

Backups are stored in /opt/backup/

The configuration backup is a single bzip file with the name: `Hawkeye_Configuration_backup.sql.bz2`

NOTE

Restore process expects the exact file name for configuration backups.

The results data backup is an entire directory. All the files in /opt/backup/elasticsearch/ are required to be in this exact path for the restore process.

You can store copies of these backup files in the database by using the **Upload** option in Hawkeye. You can copy `Hawkeye_Configuration_backup.sql.bz2` without any additional steps.

To move the results data backup, compress it into a single bzip2 file and then run the following commands on the cli:

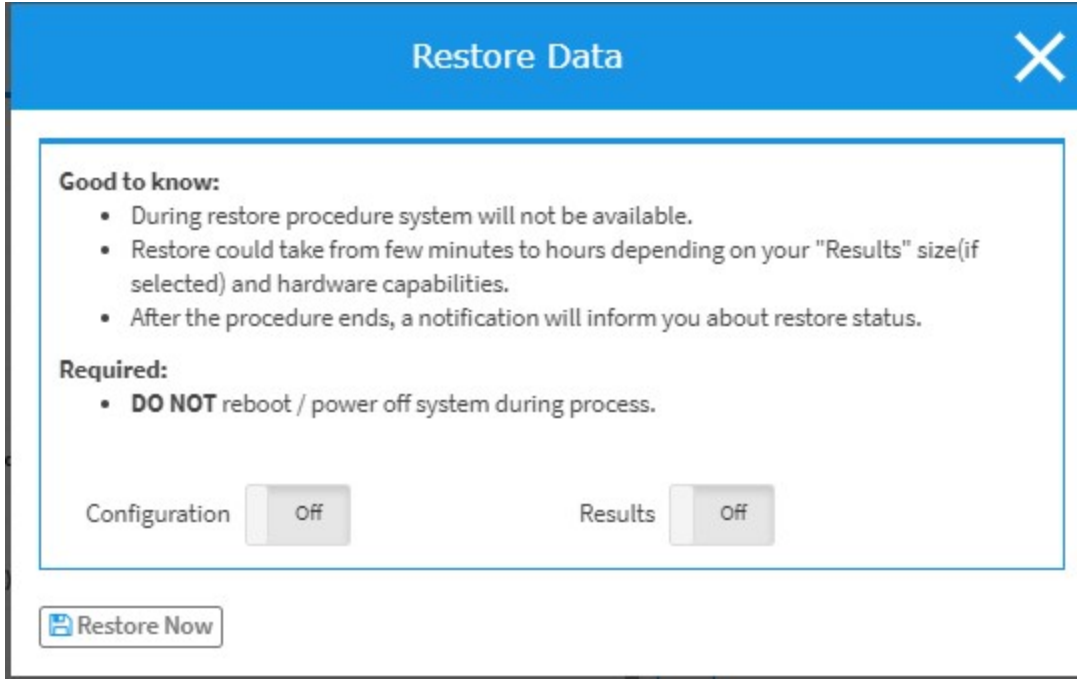
```
sudo cd /opt/backup/elasticsearch/
sudo tar -jcvf /tmp/hawkeye-results_<Date>.bz2 *
```

This will extract the bzip2 file to /tmp/hawkeye-results_<Date>.bz2. You can then move it to your off device storage for later use.

Restore Backups

The Hawkeye restore process expects the backups will be located where the backup process creates them. If files were saved to an off-server location, they will need to be returned to the original location for the restore process to work.

To start the restore process navigate to the System page on the side bar and select the Restore button in the Data Pane. This will open the restore dialog box.



The image shows a 'Restore Data' dialog box with a blue header bar containing the title 'Restore Data' and a close button (X). The main content area is white and contains the following information:

Good to know:

- During restore procedure system will not be available.
- Restore could take from few minutes to hours depending on your "Results" size(if selected) and hardware capabilities.
- After the procedure ends, a notification will inform you about restore status.

Required:

- **DO NOT** reboot / power off system during process.

At the bottom of the content area, there are two toggle switches: 'Configuration' and 'Results', both currently set to 'Off'. Below the content area is a button labeled 'Restore Now' with a document icon.

Set the slider to **ON** next to the data you want to restore and then select **Restore Now**.

Probe Management - Probe Busy

Hawkeye supports the ability for a custom script on an endpoint to determine when the probe is busy and therefore unavailable for tests. The Hawkeye Server in **Probe Management** indicates when an endpoint is down or up. This feature introduces a third state of busy for the endpoint. When a probe becomes busy, the state will be reflected in **Probe Management** in blue for that probe and it will not be available to run tests. All scheduled tests will remain in a queued state until endpoint leaves the busy state.

To turn on this feature, go to **Preferences> Advanced Options** and select **Probe Busy** parameter. A user defined script is placed on all endpoints that the probe busy feature is intended for. The purpose of the endpoint probe busy script is to determine when the probe is busy and not available for tests. You should define the script on the probe that determines when a probe is busy. The Hawkeye server will send a command down to each endpoint every so many seconds to execute the script and retrieve the busy status. The frequency of requesting probe busy status is determined by the Hawkeye System **Preference** parameter **Real services: default interval time for automatic endpoints** in **Advanced Options** tab.

When the feature probe busy is turned on, it is not necessary for all endpoints to contain the busy script. For all endpoints not containing the busy script, a result of not busy will be returned.

Below a sample script indicates a probe is busy for a defined period each day.

Use case

This is an example probe busy script which indicates a probe is busy during the peak network busy time of early evening.

NOTE

Use ssh to connect to the endpoint and create the probe busy script as below. The script must be in the specified directory and use the specified filename.

1. Create a file on the endpoint named **testIfProbeBusy** with the following script.

```
# cd /home/ixia/

# nano testIfProbeBusy.py
#!/usr/bin/python
import datetime
import sys
# use format HHMM
simulate_unavailable_start_time = "1326
simulate_unavailable_end_time = "2100
debug = 0
test_start_time = datetime.datetime.now()
test_start_time_str = test_start_time.strftime('%d-%b-%y %H:%M:%S')
test_start_hhmm = test_start_time.strftime("%H%M")
if test_start_hhmm >= simulate_uanavailable_start_time and test_start_hhmm <= simulate_uanavailable_end_time:
    endpoint_busy = 1
else:
    endpoint_busy = 0
if debug > 0:
    sys.stdout.write('\nDEBUG: %s endpoint_busy=%s unavailable from %s to %s\n' %
                    (test_start_time_str, endpoint_busy, simulate_uanavailable_start_time,
                     simulate_uanavailable_end_time))
sys.exit(endpoint_busy)
```

2. Next ensure script is available for executable

```
# chmod +x /home/ixia/testIfProbeBusy
```

3. On the Hawkeye server UI, under **Administration > Preference > Advanced Options**, turn on the **Probe Busy**.

The Hawkeye server will now send frequent commands (default every 5 seconds) down to each endpoint to run the testIfProbeBusy script. The script will return "1" if busy or "0" if available for tests. The Probe status for the endpoint will reflect busy with a blue color when busy. If an endpoint does not contain the script it will always return a not busy status.

When a test runs and the probe is blue indicating busy, the test will enter a queued state. If the test is scheduled to run every minute only one instance of that test will be queued. The scheduled test will continue as normal when probe becomes available. If three different tests are scheduled to run each minute against the probe only one instance of each of the three tests will be queued until probe becomes available.

NOTE

Probe Busy functions are only supported on XR endpoints.

Hawkeye Folder Structure

The Hawkeye application is installed in `/opt/hawkeye`. The Hawkeye application generates a number of log files which are automatically rotated to manage disk space. The user does not need to refer these log files unless advised by Keysight Customer Support.

The following directories contain log files:

- System log files are in `/var/log` directory.
- Individual Node to Node and Mesh test details or statistics files from endpoints are in `/opt/hawkeye/logs/Chariot` directory.
- Hawkeye system files and real service test logs are in `/opt/hawkeye/logs/Hawkeye` directory.

Set up e-mails with Hawkeye server

Hawkeye server supports the use of emails in three ways.

- Deliver generated reports though e-mails.
- Deliver error notification e-mails of test results including failure or error reasons.
- Deliver real service test e-mails for XR probes

To send reports and test results in e-mails, you must make available a mail client and configure it for the Hawkeye server. You do not need to configure mail client for Hawkeye server for real service tests.

Configure e-mail for reports and test results

Configure **Email** section in **Administration > Preferences** of Hawkeye server UI to set up a mail client. You will need to configure your SMTP server settings and your organization's e-mail account credentials in this section for the Hawkeye server to send e-mails.

Configure an Email SMTP server in **Preferences**. Following is a sample configuration.

SMTP server: mail.smtp2go.com

SMTP Authentication: Enable

SMTP Security: SSL (match whatever your email service uses SSL/TLS?)

Port: 465

NOTE

Configure port 465 for SSL, 587 port for TLS or 25 default SMTP port for no security, no other value is supported. Port 25 was the most commonly used port, but for security many email providers now block it as it is used by spammers.

Username: fred's_fish_n_chips@gmail.com

NOTE

This e-mail address is for reference.

Password: *****

Email address: fred's_fish_n_chips@gmail.com

Full name: fred's_fish_n_chips@gmail.com

NOTE

These are the credentials from e-mail server.

Max Email attempts: 3

NOTE

The e-mail process is abandoned after maximum attempt is attained.

Workflow of sending and retrieving e-mails

Outgoing and incoming mail are handled by two different types of server.

- SMTP server to send e-mail
- POP3/IMAP server to retrieve e-mail

You send email from your mail client (now running on Hawkeye server) by using the configured email address to the outgoing SMTP server using the configured port. The SMTP server first validates security and authentication, that is, connection security (SSL/TLS/25), port, username, password. The SMTP server works like the post office and performs a DNS lookup to find where to send the email (destination domain). The SMTP server may store the actual email on a local server (owned by domain sender) close to the Hawkeye server instead of the main SMTP server (that is, gmail in the above sample configuration). The SMTP server then sends the email to the destination domain Message Transfer Agent (MTA) server for long term storage. If the email destination domain is the same as the source (that is, gmail in the above sample configuration), the SMTP server will act as the MTA server. If different sender/dest email domain and no issues, it can be routed or forwarded. The email is sent by SMTP protocol to the (destination domain) MTA server. The received email is stored on the destination domain MTA (POP3/IMAP) server. The MTA (POP3/IMAP) server uses POP3 or IMAP protocol to transfer email to recipient's inbox (email client on the computer) when requested.

Workflow of real service test e-mail

Real service test e-mail works by using an email client on the probe. Only the XR2000 currently supports an email client. Configuring the **Email** section of **Preferences** is not used in this case.

Configure real service test email

Configure the following parameters to send and receive real service test emails.

- Sending email server credentials
- Email address
- Outgoing Mail server
- Outgoing Mail port
- SMTP authentication: SSL
- SMTP Authentication user

- SMTP Authentication password
- Receiving email server credentials
- Incoming mail server
- Incoming mail port
- Incoming mail protocol
- Mail user
- Mail Password

The real service test e-mail times different stages, but most important is when the destination POP3/IMAP server receives the sent email. Probe will poll every second to check received email.

SNMP Configuration and Verification

You can configure and verify the Simple Network Management Protocol (SNMP). This section explains how to configure and verify the SNMP.

SNMP Traps

Hawkeye server supports the generation of SNMP traps for tests originated from endpoints. Tests configured to generate SNMP traps on certain error/failure conditions sends the SNMP traps in MIB format to a designated SNMP server.

There are six steps involved:

- How to use SNMP traps
- Verify Hawkeye Server MIBs file can be interpreted by the User's SNMP server
- Configure HawkeyeServer in **Preferences** to send SNMP trap to a specific User's SNMP server
- Configure User's SNMP server to receive Hawkeye server originated SNMP traps
- Enable SNMP traps to be triggered for endpoint scheduled tests
- Verify SNMP trap successfully received and decoded by SNMP server
- Advanced Troubleshooting – Monitor Hawkeye server SNMP debug logs

How to use SNMP Traps

SNMP traps can be configured to be triggered by test results on end points. The time between tests and consequently the possible reporting time between SNMP traps is Customer configurable. The shortest time between tests on an endpoint is one minute.

SNMP traps are supported for all test types. When configuring a test, select **Show Alarm options** to see the SNMP configuration.

Each test can be enabled to trigger SNMP traps, based on a failure or error result. There is the option to only report an SNMP trap on a **change**. This means if a test on an endpoint is scheduled to run every 5 minutes and it is failing every time, it is only reported as an SNMP trap the very first time it fails, then it generates another SNMP trap once the endpoint test result changes to be an error or passes.

Verify Hawkeye Server MIBs file can be interpreted by the User's SNMP server

Hawkeye SNMP traps encoding is identified by three files. Retrieve the following three MIBs files from the Hawkeye server so that they can be verified:

- /usr/share/snmp/mibs/NET-SNMP-MIB.txt
- /opt/hawkeye/includes/MIBS/Hawkeye-MIB.txt
- /opt/hawkeye/includes/MIBS/IXIA-SMI.txt

There are a number of independent free web sites that verify that custom MIB files confirm to correct standards.

Suggested free independent web sites to verify Hawkeye MIBs format:

```
http://www.simpleweb.org/ietf/mibs/validate/upload.php
http://www.muonics.com/Tools/smichk.php
```

From these web sites pull in the retrieved two Hawkeye files and validate to MIBs standards. There are three supported versions of MIB files (v1, v2 and v3).

Configure Hawkeye Server to send SNMP trap to a specific SNMP server

Enable SNMP traps to be generated by the Hawkeye server.

To enable the SNMP traps, do the following:

Go to **Administration > preferences > Alarms** toggle the **Enable/disable the alarms received using SNMP**.

Also configure the IP address of the your SNMP trap server, to which the SNMP traps are to be sent to.

Multiple SNMP trap receivers can be supported by using a third party application that will forward received UDP packets from the Hawkeye Server IP address/port to a defined list of destination SNMP trap receivers (IP/port). There are many free applications that are available for this purpose, such as the free linux **samplicator** application. The SNMP server IP defined on the Hawkeyeserver to achieve this will be the IP of the third party application doing the forwarding.

Configure your SNMP server to receive Hawkeye server originated SNMP traps

There are many SNMP servers (receiver) available but the required basic configuration is as follows:

1. Load in the two Hawkeye MIB modules identified and retrieved from the Hawkeye server in the above steps (NET-SNMP-MIB.txt, IXIA-SMI.TXT and Hawkeye-MIB.txt). This enables your SNMP trap server to decode the Hawkeye SNMP traps.
2. Configure the host address of the Hawkeye server that will be the source of the SNMP traps.
3. Set the host port used to generate and be monitored for the SNMP traps to **162**.
4. Set the **community** to **ixia**.

- Following a Hawkeye Server upgrade, remove, flush, or delete any Hawkeye MIB files from the SNMP server (receiver) and pull in the latest version as new test types. Else, changed test metrics will cause decode issues.

Enable SNMP traps to be triggered for endpoint scheduled tests

- Schedule a test (example, node to node Skype4B traffic test) on probes expected to generate SNMP traps on error/failure conditions.
- When configuring the test, set thresholds to expected values additionally select **Show Alarm Options**.
- Go to **Show Alarm Options** and enable the **SnmpTrap**.
- Also select the conditions for the alarm to be triggered that generates the SNMP Trap (Error/Failed/Status change).

Verify SNMP trap successfully received and decoded by SNMP server

Invoke SNMP trapviewer from your SNMP server. Generate SNMP trap by running Hawkeye test that fails. Confirm that the SNMP trap received in the trap viewer is as expected and can be decoded.

Advanced Troubleshooting – Debug SNMP packet routing issues

Putty (ssh) to Hawkeye server and monitor SNMP trap logs to confirm that tests are triggering sending of SNMP traps to SNMP server.

On the Hawkeye server go to directory **/opt/hawkeye/logs/Hawkeye** and monitor server daemon logs with the following command.

```
tail -f ServerDaemon.log
```

Confirm able to see logs such as the one below where the first IP address is the IP address of the SNMP server to receive the SNMP trap.

```
20:24:58 UTC (Alarm send) sending snmptrap -
snmptrap -M /usr/share/snmp/mibs/:/opt/hawkeye/includes/MIBS -c ixia -v 2c 10.220.120.11
"" HAWKEYE-MIB::hawkeye-notification trapID s "xr2000autoqa2xrpiWi-
Fiqa2Skype4BTrafficDelaysmsAudioRTPfromto12kbpsFAIL" summary s "xr2000autoqa2 to xrpiWi-
Fiqa2 Skype4B Traffic Audio RTP fromto 12 kbps Delay ms Failed" runID s "8285" TimeStamp
s "2016-11-30 20:24:57" TestType s "Skype4B Traffic" TestStatus s "Failed" From s
"xr2000-auto-qa-2" To s "xrpi-Wi-Fi-qa-2" PAIRNAME s "Audio RTP from->to 12 kbps"
MetricName s "Delay (ms)" METRICVALUE s "2.5" DefaultThresholdType s "1" THRESHOLD s
"9000" METRICSTATUS s "Failed" FailReason s "Threshold failed on Audio RTP from->to 12
kbps Delay (ms) "
```

Additionally use TCP dump command to confirm that the SNMPtrap packet is not being blocked in the network. The TCP logs below (taken from Hawkeye server) show that a SNMP trap is generated from Hawkeye server (10.220.120.127) and sent to the SNMP trap server/receiver (10.220.20.45) but is blocked as unreachable.

TCP dump from Hawkeye server:

```
tcpdump -n -i eth0 | grep -i "10.220.120.127"
21:11:38.246127 IP 10.220.120.127.52365 > 10.220.20.45.snmptrap: C=ixia V2Trap(183)
.1.3.6.1.2.1.1.3.0=148999358 .1.3.6.1.6.3.1.1.4.1.0=.1.3.6.1.4.1.8072.2.1.0.3
.1.3.6.1.4.1.8072.2.1.2.1=3 .1.3.6.1.4.1.8072.2.1.2.2="Amazon-KFFOWI"
```

```
.1.3.6.1.4.1.8072.2.1.2.3="Amazon-KFFOWI" .1.3.6.1.4.1.8072.2.1.2.4="0"
.1.3.6.1.4.1.8072.2.1.2.5="TestAgent1"
21:11:38.247491 IP 10.220.20.45 > 10.220.120.127: ICMP 10.220.20.45 udp port snmptrap
unreachable, length 234
```

In the above example, unlike the Network Unreachable and Host Unreachable messages which come from routers, the Port Unreachable message comes from the SNMP server/receiver. The primary implication for troubleshooting is that the frame was successfully routed across the communications infrastructure, the last router ARP'ed for the host, got the response, and sent the frame. Furthermore, the intended SNMP server (destination host) was on-line and willing to accept the frame into its communications buffer. The frame was then processed and an attempt was made to send the data up to the destination port number (UDP port 162) and the port process (SNMP server) did not exist. The protocol handler then reports Destination Port Unreachable.

For SNMP traps generated by Hawkeye Server to reach a third party SNMP trap receiver the UDP ports 161 and 162 need to be open for UDP traffic on the Hawkeye server and switches between the two parties. One way to open the ports on the Hawkeye Server is to putty to the Hawkeye Server then using the following CLI commands to open the firewall for SNMP traffic:

NOTE

You need privilege escalation for the following steps.

```
ksuser@he-qa-deb:~$ su -
Password:
root@he-qa-deb:~# firewall-cmd --add-service=snmp --permanent
success
root@he-qa-deb:~# firewall-cmd --add-service=snmptrap --permanent
success
root@he-qa-deb:~# firewall-cmd --reload
success
```

Hawkeye SNMP MIB Information

This section covers information about SNMP MIB provided by Hawkeye

MIB file location

MIB files can be found into the following directory /opt/hawkeye/includes/MIBS

Associated MIB Descriptors

Associated MIB descriptors can be needed to import MIB. The associated MIB descriptor are available in the same subdirectory:

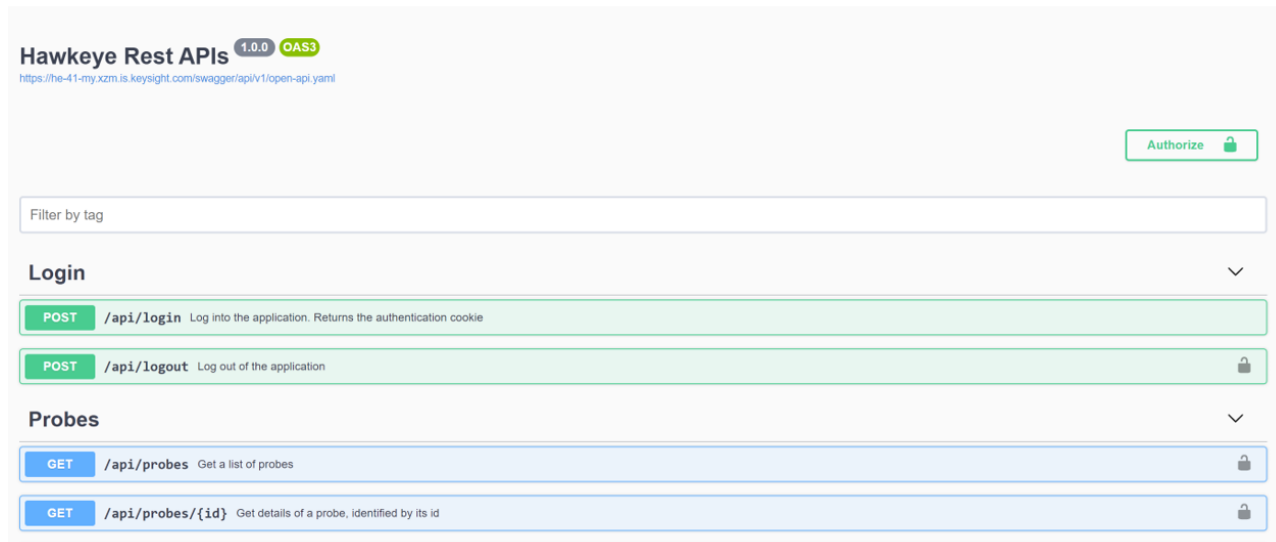
/usr/share/snmp/mibs

Rest API

Hawkeye solution has a web-based GUI. You can access functions of Hawkeye without the web user interface using Rest APIs directly.

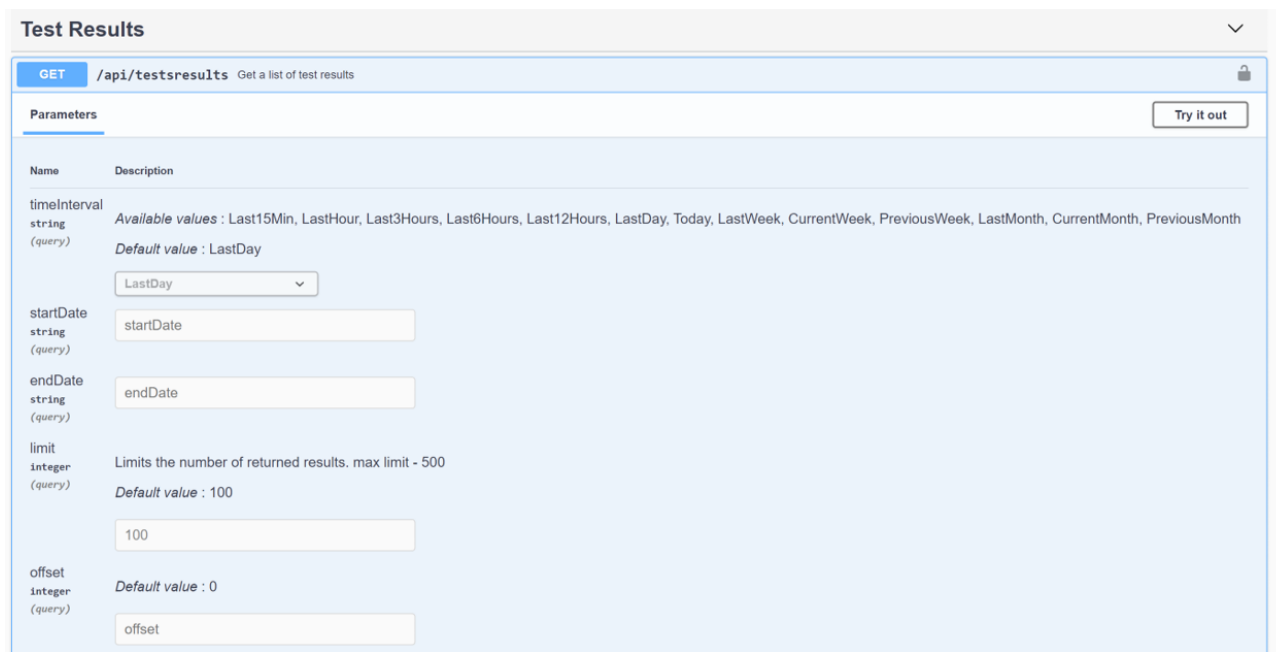
List of supported Rest APIs are available under Documentation > Rest API. Swagger is a tool, which is used to visualize and interact with the API's resources without applying any implementation logic.

You can download open API spec from the <open-api.yaml> link at the top of Documentation > Rest API and use it in any other tool.



Example: Rest API to get a list of test results.

Below is the image of test result request. Select **Try it out** button on the tool to run it without implementing on client side for testing.



Below is image of **test result response** showing JSON format to be returned on the request call.

Responses		
Code	Description	Links
200	Tests Results	No links
	<div>Media type application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "total_count": 0, "count": 0, "limit": 0, "offset": 0, "has_more": 0, "records": [{ "id": "string", "date": "2020-08-03T18:10:38.977Z", "duration": 0, "meshid": 0, "mesh": 0, "module": "N2N", "probeFrom": "string", "probeTo": "string", "reasonCause": "string", "status": "Passed", "testId": 0, "testOptions": "string", "testTag": "string", "testType": "string" }] }</pre></div>	
400	Parameters or body were in an incorrect format	No links
	<div>Media type application/json</div> <div>Example Value Schema</div> <div><pre>{ "error": "string", "code": "404 Notfound" }</pre></div>	

To check the schema of response and its elements click on schema link as highlighted below

Responses		
Code	Description	Links
200	<p>Tests Results</p> <p>Media type <input type="text" value="application/json"/></p> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre> TestsResultsList { total_count integer count integer limit integer offset integer has_more BooleanType integer Enum: [0, 1] records { TestsResults { id string date date-time string(\$date-time) duration integer meshId integer mesh BooleanType integer Enum: [0, 1] module string Enum: [2] probeFrom string probeTo string reasonCause string status string Enum: [2] testId integer testOptions string testTag string testType string } } } </pre>	No links
400	<p>Parameters or body were in an incorrect format</p> <p>Media type <input type="text" value="application/json"/></p> <p>Example Value Schema</p> <pre> { "error": "string", "code": "404 NotFound" } </pre>	No links

How to create a new test:

The Tests **topology** field (NodeToNode, RealService, Mesh) in a request defines the type of test that is going to be created.

Example requests for each test topology are available in [REST examples](#) section.

The **parameters** array contains list of basic and pairs (for NodeToNode and Mesh) of a testtype. Use `/api/testtypes/{id}/parameters` API where {id} is testtype Id to fetch list of supported test parameters for a specific test type.

The **threshold** and **alarm** are optional data. Use it if client wants to change alarm or threshold's default settings. Outlier under threshold is also optional.

The **startDate** and **endDate** fields are in UTC time format **YYYY-MM-DDTHH:MM:SSZ**

REST examples

NodeToNode Request:

```
{
  "identifier": "",
  "isMesh": "0",
  "testtypeId": "5163",
  "fromProbeId": "2",
  "toProbeId": "3",
  "meshId": "2",
  "enforceSchedule": "0",
  "frequency": "2",
  "startDate": "2020-08-11T18:30:44Z",
  "endDate": "2120-08-11T18:16:48Z",
  "testTemplateExecutionId": "0",
  "topology": "NodeToNode",
  "fromIps": "Default",
  "toIps": "Default",
  "parameters": {
    "basic": [
      {
        "id": "1",
        "value": "30"
      },
      {
        "id": "4",
        "value": "5"
      }
    ],
    "pairs": [
      {
        "id": "15175",
        "parameters": [
          {
            "id": "105",
            "value": "BestEffort"
          }
        ]
      },
      {
        "id": "15176",
        "name": "TCP from->to N pairs COS 2",
        "parameters": [
          {
            "id": "105",
```

```

    "value": "BestEffort"
  }
]
},
{
  "id": "15177",
  "name": "TCP from->to N pairs COS 3",
  "parameters": [
    {
      "id": "105",
      "value": "BestEffort"
    }
  ]
},
{
  "alarm": {
    "trigger": {
      "failure": "0",
      "error": "0",
      "change": "0",
      "outlier": "0"
    },
    "notification": {
      "email": {
        "enabled": false,
        "address": ""
      },
      "snmp": {
        "enabled": false
      }
    }
  },
  "thresholds": [
    {
      "pairId": "15175",
      "pairName": "TCP from->to N pairs COS 1",
      "metric": "Throughput from->to (Mbps)",
      "value": "5571508.445",
      "operator": "1",
      "unit": "Mbps",
      "factor": "1",
      "autoLearnEnabled": "0",
      "sensitivity": 95,
      "outlier": {
        "enabled": "0",

```

```

    "windowSec": "1740",
    "windowPct": "52"
  },
  {
    "pairId": "15176",
    "pairName": "TCP from->to N pairs COS 2",
    "metric": "Throughput from->to (Mbps)",
    "value": "5229714.123",
    "operator": "1",
    "unit": "Mbps",
    "factor": "1",
    "autoLearnEnabled": "1",
    "sensitivity": 99,
    "outlier": {
      "enabled": "1",
      "windowSec": "1800",
      "windowPct": "50"
    }
  },
  {
    "pairId": "15177",
    "pairName": "TCP from->to N pairs COS 3",
    "metric": "Throughput from->to (Mbps)",
    "value": "4844271.8789",
    "operator": "1",
    "unit": "Mbps",
    "factor": "1",
    "autoLearnEnabled": "1",
    "sensitivity": 99.9,
    "outlier": {
      "enabled": "1",
      "windowSec": "1800",
      "windowPct": "50"
    }
  }
]
}

```

RealService Request:

```

{
  "identifier": "api call",
  "isMesh": "0",
  "testtypeId": "7148",
  "fromProbeId": "1",

```

```

"toProbeId": "0",
"meshId": "0",
"enforceSchedule": "0",
"frequency": "4",
"startDate": "2020-08-14T15:43:32Z",
"endDate": "2120-08-14T15:43:32Z",
"testTemplateExecutionId": "0",
"topology": "RealService",
"parameters": {
  "basic": [
    {
      "id": "202",
      "description": "Destination Servers",
      "value": "www.test.com"
    },
    {
      "id": "203",
      "description": "Interval",
      "value": "0.02"
    },
    {
      "id": "204",
      "description": "Count",
      "value": "100"
    },
    {
      "id": "205",
      "description": "Packet Size",
      "value": "100"
    },
    {
      "id": "227",
      "description": "ip protocol",
      "value": "ipv4"
    },
    {
      "id": "3006",
      "description": "Class of Service",
      "value": "0"
    },
    {
      "id": "3007",
      "description": "Jitter Calculation",
      "value": "1"
    }
  ]
}

```

```

},
"thresholds": [
{
"pairId": "7148",
"pairName": "ICMP Test",
"metric": "Availability",
"value": "1",
"operator": "1",
"unit": null,
"factor": "1",
"autoLearnEnabled": 0,
"sensitivity": 100
},
{
"pairId": "7148",
"pairName": "ICMP Test",
"metric": "Jitter (ms)",
"value": "8",
"operator": "0",
"unit": "ms",
"factor": "1",
"autoLearnEnabled": 0,
"sensitivity": "0"
},
{
"pairId": "7148",
"pairName": "ICMP Test",
"metric": "Jitter Max (ms)",
"value": "8",
"operator": "0",
"unit": "ms",
"factor": "1",
"autoLearnEnabled": 0,
"sensitivity": "0"
},
{
"pairId": "7148",
"pairName": "ICMP Test",
"metric": "Jitter standard deviation",
"value": "2",
"operator": "0",
"unit": null,
"factor": "1",
"autoLearnEnabled": 0,
"sensitivity": "0"
},
}

```

```

{
  "pairId": "7148",
  "pairName": "ICMP Test",
  "metric": "Packet loss",
  "value": "1",
  "operator": "0",
  "unit": null,
  "factor": "1",
  "autoLearnEnabled": 0,
  "sensitivity": "0"
},
{
  "pairId": "7148",
  "pairName": "ICMP Test",
  "metric": "Round trip time avg (ms)",
  "value": "300",
  "operator": "0",
  "unit": "ms",
  "factor": "1",
  "autoLearnEnabled": 0,
  "sensitivity": "0"
},
{
  "pairId": "7148",
  "pairName": "ICMP Test",
  "metric": "Round trip time max (ms)",
  "value": "400",
  "operator": "0",
  "unit": "ms",
  "factor": "1",
  "autoLearnEnabled": 0,
  "sensitivity": "0"
},
{
  "pairId": "7148",
  "pairName": "ICMP Test",
  "metric": "Round trip time min (ms)",
  "value": "300",
  "operator": "0",
  "unit": "ms",
  "factor": "1",
  "autoLearnEnabled": 0,
  "sensitivity": null,
  "outlier": {
    "enabled": "1",
    "windowSec": "1800",

```

```

    "windowPct": "50"
  }
},
{
  "pairId": "7148",
  "pairName": "ICMP Test",
  "metric": "Standard deviation (ms)",
  "value": "20",
  "operator": "0",
  "unit": "ms",
  "factor": "1",
  "autoLearnEnabled": 0,
  "sensitivity": "0"
}
] }

```

create a Test using Mesh Id:

```

{
  "identifier": "MeshTestFromAPI",
  "isMesh": "1",
  "testtypeId": "5129",
  "meshId": "1",
  "enforceSchedule": "0",
  "frequency": "5",
  "startDate": "2020-08-14T15:50:18Z",
  "endDate": "",
  "testTemplateExecutionId": "0",
  "topology": "Mesh",
  "parameters": {
    "basic": [
      {
        "id": "1",
        "description": "Traffic Duration",
        "value": "30"
      },
      {
        "id": "105",
        "description": "DSCP Setting",
        "value": "BestEffort"
      }
    ]
  }
}

```

Server upgrade

You can upgrade the Hawkeye server in both online and offline processes.

Online upgrade

You can upgrade the Hawkeye server running Hawkeye 5.0.1 or later versions with a connection to the internet to the latest release of Hawkeye from the UI:

Select **System** from the side bar and select the **Check for updates** button:

The screenshot shows the Hawkeye System administration interface. The top navigation bar includes a hamburger menu, the word 'System', and user icons for notifications, help, and 'sysadmin'. A 'Check for updates' button is located in the top right. Below the navigation bar, there are two tabs: 'System' (selected) and 'License Usage'. The main content area is divided into two panels. The left panel, titled 'System', contains a table of system configuration and status. The right panel, titled 'Runtime', contains a table of runtime statistics.

System	
System Version	5.0.1 (10762971)
Server Address	10.36.32.22
Disk Usage	5% of 255G used
Client Session Idle Time (seconds)	3600
List Display Refresh Time (milliseconds)	120000
Log files Retention Time (minutes)	240
Database Records Retention (Days)	365
Time on Server	2020-12-08 03:33:37
Time on Client	2020-12-07 22:33:37
Daylight Saving Setting	0

Runtime	
Number of active probes	13
Number of active concurrent users	2
Number of Tests currently running	0
Number of Mesh Tests currently running	0
Number of Node to Node Tests currently running	0
Number of Real Service Tests currently running	0
Number of total pairs in Last Run	2
Average Pairs executed Last Day	4
Max Pairs executed Last Day	12
Min Pairs executed Last Day	1

If an update is available, the **Update Server** button will appear. Select **Update Server** button to start the update dialog.

Update Server
✕

Good to know:

- Update could take from 1 hour to many depending on your hardware capabilities.
- It is recommended to leave "Backup Configuration" and "Backup Results" set on.
(In case of failure you can restore your system.)
- During update your system will be unavailable to use.
- After update your system might require a reboot.

Required:

- Make sure your system is connected to internet.
- **DO NOT** interact with system during the update.
(The only one interaction allowed is to access system from a web browser.)
- **DO NOT** power off system during update.

Backup Configuration

On

Backup Results

On

Update

Here you have the option to have either your configuration or your results or both backed up prior to the upgrade. Once your selections are made select **Update** button. You will be asked to confirm that update by selecting **OK** to start you update.

After the Update has started you will be redirected to an Update Status page. This page reports when each step in the update is done and will check to see if the system will need to be rebooted once the update is complete. If a reboot is required, you will be prompted to reboot by selecting the **Reboot Now** button otherwise you will be redirected to the login page when the update completes.

Updating Hawkeye in progress ...

Start checking basic requirements ...
DONE: Basic requirements for update have been met.
Configuration backup in progress ...
DONE: Configuration backup was executed with success.
Results backup in progress ...
DONE: Results backup executed with success.
Upgrading server in progress ...
Dependency checking ...
DONE: Dependency checking
Downloading packages ...
DONE: Downloading packages
Installing updates ...
DONE: Installing updates
Cleaning Up ...
DONE: Cleaning Up
Verifying updates ...
DONE: Verifying updates
DONE: Server was updated with success.
Check if reboot machine is needed ...
Now you have to reboot the system ...

Reboot Now

NOTE

During the update the system will not be available for tests or access by other users.

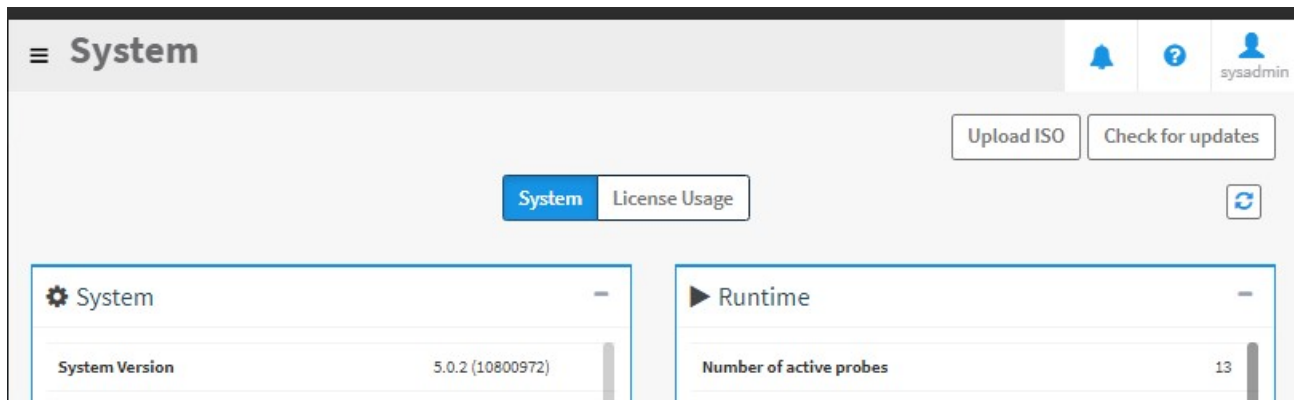
Offline upgrade

You can upgrade Hawkeye server in offline mode either from user interface (UI) or from the command line program (CLI). Both these processes are mentioned in this section.

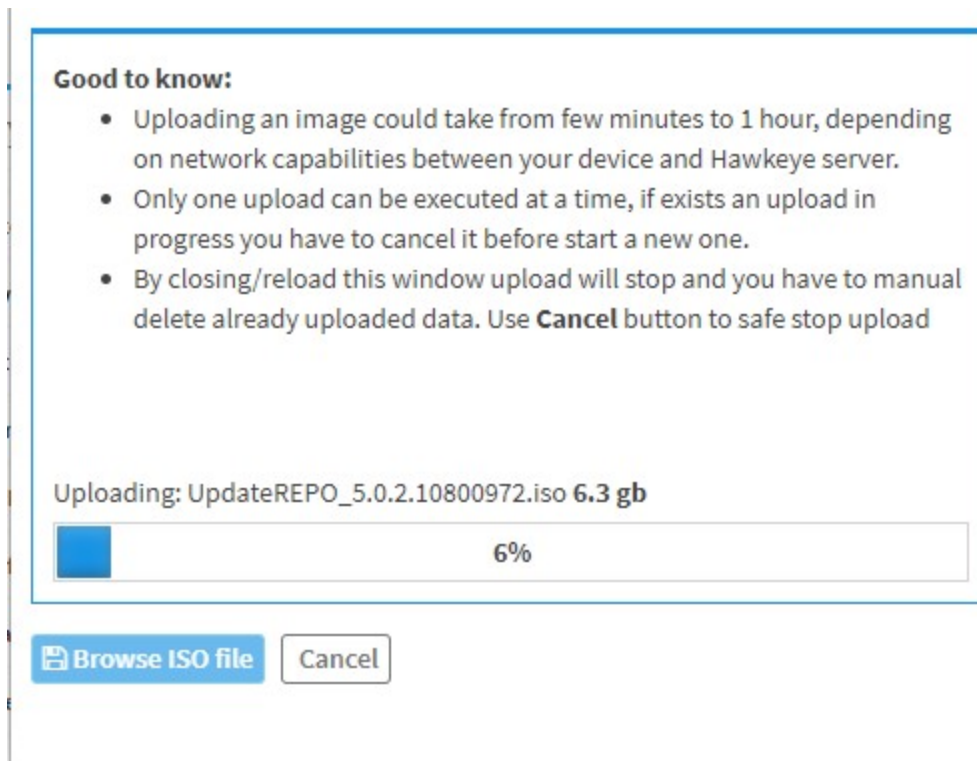
UI process

Hawkeye Servers running Version 5.0.2 or later support offline upgrade from the UI. This does require you to have already downloaded the UpdateRepo .iso file from support.ixiacom.com to your local machine ahead of time. After the file has been downloaded, you can perform the following steps to upload it to your Hawkeye server and upgrade your server.

Select **System** from the side bar and select the **Upload ISO** button.



The **Upload** dialog will open in a new browser window, select the **Browse ISO File** button and select the UpdateRepo.iso file that you previously downloaded. After you select **Open** the upload will start immediately. If you do not refresh or close this browser window, the update will continue while you continue to use Hawkeye.



After the upload completes the system will validate that it is a Keysight update repository ISO file and show a notification in the Hawkeye UI.

Notifications Details		✕	
Id	24		
Type	ISO image validation success		
Content	Image: UpdateREPO_5.0.2.10800972.iso was validated with success. Please refresh your browser.		
Is Active	yes		
Is Read	no		
Login	sysadmin		
Created At	2021-02-08 15:34:26		

After Hawkeye validates the image, it will display the image available offline for update on the System page. Then you can select the **Update Server** button to start the update process and continue the process similar to online update.



CLI process

You can upgrade the Hawkeye server running Hawkeye 5.0.1 or higher versions offline to the latest release of Hawkeye from the CLI. To do this, you must download the .iso image from support.ixiacom.com.

After you have downloaded the .iso file, transfer it to the /tmp directory of your Hawkeye server and run the following command:

```
#updateServer --offline <path to .iso file>
```

If a reboot is required, Hawkeye will prompt you to reboot the system after the upgrade.

Note:

NOTE

The updateServer command can also perform backups and check for available updates if the server has an internet connection. To see other options available, run the following command:

```
#updateServer -h
```

Configure Time Zone and NTP

Configure Time Zone

When configuring the time zone for various devices in your Hawkeye environment please follow the table below for the commands required.

Action	Hawkeye Server/XR-E1S/XR-VM/XR3000	Xr2000/xrpi
List available time zones	timedatectl list-timezones	ls /usr/share/zoneinfo/
Set time zone	timedatectl set-timezone ZONE	ln -sf /usr/share/zoneinfo/ZONE /etc/localtime
Example set time zone	timedatectl set-timezone America/New_York	ln -sf /usr/share/zoneinfo/America/New_York /etc/localtime

NOTE

Changing the Time Zone in the Hawkeye Server after recording the test results leads to incorrect display of previously recorded results in the current time zone. There is either a gap in results or results into the future that equals to the offset between the prior and current time zones.

Configure NTP Client

All probes when connected to a Hawkeye Server will configure their NTP solution to sync time with the Hawkeye Server by default you can configure additional NTP servers to help keep probe time in sync. When configuring the Network Time Protocol (NTP) on various devices in your Hawkeye environment please follow the table below for the configuration file to use and the Service to restart after making the configuration change.

Action	Hawkeye Server/XR-E1S/XR-VM/XR3000	Xr2000/xrpi
Configuration File	/etc/chrony/chrony.conf	/etc/ntp.conf
Restart Service	systemctl restart chronyd	Xr2000: service ntpd restart XRpi: service ntp restart
Example server key	server your.ntp.server iburst	server your.ntp.server iburst

To Configure NTP on your device find your devices configuration file above and add a server key at the bottom of the file using the format presented above. Following is an example of the config file for an XR-VM:

```
# Welcome to the chrony configuration file. See chrony.conf(5) for more
# information about usable directives.

# Include configuration files found in /etc/chrony/conf.d.
```

```

confdir /etc/chrony/conf.d

# Use Debian vendor zone.
pool 2.debian.pool.ntp.org iburst

# Use time sources from DHCP.
sourcedir /run/chrony-dhcp
.
.
.
# Log files location.
logdir /var/log/chrony
.
.
.
# Step the system clock instead of slewing it if the adjustment is larger than
# one second, but only in the first three clock updates.
makestep 1 3

# Get TAI-UTC offset and leap seconds from the system tz database.
# This directive must be commented out when using time sources serving
# leap-smeared time.
leapsectz right/UTC

```

After editing this file to add your servers, restart the time service using the appropriate restart service command from the above table.

Configure Hawkeye as NTP Server

By default, the Hawkeye Server denies all NTP sync requests it gets from other NTP clients. The system supports allowing client connections by subnet, IP , or allow all client connections. To configure this, add allow statements to the /etc/chrony.conf file in the section by using the following command:

```
# Allow NTP client access from local network.
```

Allow Statement	Description
allow 1.2.3.4	Allows the IP address 1.2.3.4 to connect as client
allow 1.2	Allows any IPv4 Address that start with 1.2 to connect as a client
allow 192.168.22.0/24	Allows the 192.168.22.0 class c subnet to connect as a client
Allow 265d:dc10::/64	Allows IPv6 265d:dc10:: with a 64 bit prefix connect as client
allow 0/0	Allows all IPv4 addresses to connect as a client
allow ::/0	Allows all IPv6 addresses to connect as a client
allow	Allows all addresses to connect as a client

Configure UI Time Zone

Each user can assign the required time zone. The required time zone and the time zone used for creating the reports requested by a user, is available in the Graphical User Interface (GUI).

For users with the System Administrator role, this preference is found in the **Preferences > Administration > GUI** tab. Select the required time zone in the **Timezone** drop down list.

Main				
	Parameter Name	Value	Description	Details (Current Setting)
GUI	Timezone	<div>America/New_York</div>	User Timezone	America/New_York
Alarms	Default Calendar Range	<div>LastDay</div>	This parameters may take the following values: LastDay, LastHour, LastWeek. It defines the default time scale for the system when the user logs in. For best performance, use LastHour or LastDay.	Default calendar range is LAST DAY
Maps	Refresh Table Interval (seconds)	<div>120</div>	Refresh table timer (in seconds). This will allow refreshing any table with autorefresh option on every defined interval. It is recommended to keep it above 10 seconds for optimized system performance.	
Email	Allow URL	<div>1</div>	Set it to 1 to allow probe to have url instead of IP	Allow probe to have URL instead of IP
VisionEIS				

For Users with any other role this preference is found in the **Preferences > Administration > General** tab. Select the required time zone in the **Timezone** drop down list.

Notification Alerts Silent Mode	0	Silence the notification alerts (bell icon in the top right corner)	Disable Notification alerts silent mode
Timezone	America/New_York	User Timezone	America/New_York

Save

Cancel

Restore To Default

This page intentionally left blank.

Index

C

customer assistance 3

P

product support 3

T

technical support 3

U

upgrade server 32

 online upgrade 32

upgrade server;offline upgrade 32

upgrading hawkeye server 32



© Keysight Technologies, 2025

This information is subject to change
without notice.

www.keysight.com

913-2383-01 Rev A