

JT-NM Tested March 2020
AMWA NMOS / JT-NM TR-1001-1
Media Nodes, Registries, Controllers
Results Catalog





“JT-NM Tested March 2020 Program” NMOS/TR-1001 Test Plan v1.4

Due to the COVID-19 crisis, the face-to-face stage of the testing was canceled. The test plan was adapted post-factum to reflect the changes made to the program.

Self-Tested Results

For the first time, this catalog includes Self-Tested results, and, for vendors who submitted their own test results, JT-NM has created a Self-Tested badge. The JT-NM was unable to hold a planned face-to-face meeting in March of 2020 where vendor-submitted tests could be validated. We decided to modify our program so that vendors who self-tested their products in accordance with the JT-NM Tested test plans could submit their results for publication in this catalog. JT-NM is responsible for the development and publication of the test plans. Vendors are solely responsible for any representations they have made in the self-tested results they have submitted; JT-NM is simply reprinting the submitted results in this catalog.

JT-NM Tested Results

For NMOS Controllers, the JT-NM Test Team was able to independently validate the vendor submitted test results using a Virtual Private Network and a lab at CBC/Radio-Canada. In the case of these controller tests, the JT-NM is responsible for development and publication of the controller test plan, and also for validating vendor results included in this catalog.



Version 1.4 Changelog

- The test plan was adapted to reflect the changes made due to the cancellation of the face-to-face stage of the testing.

Version 1.3 Changelog

- Clarified that 'rtp_enabled' will be set to true during stream tests for Senders in section 4.1.
- Add notes about test exclusions for March 2020
- Further highlight notes around which registries and controllers can participate

Version 1.2 Changelog

- Added public link to download v1.6 of Riedel NMOS Explorer

Version 1.1 Changelog

- Revised wording of controller tests in sections 7 and 8 to improve clarity

Version 1.0 Changelog

- First draft. This document may undergo substantial changes ahead of the final version.
- Updated available automated tests based upon changes to the NMOS Testing Tool.
- Added sections for registry and controller testing.

Abstract and motivation behind the program

The JT-NM Tested program offers documented insight into how vendor equipment conforms to specific SMPTE standards and AMWA NMOS specifications.

Vendors who meet the testing criteria will have the opportunity to display badges and to make public statements about their participation. JT-NM will publish the JT-NM Tested Catalog listing vendor results along with a detailed test plan.

In a major change to the program, vendors will conduct self-testing prior to a face-to-face event where JT-NM validates the results.

It is important to note that the JT-NM Tested program is not a certification program; rather it is a snapshot in time of how vendor equipment conforms to key parts of SMPTE standards and AMWA NMOS specifications.



Acknowledgment

This Test Plan was prepared by an expert group within the Joint Task Force on Networked Media (JT-NM) with key contributions from: Andrew Bonney (BBC R&D), Arne Bönninghoff (Riedel), Bill McLaughlin (EEG), Félix Poulin (CBC), Jean Lapierre (Matrox), John Mailhot (Imagine Communications), Leigh Whitcomb (Imagine Communications), Mike Overton (Tektronix), Pedro Ferreira (Bisect), Peter Brightwell (BBC R&D), Serge Grondin (Grass Valley), Sunday Nyamweno (CBC), Thomas Kernen (Mellanox Technologies) and Willem Vermost (EBU).

Applicable Standards, Specifications, and versions

- **IETF RFC 792** Internet Control Message Protocol
- **IETF RFC 1034** Domain Names - Concepts and Facilities
- **IETF RFC 2131** Dynamic Host Configuration Protocol
- **IETF RFC 3376** Internet Group Management Protocol, Version 3
- **IETF RFC 4566** SDP: Session Description Protocol
- **IETF RFC 6763** DNS-Based Service Discovery
- **AMWA NMOS IS-04 v1.2 or later** NMOS Discovery and Registration Specification
- **AMWA NMOS IS-05 v1.0 or later** NMOS Device Connection Management Specification
- **AMWA NMOS IS-08 v1.0** NMOS Audio Channel Mapping Specification
- **IEEE 802.1AB-2016** Station and Media Access Control Connectivity Discovery (LLDP)
- **IEEE 1588-2008** Precision Clock Synchronization Protocol for Networked Measurement and Control Systems
- **SMPTE ST 2059-1:2015** Generation and Alignment of Interface Signals to the SMPTE Epoch
- **SMPTE ST 2059-2:2015** SMPTE Profile for Use of IEEE-1588 Precision Time Protocol in Professional Broadcast Applications
- **SMPTE ST 2110-10:2017** Professional Media over Managed IP Networks: System Timing and Definitions
- **SMPTE ST 2110-20:2017** Professional Media over Managed IP Networks: Uncompressed Active Video
- **SMPTE ST 2110-21:2017** Professional Media over Managed IP Networks: Traffic Shaping and Delivery Timing for Video
- **SMPTE ST 2110-30:2017** Professional Media over Managed IP Networks: PCM Digital Audio
- **SMPTE ST 2110-40:2018** Professional Media over Managed IP Networks: SMPTE ST 291-1 Ancillary Data
- **JT-NM TR-1001-1:2018** System Environment and Device Behaviors For SMPTE ST 2110 Media Nodes in Engineered Networks



Equipment used during testing

The equipment lists were adapted to reflect the changes made to the program because of the cancellation of the face-to-face event.

The following equipment was selected by the JT-NM Test Experts Group to support the “JT-NM Tested” event. In a number of cases, other vendor’s equipment would have worked equally well. To preserve the integrity of the testing environment, the team needed to choose one set of support equipment. The listing of a company below should not be taken to indicate that this is the only equipment that could have performed the tasks required. Note also that support equipment was *not* “JT-NM Tested” unless it is specifically listed in the test results matrix, and that no special status is awarded by the JT-NM to these companies other than to note that the JT-NM appreciates their support.

Supporting equipment and software used during the event

- AMWA NMOS Testing Tool
 - <https://github.com/AMWA-TV/nmos-testing>
 - See the Appendix for usage instructions. Latest ‘master’ version used.
- SDPoker
 - <https://github.com/AMWA-TV/sdpoker>
 - Note: The latest ‘master’ version of this fork is used as opposed to the original Streampunk repository in order to benefit from recent additions and bug fixes.
- Open-Source Sony nmos-cpp IS-04 registry and TR-1001 system resource
 - <https://github.com/sony/nmos-cpp>
 - Latest ‘master’ version used.
- Riedel NMOS Explorer v1.6 (Freeware)
 - <http://bit.ly/37qYEow>
- ISC DHCP server v4.3.5 on Ubuntu Linux 18.04
- Glass DHCP server user interface
 - <https://github.com/Akkadius/glass-isc-dhcp>
- BIND DNS server v9.11.3 on Ubuntu Linux 18.04



Reference NMOS Sender (RNS) and receivers (RNR) used during the event (only applicable to the NMOS Controllers testing)

- RNS: Riedel MuoN A10 with APP SDI-ST2110-2E Gateway - SDI HD/3G to ST 2110 Dual Encapsulator SFP Gateway. HW v100, SW v3.3
 - <https://www.riedel.net/products-solutions/distributed-video-networks/mediornet-muo-n/hardware/>
- RNR: Riedel MuoN A10 with APP SDI-ST2110-2D Gateway - ST 2110 to SDI HD/3G Dual De-Encapsulator SFP Gateway. HW v100, SW v3.3
 - <https://www.riedel.net/products-solutions/distributed-video-networks/mediornet-muo-n/hardware/>
- RNR: Bridge Technologies VB440
 - <https://bridgetech.tv/products/vb440>

Network switches used during the event (only applicable to the NMOS Controllers testing)

- Arista 7060CX2-32 (EOS Version 4.22.3M)
 - https://www.arista.com/assets/data/pdf/Datasheets/7060X_7260X_DS.pdf
- Arista 7280SR48-YC6 (EOS Version 4.22.3M)
 - <https://www.arista.com/assets/data/pdf/Datasheets/7280R-DataSheet.pdf>
- Arista 7280SR2-48YC6 (EOS Version 4.22.3M)
 - <https://www.arista.com/assets/data/pdf/Datasheets/7280R-DataSheet.pdf>
- Arista 7020TR-48 (EOS Version 4.22.3M)
 - https://www.arista.com/assets/data/pdf/Datasheets/7020R-48_Datasheet.pdf

PTP configuration used during the event (only applicable to the NMOS Controllers testing)

The PTP Grandmaster(s) used during the event - Tektronix **SPG8000A**.

The network switches will be configured in a **Boundary Clock** mode.

Multicast communication mode for all messages will be used (except for management TLV responses).

The PTP profile details will be provided at the time of testing. The parameters provided will be within the values allowed by SMPTE ST 2059-2:2015.



General statements and terms (the list was adapted because of the cancellation of the face-to-face event)

- This test plan outlines the principles and methods for the IP Media Endpoints testing which are applied for on-site testing at the face-to-face events, but can and should also be applied for self-testing as well as independent testing. Therefore, except for the JT-NM face-to-face events, it is never expected that the same test and measurement, network, PTP or reference equipment is used for self-testing or independent testing.
- Vendors may not change software/firmware once testing has commenced but may adjust settings on their products as necessary during the testing process.
- The Test Plan team reserves the right to update the “JT-NM Tested Program” Test Plan as required.

Media Node Tests

This set of tests validates a Media Node according to the requirements of JT-NM TR-1001-1. Where implemented, Media Nodes will also be tested against the AMWA NMOS IS-08 specification.

1. Media Node Startup Behaviour

1.1 DHCP

Description: The management interface(s) and media interface(s) of the Device Under Test (DUT) will be connected to network ports of appropriate speed and media type. The DUT shall participate in a DHCP negotiation with the DHCP server on each network connection. In addition to host address and CIDR mask, the DHCP server will offer a gateway route and DNS server information to each interface. A large enough DHCP pool will be used in order to ensure that each device gets a unique IP address. This test is expected to be carried out in a shared network environment.

Validation method: The DHCP server will be examined to find the addresses assigned to the DUT. The MAC address associated with the DHCP request will also be noted. ICMP Echo-Requests (Ping) will be sent from another subnet to each of the allocated addresses. Vendor-specific methods, if available, will also be used to examine the assigned address information.

Pass criteria:

- The MAC addresses in the DHCP server for each assigned address shall be unique. If vendor-specific means were available to determine the assigned addresses, these address details will be checked against the DHCP server entries.



- The DUT shall return the ICMP echo-request that is sent to each of the DHCP-assigned interface addresses. This will validate that the gateway route is used.

1.2 LLDP

Description: The management interface(s) and media interface(s) of the Device Under Test (DUT) will be connected to network ports of appropriate speed and media type. The DUT shall supply the LLDP mandatory TLV items to the attached switch on each interface. This test is expected to be carried out in a shared network environment.

Validation method: The LLDP information tables on the attached switches will be examined to find the information from the DUT. The presence of mandatory TLV items is checked. The reported MAC addresses will be matched to the MAC addresses in the DHCP server, and the reported chassis-ID and port-ID information noted.

Pass criteria:

- The LLDP mandatory TLV items are provided. These are listed in Table 8.1 of IEEE 802.1AB-2016, and are: chassis-ID, port-ID, TTL.
- The LLDP-reported port-ID information shall be unique for each interface on the DUT.

1.3 System Resource Params*

* Note: The PTP lock status of the device will not be checked after performing this test as it would require disruptive network changes. Passes are marked based upon observation of correct configuration changes in vendors' user interfaces.

Description: The management interface(s) and media interface(s) of the Device Under Test (DUT) will be connected to network ports of appropriate speed and media type. A JT-NM TR1001-1 System Resource shall be available on the network, reachable from each network segment, and advertised in the DNS server via DNS-SD. The DUT shall have already passed the DHCP test above or otherwise be manually configured with Host, CIDR, Gateway, and DNS server information. The DUT shall use unicast DNS Service Discovery (DNS-SD) to locate the System Resource (JT-NM TR1001-1 Appendix A). The DUT shall determine the prevailing PTP domain ID from the system resource and apply it to its running configuration without any user intervention. This test is expected to be carried out in an isolated network environment.

Validation method: Vendor shall provide some reasonable method to observe the prevailing PTP domain and the lock status of PTP locking to that domain.

Pass criteria:

- The vendor-specific status shall indicate that the device is locked to the correct PTP domain.

1.4 Using DNS-SD to find the IS-04 Registry

Description: The management interface(s) and media interface(s) of the Device Under Test (DUT) will be connected to network ports of appropriate speed and media type. An AMWA IS-04 registration service supporting v1.2 and later shall be available on the network, reachable from



each network segment, and advertised in the DNS server via DNS-SD. The DUT shall have already passed the DHCP test above or otherwise be manually configured with Host, CIDR, Gateway, and DNS server information. The DUT shall use unicast DNS Service Discovery (DNS-SD) to locate the IS-04 registration service. In order to avoid any issues with the length of the service name, the registry will be advertised identically under the service names `_nmos-registration` and `_nmos-register` for IS-04 v1.2. The DUT shall register into the best priority registry as determined in accordance with IS-04. This test is expected to be carried out in an isolated network environment.

Validation Method: The (potentially multiple advertised) registries will be examined to determine which registry the DUT has registered into (if any).

Pass Criteria:

- The DUT registration shall be in the correct registry.

2. Media Nodes & IS-04

Description: This set of tests validates a Media Node's ability to interact with an IS-04 registry, with discovery achieved via unicast DNS. Tests will be carried out using the NMOS Testing Tool. See the Appendix for usage instructions. These tests are expected to be carried out in an isolated network environment.

First steps: The management interface(s) and media interface(s) of the Device Under Test (DUT) will be connected to network ports of appropriate speed and media type. The DuT will be configured with a DNS server IP address and a DNS search domain, either via DHCP or via manual configuration. These DNS addresses will match those required by the NMOS testing tool. The testing tool will be configured to use a 'DNS_SD_MODE' of 'unicast' and a 'DNS_SD_ADVERT_TIMEOUT' sufficient to cover the DuT's boot time. The vendor will be asked to provide the IP address(es) and port which the IS-04 Node API is running on.

The following 'test suites' within the NMOS testing tool will be used to enable these tests:

- IS-04 Node API

Media Nodes will be tested using the highest version of IS-04 which they support, with a minimum version of v1.2.

The following tests will not be used as they cover deprecated or unnecessary items for TR-1001:

- test_01: Node can discover network registration service via multicast DNS
- test_01_01: Node does not attempt to register with an unsuitable mDNS registry
- test_12: Node publishes its own mDNS announcement
- test_12_01: Node does not advertise a Node type mDNS announcement in the presence of a Registration API (v1.3+)



- test_13: Node supports IS-04 native connection management (deprecated)
- test_14: Node supports IS-04 native connection parking (deprecated)
- test_17_01: IS-04 Device correctly indicates linked Senders and Receivers (deprecated)

2.1 Schema Conformance

Description: Tests that the Media Node's IS-04 API exposes all of the expected resources, and that each of these conforms to the JSON schema included in the specification.

Validation method: Execution of the NMOS testing tool, using the following tests:

- All auto_tests: Schema validation of GET requests

Pass criteria:

- All tests indicate a 'PASS' state.

2.2 API Behaviour

Description: Tests that a Media Node correctly implements various behavioural aspects of the IS-04 specification, including the BCP-002-01 grouping syntax where applicable.

Validation method: Execution of the NMOS testing tool, using the following tests:

- test_17: All Node resources use different UUIDs
- test_18: All Node clocks are unique, and relate to any visible Sources' clocks
- test_19: All Node interfaces are unique, and relate to any visible Senders and Receivers' 'interface_bindings'
- test_20: Node's resources correctly signal the current protocol and IP/hostname
- test_23: Senders and Receivers correctly use BCP-002-01 grouping syntax
- test_24: Periodic Sources specify a 'grain_rate'
* Note: Warnings will be marked as a fail for ST.2110 video and audio as these are expected to be periodic
- test_24_01: Periodic Flows' 'grain_rate' is divisible by their parent Source 'grain_rate'
* Note: Warnings will be marked as a pass for this test, but including a grain_rate for periodic Flows is strongly recommended
- test_25: Receivers expose expected 'caps' for their API version
* Note: Warnings will be marked as a fail unless the device accepts all media_types (for example a generic stream analyser)
- test_26: Source 'format' matches Flow 'format'

Pass criteria:

- All tests indicate a 'PASS' state.

2.3 Unicast Discovery

Description: Tests that a Media Node is capable of automatically locating the IS-04 registry service using unicast DNS service discovery. This tests the same behaviour as 1.4, but is a prerequisite for further tests in this section. Test 1.4 will be carried out in a shared network segment, whereas this test will be carried out in an isolated network segment.

Validation method: Execution of the NMOS testing tool, using the following tests:

- test_02: Node can discover network registration service via unicast DNS

Pass criteria:



- All tests indicate a 'PASS' state.

2.4 Basic Registration

Description: Tests that a Media Node is capable of registering all of its advertised resources with an IS-04 registry service. Additionally requires that Nodes maintain themselves in the registry by using the heartbeat mechanism using the default interval of 5 seconds as specified in IS-04 (the system resource IS-04 heartbeat_interval will also be set to 5 seconds).

Validation method: Execution of the NMOS testing tool, using the following tests:

- test_03: Registration API interactions use the correct Content-Type
 - * Note: Warnings about 'charset' will be marked as a pass for this test.
- test_03_01: Registration API interactions use the correct versioned path
- test_04: Node can register a valid Node resource with the network registration service, matching its Node API self resource
- test_05: Node maintains itself in the registry via periodic calls to the health resource
 - * Note: Warnings about inclusion of a Content-Type header will be marked as a pass for this test.
- test_07: Node can register a valid Device resource with the network registration service, matching its Node API Device resource
- test_07_01: Registered Device was POSTed after a matching referenced Node
- test_08: Node can register a valid Source resource with the network registration service, matching its Node API Source resource
- test_08_01: Registered Source was POSTed after a matching referenced Device
- test_09: Node can register a valid Flow resource with the network registration service, matching its Node API Flow resource
- test_09_01: Registered Flow was POSTed after a matching referenced Device or Source
- test_10: Node can register a valid Sender resource with the network registration service, matching its Node API Sender resource
- test_10_01: Registered Sender was POSTed after a matching referenced Device
- test_11: Node can register a valid Receiver resource with the network registration service, matching its Node API Receiver resource
- test_11_01: Registered Receiver was POSTed after a matching referenced Device
- test_21: Node correctly interprets a 200 code from a registry upon initial registration

Pass criteria:

- All tests indicate a 'PASS' state.

2.5 Advanced Registration

Description: Tests that a Media Node is capable of handling the presence of multiple clustered registries on a network. It should be able to demonstrate selection of the preferred registry using the defined priority order, and demonstrate an ability to failover between the available instances in the event that one registry indicates an error state.

Validation method: Execution of the NMOS testing tool, using the following tests:



- test_02_01: Node does not attempt to register with an unsuitable unicast DNS registry
- test_15: Node correctly selects a Registration API based on advertised priorities
- test_16: Node correctly fails over between advertised Registration APIs when one fails
- test_16_01: Node correctly handles Registration APIs whose connections time out

Pass criteria:

- All tests indicate a 'PASS' state.

2.6 Consistency of UUIDs

Description: This test validates that Media Nodes expose the same UUID values after a reboot or restart event, even when the Media Node has moved to a new network environment.

Validation method: During initial setup in the “zoo” environment, with the consent of the unit under test, a snapshot will be taken of the UUIDs for the Node and its Devices, Senders, and Receivers. In the later “clean room” test, when the Media Node is on a different subnet, attached to a different registry, the UUIDs will be compared against the previous snapshot. In order to assist with this test, the UUID Checker tool from

<https://github.com/AMWA-TV/nmos-testing/tree/master/utilities/uuid-checker> will be used.

Pass Criteria:

- The UUID values for Node, Devices, Senders, and Receivers shall be the same, for elements of the Media Node which are in the same configuration.

3. Media Nodes & IS-05

Description: This set of tests validates a Media Node’s ability to be controlled by a Broadcast Controller using the IS-05 Connection Management API. Tests will be carried out using the NMOS Testing Tool. See the Appendix for usage instructions. These tests are expected to be carried out in a shared network environment.

First steps: The management interface(s) and media interface(s) of the Device Under Test (DUT) will be connected to network ports of appropriate speed and media type. The vendor will be asked to provide the IP address(es) and port(s) which the IS-04 Node API and IS-05 Connection Management API are running on. The vendor will be asked to configure their device as necessary, with the option to have the test run with or without ST.2022-7 support enabled (if this option is available). The tests will not be run twice to cover both modes due to time constraints.

The following ‘test suites’ within the NMOS testing tool will be used to enable these tests:

- IS-05 Connection Management API (used by 3.1 through 3.7 below)
- IS-05 Interaction with IS-04 (used by 3.8 only below)

Media Nodes will be tested using the highest versions of IS-04 and IS-05 which they support, with a minimum of v1.2 and v1.0 respectively.

3.1 Schema Conformance

Description: Tests that the Media Node's IS-05 API exposes all of the expected resources, and that each of these conforms to the JSON schema included in the specification.

Validation method: Execution of the NMOS testing tool, using the following tests:

- All auto_ tests: Schema validation of GET requests

Pass criteria:

- All tests indicate a 'PASS' state.

3.2 API Behaviour

Description: Tests that the IS-05 API does not expose an invalid configuration, and that violations of the advertised constraints are not permitted.

Validation method: Execution of the NMOS testing tool, using the following tests:

- test_09: All params listed in /single/senders/{senderId}/constraints/ matches /staged/ and /active/
- test_10: All params listed in /single/receivers/{receiverId}/constraints/ matches /staged/ and /active/
- test_11: Senders are using valid combination of parameters
- test_11_01: Sender /active parameters do not use the keyword 'auto'
- test_12: Receiver are using valid combination of parameters
- test_12_01: Receiver /active parameters do not use the keyword 'auto'
- test_13: Return of /single/senders/{senderId}/staged/ meets the schema
- test_14: Return of /single/receivers/{receiverId}/staged/ meets the schema
- test_31: Sender active response schema is valid
- test_32: Receiver active response schema is valid
- test_40: Only valid transport types for a given API version are advertised

Pass criteria:

- All tests indicate a 'PASS' state.

3.3 Single Sender Routing

Description: Tests that a Media Node with a ST.2110 transmission capability can have its configuration managed via the IS-05 'single' resource.

Validation method: Execution of the NMOS testing tool, using the following tests:

- test_15: Staged parameters for senders comply with constraints
- test_17: Sender patch response schema is valid
- test_19: Sender invalid patch is refused
- test_22: Receiver id on staged sender is changeable
- test_23: Sender transport parameters are changeable
- test_25: Immediate activation of a sender is possible



- test_38: Number of legs matches on constraints, staged and active endpoint for senders
- test_41: SDP transport files pass SDPoker tests
 - * Note: Whitespace warnings will be marked as a pass for this test. This result is excluded from the March 2020 results due to a bug in the NMOS Testing Tool. Please refer to the ST.2110 test results for SDP file testing.
- test_42: Transport files use the expected Content-Type
 - * Note: This result is excluded from the March 2020 results due to a mis-interpretation of testing conditions.

Pass criteria:

- All tests indicate a 'PASS' state.

3.4 Single Receiver Routing

Description: Tests that a Media Node with a ST.2110 reception capability can have its configuration managed via the IS-05 'single' resource.

Validation method: Execution of the NMOS testing tool, using the following tests:

- test_16: Staged parameters for receivers comply with constraints
- test_18: Receiver patch response schema is valid
- test_20: Receiver invalid patch is refused
- test_21: Sender id on staged receiver is changeable
- test_24: Receiver transport parameters are changeable
- test_26: Immediate activation of a receiver is possible
- test_39: Number of legs matches on constraints, staged and active endpoint for receivers

Pass criteria:

- All tests indicate a 'PASS' state.

3.5 Scheduled Activation

Description: Tests that the Media Node permits activation of new IS-05 transmission or reception parameters after a relative time offset or at an absolute time instant. This is in addition to immediate activations which are tested in 4.2 and 4.3.

Validation method: Execution of the NMOS testing tool, using the following tests:

- test_27: Relative activation of a sender is possible
- test_28: Relative activation of a receiver is possible
- test_29: Absolute activation of a sender is possible
- test_30: Absolute activation of a receiver is possible

Pass criteria:

- All tests indicate a 'PASS' state.

3.6 Bulk Sender Routing

Description: Tests that a Media Node with a ST.2110 transmission capability can have its configuration managed via the IS-05 'bulk' resource, which allows multiple Senders to be re-configured at the same time.



Validation method: Execution of the NMOS testing tool, using the following tests:

- test_34: GET on /bulk/senders returns 405
- test_36: Bulk interface can be used to change destination port on all senders

Pass criteria:

- All tests indicate a 'PASS' state.

3.7 Bulk Receiver Routing

Description: Tests that a Media Node with a ST.2110 reception capability can have its configuration managed via the IS-05 'bulk' resource, which allows multiple Receivers to be re-configured at the same time.

Validation method: Execution of the NMOS testing tool, using the following tests:

- test_35: GET on /bulk/receivers returns 405
- test_37: Bulk interface can be used to change destination port on all receivers

Pass criteria:

- All tests indicate a 'PASS' state.

3.8 Integration with IS-04

Description: Tests that the IS-05 Connection Management API is correctly advertised via the IS-04 Node API, and that the Senders and Receivers advertised in IS-05 match those in IS-04. This test also ensures that when the IS-05 configuration is updated, the matching IS-04 resources are updated to reflect any changes to their state (parked/unparked) and which remote Sender or Receiver they are currently connected to.

Validation method: Execution of the NMOS testing tool, using the following tests:

- test_01: Check that version 1.2 or greater of the Node API is available
- test_02: At least one Device is showing an IS-05 control advertisement matching the API under test
- test_03: Receivers shown in Connection API matches those shown in Node API
- test_04: Senders shown in Connection API matches those shown in Node API
- test_05: Activation of a receiver increments the version timestamp
- test_06: Activation of a sender increments the version timestamp
- test_07: Activation of a receiver from an NMOS sender updates the IS-04 subscription
- test_08: Activation of a receiver from a non-NMOS sender updates the IS-04 subscription
- test_09: Activation of a sender to a multicast address updates the IS-04 subscription
- test_10: Activation of a sender to a unicast NMOS receiver updates the IS-04 subscription
- test_11: Activation of a sender to a unicast non-NMOS receiver updates the IS-04 subscription
- test_12: IS-04 interface bindings array matches length of IS-05 transport_params array
- test_13: IS-04 manifest_href matches IS-05 transportfile
- test_14: IS-05 transportfile rtpmap parameters match IS-04 Source and Flow
- test_15: IS-05 transportfile fmtp parameters match IS-04 Source and Flow



- test_16: IS-05 transportfile optional fmtp parameters match IS-04 Source and Flow
- test_17: IS-05 transportfile ts-refclk matches IS-04 Source and Node
- test_18: Receiver correctly translates SDP file attributes into transport_params

Pass criteria:

- All tests indicate a 'PASS' state.

4. Media Node Streams

Description: This set of tests validates that IS-05 usage on the DuT matches the production or consumption of real ST.2110 streams on the network. These tests are expected to be carried out in a shared network environment. All media streams shall remain confined to the media portion of the IP network. Testing will be performed using the IS-05 Control utility as part of the NMOS Testing Tool: <https://github.com/AMWA-TV/nmos-testing/tree/master/utilities/is-05-control>

First steps:

The management interface(s) and media interface(s) of the Device Under Test (DUT) will be connected to network ports of appropriate speed and media type. At the time of testing the vendor should provide the following:

- A sender and/or receiver (adhering to ST.2110-20/30/40) registered in the provided IS-04 registry.
- The DuT is expected to support IGMPv3 and be configured to use it if this is not automatic.

4.1 Configuring Senders and successful reception of configured ST.2110-20/30/40 stream via IS-05

A reference receiver that supports 2022-7 will be used to validate the sender configuration. If DuTs support a 2022-7 and non-2022-7 mode, the participant may choose which one of these is tested. If the DuT supports video, audio and ANC only one of these will be tested.

Description: The Test validates that IS-05 transport parameters can be used to successfully set a multicast destination on the sender. The following steps will be done in IS-05 single mode. Where a 2022-7 capable sender is being tested, the same checks and configuration will be performed for both the 'primary' and 'secondary' streams.

Validation method:

- DuT starts with a "no streaming" state. Active state should return 'master_enable' as false.
- Validate that the reference receiver can not receive the stream at the configured multicast destination.
- Stage and activate the DuT sender with one multicast destination per stream with 'master_enable' set to true and 'rtp_enabled' set to true.
- Validate that the reference receiver can now receive the stream at the configured multicast destination.

- Stage and activate the DuT sender with 'master_enable' set to false.
- Validate that the stream has stopped on the reference receiver.

Pass criteria:

- The DuT accepts the configuration and can be seen to send a stream using the correct multicast configuration.
- The DuT could be successfully enabled and disabled using the 'master_enable' flag.

4.2 Configuring Receivers and successful reception of a ST.2110-20/30/40 stream using an SDP file via IS-05

A reference sender that supports ST.2022-7 will be used to validate the receiver configuration. If DuTs support a 2022-7 and non-2022-7 mode, the participant may choose which one of these is tested. If the DuT supports video, audio and ANC only one of these will be tested.

Description: The Test validates that IS-05 can be used to assign a configuration to a receiver using the 'transport_file' parameter. The following steps will be done in IS-05 single mode. Where a 2022-7 capable receiver is being tested, the same checks and configuration will be performed for both the 'primary' and 'secondary' streams.

Validation method:

- DuT starts with a "no streaming" state. Active state should return 'master_enable' as false.
- Validate that the DuT receiver is not receiving anything.
- Stage and activate the DuT receiver by providing the reference sender's SDP in the 'transport_file' staging. 'master_enable' will be set to true during this step.
- Validate that the DuT receiver can now receive the stream at the multicast destination.
- Stage and activate the DuT receiver with 'master_enable' set to false.
- Validate that the stream has stopped on the DuT receiver.

Pass criteria:

- The DuT accepts the configuration and can be seen to receive the stream.
- The DuT could be successfully enabled and disabled using the 'master_enable' flag.

4.3 IGMPv3 including source addresses

Description: The Test validates that IS-05 can be used to assign a 'source_ip' to a receiver by using SDP files containing multicast source-filters. The receiver should use IGMPv3 in order to select the right source.

Validation method:

- Validate that the receiver is not receiving anything.
- Stage and activate the receiver using the SDP transport_file of a reference sender, including a valid source-filter line.



- Validate that the IGMP snooping table in the switch indicates that the source IP address from the SDP file was included in the IGMPv3 request.
- Validate that the receiver can now receive the stream at the multicast destination.

Pass criteria:

- The DuT sends a valid IGMPv3 request for the requested multicast group and can be seen to receive the stream.
- The IGMPv3 request includes a filter matching the source IP address.

4.4 SDP files including source-filters in senders

Description: The Test validates that usage of source-filters in SDPs match the stream being sent by the sender.

Validation method:

- Get the SDP file from the DuT using the IS-05 /transportfile resource.
- Validate that the stream generated by the DuT is sending packets from a source IP address which matches the SDP source-filter line provided by the sender.

Pass criteria:

- The DuT is providing an SDP with a source-filter present.
- The source-filter line(s) match the source interface IP addresses.

5. Media Nodes & IS-08

Description: This set of tests validates a Media Node's ability to re-map audio channels either prior to sending a stream onto the network, or after receiving one. Tests will be carried out using the NMOS Testing Tool. See the Appendix for usage instructions. These tests are expected to be carried out in a shared network environment.

First steps: The management interface(s) and media interface(s) of the Device Under Test (DUT) will be connected to network ports of appropriate speed and media type. The vendor will be asked to provide the IP address(es) and port(s) which the IS-04 Node API and IS-08 Audio Channel Mapping API are running on. The vendor will be asked to configure their device as necessary.

The following 'test suites' within the NMOS testing tool will be used to enable these tests:

- IS-08 Audio Channel Mapping API (used by 5.1 through 5.3 below)
- IS-08 Interaction with IS-04 (used by 5.4 only below)

Media Nodes will be tested using the highest versions of IS-04 and IS-08 which they support, with a minimum of v1.2 and v1.0 respectively.

5.1 Schema Conformance

Description: Tests that the Media Node's IS-08 API exposes all of the expected resources, and that each of these conforms to the JSON schema included in the specification.

Validation method: Execution of the NMOS testing tool, using the following tests:

- All auto_ tests: Schema validation of GET requests

Pass criteria:

- All tests indicate a 'PASS' state.

5.2 API Behaviour

Description: Tests that the IS-08 API does not expose an invalid configuration, and that violations of the advertised constraints are not permitted.

Validation method: Execution of the NMOS testing tool, using the following tests:

- test_01: Content of the /io view matches resources elsewhere in the API
- test_07: Channels in the active resource where no input channel is routed have `null` set as the `input` and `channel_index`
- test_08: If the device allows re-entrant matrices, the constraints are set such that it is not possible to create a loop
- test_11: Inputs have at least one channel represented in their channels resource
- test_12: Outputs have at least one channel represented in their channels resource
- test_13: Attempting to violate routing constraints results in an HTTP 400 response
- test_14: It is not possible to re-order channels when re-ordering is set to `false`
- test_15: It is not possible to make an out-of-block route when block_size is anything other than 1

Pass criteria:

- All tests indicate a 'PASS' state.

5.3 Activations

Description: Tests that the Media Node permits activation of new IS-08 parameters either immediately, or after a time offset.

Validation method: Execution of the NMOS testing tool, using the following tests:

- test_02: Immediate activation can be called on the API
- test_03: Relative offset activations can be called on the API
- test_04: Absolute offset activations can be called on the API
- test_05: Activations can be deleted once created
- test_06: Attempting to change a locked route results in a 423 response

Pass criteria:

- All tests indicate a 'PASS' state.

5.4 Integration with IS-04

Description: Tests that the IS-08 Audio Channel Mapping API is correctly advertised via the IS-04 Node API, and that the Source and Receiver IDs advertised in IS-08 match those in IS-04. This test also ensures that when the IS-08 configuration is updated, the matching IS-04 resources are updated to reflect any changes to their state.

Validation method: Execution of the NMOS testing tool, using the following tests:

- test_01: Activations result in a Device version number increment
- test_02: API is correctly advertised as a control endpoint
- test_03: All Output Source IDs match up to the IS-04 Node API
- test_04: All Input Source/Receiver IDs match up to the IS-04 Node API

Pass criteria:

- All tests indicate a 'PASS' state.

NMOS Registry Tests

This set of tests validates an IS-04 registry under test to ensure that it can correctly handle interactions with Media Nodes and controllers.

NOTE: In order to participate, registries must expose both an IS-04 Registration and Query API, operating with the default heartbeat and garbage collection intervals as specified. It is recognised that other valid implementations exist (typically a Registration API paired directly with a controller), but in order to perform consistent and thorough testing these are not being tested at this event.

6. IS-04 Registries

Description: This set of tests validates the behaviour of an IS-04 Registry implementation. Tests will be carried out using the NMOS Testing Tool. See the Appendix for usage instructions.

First steps: Any required network interface(s) of the Device Under Test (DUT) will be connected to network ports of appropriate speed and media type. The vendor will be asked to provide the IP address(es) and port(s) which the IS-04 Registration API and IS-04 Query API are running on. The vendor will be asked to configure their device as necessary.

The following 'test suites' within the NMOS testing tool will be used to enable these tests:

- IS-04 Registry APIs

IS-04 registries will be tested using the highest API version which they support, with a minimum supported version of v1.2 expected for this event.



The following tests will not be used as they cover unnecessary items for TR-1001:

- test_01: Registration API advertises correctly via mDNS
- test_02: Query API advertises correctly via mDNS

6.1 Schema Conformance

Description: Tests that the registry's IS-04 APIs expose all of the expected endpoints, and that each of these conforms to the JSON schema included in the specification.

Validation method: Execution of the NMOS testing tool, using the following tests:

- All auto_tests: Schema validation of GET requests

Pass criteria:

- All tests indicate a 'PASS' state.

6.2 Registration

Description: Tests that the IS-04 Registration API behaves as expected, accepting valid registrations from Nodes, and rejecting invalid registrations. Ensures that the 'garbage collection' mechanism specified as part of the registry operates correctly and removes stale data from the database after a default timeout period.

Validation method: Execution of the NMOS testing tool, using the following tests:

- test_03: Registration API accepts and stores a valid Node resource
- test_03_01: Registration API responds with correct Location header
- test_03_02: Registration API accepts and stores a valid Node resource containing unicode characters
- test_04: Registration API rejects an invalid Node resource with a 400 HTTP code
- test_05: Registration API accepts and stores a valid Device resource
- test_06: Registration API rejects an invalid Device resource with a 400 HTTP code
- test_07: Registration API accepts and stores a valid Source resource
- test_08: Registration API rejects an invalid Source resource with a 400 HTTP code
- test_09: Registration API accepts and stores a valid Flow resource
- test_10: Registration API rejects an invalid Flow resource with a 400 HTTP code
- test_11: Registration API accepts and stores a valid Sender resource
- test_11_1: Registration API accepts and stores a valid Sender resource with null flow_id
- test_12: Registration API rejects an invalid Sender resource with a 400 HTTP code
- test_13: Registration API accepts and stores a valid Receiver resource
- test_14: Registration API rejects an invalid Receiver resource with a 400 HTTP code
- test_15: Registration API responds with 200 HTTP code on updating a registered Node
- test_16: Registration API responds with 200 HTTP code on updating a registered Device
- test_17: Registration API responds with 200 HTTP code on updating a registered Source
- test_18: Registration API responds with 200 HTTP code on updating a registered Flow
- test_19: Registration API responds with 200 HTTP code on updating a registered Sender
- test_20: Registration API responds with 200 HTTP code on updating a registered Receiver

- test_26: Registration API responds with 400 HTTP code on posting a resource without parent
- test_27: Registration API cleans up Nodes and their sub-resources when a heartbeat doesn't occur for the duration of a fixed timeout period
- test_28: Registry removes stale child-resources of an incorrectly unregistered Node
- test_30: Registration API accepts heartbeat requests for a Node held in the registry
- test_32: Registration API generates 409 code when a conflicting registration exists

Pass criteria:

- All tests indicate a 'PASS' state.

6.3 Basic Querying

Description: Tests that the IS-04 Query API behaves as expected, specifically ensuring that it is possible to create and delete WebSocket subscriptions, and that these return the correct data as it is created via the Registration API.

Validation method: Execution of the NMOS testing tool, using the following tests:

- test_29: Query API supports websocket subscription request
- test_29_1: Query API websocket subscription requests default to the current protocol
- test_31: Query API sends correct websocket event messages for UNCHANGED (SYNC), ADDED, MODIFIED and REMOVED

Pass criteria:

- All tests indicate a 'PASS' state.

6.4 Advanced Querying

Description: Tests that the optional features of an IS-04 Query API are present and behave as expected. Ensures that basic and advanced querying, pagination, downgrade and ancestry queries meet the specification where implemented.

Validation method: Execution of the NMOS testing tool, using the following tests:

- test_21_1: Query API implements pagination (no query or paging parameters)
- test_21_1_1: Query API implements pagination (when explicitly requested)
- test_21_2: Query API implements pagination (documentation examples)
- test_21_3: Query API implements pagination (edge cases)
- test_21_4: Query API implements pagination (requests that require empty responses)
- test_21_5: Query API implements pagination (filters that select discontinuous resources)
- test_21_6: Query API implements pagination (bad requests)
- test_21_7: Query API implements pagination (updates between paged requests)
- test_21_8: Query API implements pagination (correct encoding of URLs in Link header)
- test_21_9: Query API implements pagination (correct protocol and IP/hostname in Link header)
- test_22: Query API implements downgrade queries
- test_22_1: Query API subscriptions resource does not support downgrade queries



- test_22_2: Query API WebSockets implement downgrade queries
- test_23: Query API implements basic query parameters
- test_23_1: Query API WebSockets implement basic query parameters
- test_24: Query API implements RQL
- test_24_1: Query API WebSockets implement RQL
- test_25: Query API implements ancestry queries

Pass criteria:

- All tests indicate a 'PASS' state.

NMOS Controller Tests

This set of tests validates a Broadcast Controller under Test's (BCuT) ability to query an IS-04 Registry with the IS-04 Query API and to control a Media Node using the IS-05 Connection Management API.

NOTE: In order to participate, controllers must be capable of discovering and operating with an IS-04 Query API. Controllers which are tightly coupled to a specific registry implementation cannot be tested. It is recognised that other valid implementations exist (typically a Registration API paired directly with a controller), but in order to perform consistent and thorough testing these are not being tested at this event.

Tests will be executed by manually examining:

- the Registry,
- the BCuT's User Interface and
- the behavior of the Reference Receiver.

These tests are expected to be carried out in a shared network environment.

Where controllers are tested to ensure they respond to changes in the IS-04 registry, they will be expected to indicate any changes within 30 seconds. This is based upon a perceived user expectation as opposed to a figure from a specification. In practice, controllers which make use of the preferred WebSocket connection mechanism should be able to respond far quicker than this.

First steps: The management interface of the BCuT will be connected to the management network at the appropriate speed and media type. An AMWA IS-04 v1.2/1.3 reference registration service shall be available on the network, reachable from each network segment, and advertised in the DNS server via unicast DNS-SD. The vendor will be asked to configure any other parameter on their device as necessary. The vendor will be asked to provide the IP address(es) and port which the NMOS controller is running on (if web accessible), or access to a graphical user interface via other means.

7. NMOS Controllers & Integration with the IS-04 Query Service

Description: The BCuT should be capable of browsing an IS-04 Registry in order to obtain a list of online resources and connection information. Browsing of the Query API may be performed via the WebSocket protocol which is preferred for larger scale applications, or the RESTful API. Tests will be performed based on the BCuT's selected method of browsing the Registry. Where the RESTful API is used, the BCuT must correctly implement pagination in order to pass the tests.

First steps: The BCuT will be configured with a DNS server IP address and a DNS search domain, either via DHCP or via manual configuration by the vendor.

7.1 Using unicast DNS-SD to find the IS-04 Registry

Description: The BCuT shall use unicast DNS Service Discovery (DNS-SD) to locate the IS-04 Reference Registry. A DHCP server will provide all the connection details to discover the Registry. Alternatively, manual configuration of the BCuT's IP address, CIDR, Gateway, and DNS server information will be permitted, but is less desirable.

Validation Method: The BCuT's configuration interface will be observed to ensure that manual configuration of an IS-04 registry has not been performed. The DNS server logs and/or a packet capture on the DNS server will be used to confirm that the BCuT has correctly performed queries for the relevant DNS-SD PTR, SRV, TXT and A records. An active connection to a discovered and selected IS-04 Query API instance will be observed in the BCuT's UI.

Pass Criteria:

- The BCuT's UI is able to browse the Registry.

7.2 Discovering registered resources via the IS-04 Query API

Description: The Query API should be able to discover *all* the devices that are registered in the Registry. If using the RESTful API rather than WebSockets, Pagination must be implemented [see reference](#).

Validation Method: The RNS and RNR will be confirmed as online and properly configured. Their presence in the Registry will be confirmed by browsing the Riedel NMOS Explorer before checking in the BCuT.

Pass Criteria:

- The BCuT shall display the RNS and the RNR along with their advertised senders and receivers in its UI.

7.3 Updating the list of resources dynamically from the IS-04 Registry

Description: A BCuT is able to update its list of resources, preferably by dynamic WebSocket connection (if supported) or by polling the Registry with the RESTful API. This test confirms this behaviour by disabling the NMOS network interface on the RNS and inspecting the UI for updates.

Validation Method:

1. Reference Sender is put offline (e.g., port shut) and the change is confirmed using both the Riedel NMOS Explorer and the BCuT.
2. Then Reference Sender is put back online (e.g., port back up) and the change is confirmed using both the Riedel NMOS Explorer and the BCuT.

Pass Criteria:

- Changes in presence/absence of Reference Sender shall be reflected in the BCuT's UI after a maximum of 30 seconds. Where the design of the UI requires a button press to display changes this will be permitted.

7.4 Discovering a Receiver's Connection Management API capability via the Query API

Description: The BCuT shall be able to identify which Receiver devices are controllable via IS-05 and which are not based upon the advertisement of a relevant IS-04 Device 'control' entry. Devices which omit the relevant 'control' must not be listed as available for control via IS-05.

Validation Method: A Node will be made available via the registry which does not expose the relevant IS-05 control type. The lack of a capability to control this Node via the Connection Management API will be confirmed in the Riedel NMOS Explorer.

Pass Criteria: The BCuT's UI indicates that the Reference Receiver has the capability to be connected to a Sender, but the mock Node which omits this capability will not be listed as controllable.

8. NMOS Controllers & IS-05 Connection Management

Description: The BCuT shall use the IS-05 Connection Management API to establish (and remove) connections between Senders and Receivers.

BCuTs should preferably use 'bulk' or may use the 'single' method in IS-05. The test results will not indicate which method was used. The selected RNS and RNR will support both mechanisms.

First steps: BCuT should establish an IS-04 Query API connection with the RRDS and have discovered the RNS and RNR (section 7). If the IS-04 Query API cannot be used for any reason, the RNS and RNR may be manually entered into the BCuT.

8.1 Instructing a Receiver to subscribe to a Sender's Flow via IS-05

Description: All flows that are available in the RNS should be able to be connected to the RNR.

Validation method: The RNR will initially be connected to a different video and audio Flow, not offered by the RNS. A video and audio flow will then be connected from the RNS to the RNR using the BCuT's UI. The BCuT will be expected to pass the SDP file from the RNS to the RNR in the IS-05 'transport_file' attribute, with 'master_enable' set to true. An 'immediate' activation should be performed by the BCuT.

Pass criteria:

- The RNR's IS-05 API /active endpoint indicates the correct 'transport_file' and a 'master_enable' state of true.
- The RNR's SDI out has the video and audio source signals from the RNS
- BCuT indicates a successful connection via updates received from the API

8.2 Disconnecting a Receiver from a connected Flow via IS-05

Description: IS-05 provides a mechanism for removing an active connection through its API. This test seeks to validate this operation on the BCuT UI.

Validation Method:

1. Establish a connection between the RNS and RNR as described in section 8.1.
2. Disable the connection from the BCuT UI. The BCuT will be expected to set the 'master_enable' attribute to false in the RNR. An 'immediate' activation should be performed by the BCuT.

As-per JT-NM TR-1001-1, BCuTs will not be penalised for setting the 'multicast_ip' to '0.0.0.0' in either the 'transport_params' or the 'transport_file', or for setting the 'transport_file' 'data' attribute to null or an empty string. It is important however that the 'master_enable' be set correctly.

Pass Criteria:

- The RNR's IS-05 API /active endpoint indicates a 'master_enable' state of false.
- The video/audio Flow present on the RNR will stop being received.
- BCuT UI updates its status to indicate RNR is disconnected.

8.3 Indicating the state of connections via updates received from the IS-04 Query API

Description: The Query API is able to monitor and update the connection status of all registered devices. This test seeks to validate the BCuT's ability to monitor connections that are made between Senders and Receivers outside on the BCuT's control.



Validation Method:

1. Ensure that the RNS and RNR have no active connections
2. Establish a connection between the RNS and RNR using the Riedel NMOS Explorer.
3. Disconnect the RNR using the Riedel NMOS Explorer.

Pass Criteria:

- BCuT's UI indicates the current state of connections established by another controller within 30 seconds of a change of state (connected / not connected).



Appendix: Using the NMOS testing tool

The majority of AMWA NMOS testing will be carried out using the automated NMOS testing tool, available from <https://github.com/AMWA-TV/nmos-testing>. If you intend to participate in TR-1001 testing, please use this tool in advance of the event to confirm your Media Node or registry's adherence to the specifications.

The tool can be used to test a range of NMOS specifications, but the following instructions target how to use it for the specific IS-04, IS-05 and IS-08 testing which will be carried out at the JT-NM Tested event. The testing tool is written in Python and is known to work on Linux and Windows.

Detailed documentation for installation and usage of the tool is hosted at <https://amwa-tv.github.io/nmos-testing/>, but the following is intended as a quick start guide.

Installation

- Install Python 3.5 or above onto a computer on the same network as the Media Node (via <https://www.python.org/downloads/> or a Linux package manager)
- Install the Git version control software (via <https://git-scm.com/download> or a Linux package manager)
- Download the testing tool via <https://github.com/AMWA-TV/nmos-testing/archive/master.zip>
- Unzip the downloaded archive into a directory of your choice
- Using the command prompt, change directory into the location of the testing tool (where you will find a file called 'requirements.txt')
- Install the tool's dependencies using Python's 'pip' package manager as follows: 'python -m pip install -r requirements.txt'
- Install the 'SDPoker' NodeJS dependency using the instructions in <https://github.com/AMWA-TV/sdpoker#command-line>

Configuration

By default, the testing tool tests IS-04 Nodes in multicast discovery mode. For the purposes of testing TR-1001 we need to use 'unicast' mode which does not make any multicast DNS advertisements, but does run a mock DNS server on the testing host. Please consult [https://amwa-tv.github.io/nmos-testing/branches/master/docs/2.1.Usage - Testing Unicast Discovery.html](https://amwa-tv.github.io/nmos-testing/branches/master/docs/2.1.Usage-Testing-Unicast-Discovery.html) for instructions on how to configure the testing tool and your Media Node (unit under test) for this mode. If your Node cannot be configured to ignore mDNS announcements you will need to run this testing on a network segment which is isolated from any IS-04 registry mDNS announcements.



Operation

The testing tool can be run by typing 'python nmos-test.py' at the command prompt from the testing tool's working directory. Note that in order to test with unicast DNS you will need to run this command with elevated permissions. On Linux this can be achieved by prefixing the command with 'sudo'. On Windows you will need to open an elevated command prompt (otherwise known as administrator mode).

Having started the testing tool, browse to <http://127.0.0.1:5000> in a web browser on the machine hosting the tool.

The following 'test suites' listed in the dropdown will be used during the JT-NM Tested event for the purposes of IS-04, IS-05 and IS-08 testing. Additional listed 'test suites' may be used if the event includes testing of further AMWA NMOS specifications.

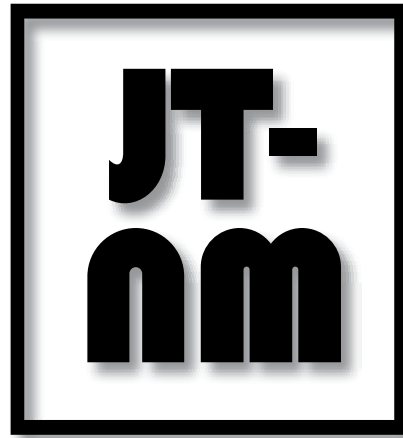
- IS-04 Node API
- IS-04 Registry APIs (registry testing only)
- IS-05 Connection Management API
- IS-05 Interaction with IS-04
- IS-08 Audio Channel Mapping API
- IS-08 Interaction with IS-04
- IS-09 System API Discovery (TR-1001-1 Appendix A testing only)

Select a test suite, and enter the IP address and port of the Node under test into the boxes. Ensure that 'all' is selected in 'Test Selection' and press 'Run'. When the tests complete, press the 'Download JSON' button at the top right of the page to download a copy of the results.

When executing the 'IS-04 Node API' or 'IS-09 System API Discovery' test suite, note that you may need to restart your Media Node after having pressed 'Run' in order to force it to discover the test suite's mock IS-04 registries or IS-09 system resource. As noted in <https://amwa-tv.github.io/nmos-testing/branches/master/docs/2.1. Usage - Testing Unicast Discovery.html>, you can increase the testing tool's 'DNS_SD_ADVERT_TIMEOUT' in order to allow enough time for your Media Node to reboot when performing this test.

Support

If you encounter issues using the testing tool, please create an issue at <https://github.com/AMWA-TV/nmos-testing/issues> and the team will be happy to assist.



Appendix

JT-NM Tested March 2020 – AMWA NMOS / JT-NM TR-1001-1
Tested and Self-Testing Results



Self-Tested Results

For the first time, this catalog includes Self-Tested results, and, for vendors who submitted their own test results, JT-NM has created a Self-Tested badge. The JT-NM was unable to hold a planned face-to-face meeting in March of 2020 where vendor-submitted tests could be validated. We decided to modify our program so that vendors who self-tested their products in accordance with the JT-NM Tested test plans could submit their results for publication in this catalog. JT-NM is responsible for the development and publication of the test plans. Vendors are solely responsible for any representations they have made in the self-tested results they have submitted; JT-NM is simply reprinting the submitted results in this catalog.

JT-NM Tested Results

For NMOS Controllers, the JT-NM Test Team was able to independently validate the vendor submitted test results using a Virtual Private Network and a lab at CBC/Radio Canada. In the case of these controller tests, the JT-NM is responsible for development and publication of the controller test plan, and also for validating vendor results included in this catalog.

JT-NM Tested March 2020 – AMWA NMOS / JT-NM TR-1001-1 Media Nodes Self-Testing Results

[illegible]

LEGEND

Green square = test passed

Grey square = test failed or not tested

White square = test not applicable



JT-NM Tested March 2020 – AMWA NMOS / JT-NM TR-1001-1 Media Nodes Self-Testing Results

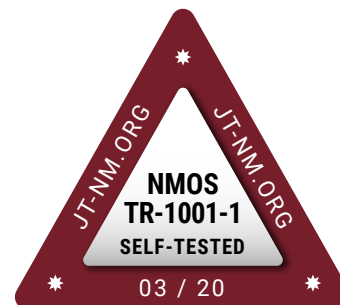
Device Under Test								2. Media Nodes & IS-04																						
Vendor Name	Model	HW Version	SW Version	IS-04 Version	IS-05 Version	IS-08 Version	Tx, Rx, Tx/Rx	2.3 Unicast discovery	2.4 Basic registration	Test_05: Registration API interactions use the correct Content-Type	Test_05_01: Registration API interactions use the correct referenced path	Test_06: Node can register a valid Node resource with the network registration service, matching its Node API self resource	Test_06_01: Node can register a valid Device resource with the network registration service, matching its Node API Device resource	Test_07: Node can register a valid Source resource with the network registration service, matching its Node API Source resource	Test_07_01: Registered Device was POSTed after a matching referenced Node	Test_08: Node can register a valid Flow resource with the network registration service, matching its Node API Flow resource	Test_08_01: Registered Flow was POSTed after a matching referenced Device	Test_09: Node can register a valid Sender resource with the network registration service, matching its Node API Sender resource	Test_09_01: Registered Sender was POSTed after a matching referenced Device	Test_10: Node can register a valid Receiver resource with the network registration service, matching its Node API Receiver resource	Test_10_01: Registered Receiver was POSTed after a matching referenced Device	Test_11: Node can register a valid Receiver resource with the network registration service, matching its Node API Receiver resource	Test_11_01: Registered Receiver was POSTed after a matching referenced Device	Test_12: Node correctly interprets a 200 code from a registry upon initial registration	2.5 Advanced registration	Test_02_01: Node does not attempt to register with an unavailable resource and registry	Test_15: Node correctly selects a Registration API based on advertised priorities	Test_16: Node correctly selects a priority between advertised Registration APIs when an advertised priority is not available	2.6 Compliance of UUIDs	
Avion	NP61600	0.2	2.7.4	v1.3	v1.1		Tx/Rx																							
Adress/Resolvers	201 IP Gateway	v1.0	v1.1	v1.1	v1.0		Tx/Rx																							
Adtel	Arro	N/A	0.19	v1.3	v1.1	v1.0	Tx/Rx																							
EEG	Alfa	2	SW 2.40.0	v1.0	v1.0		Tx/Rx																							
EVS	AT-VIA	0.2		v1.2	v1.0																									
Evertx	Scorpion	102		v1.3	v1.1		Tx/Rx																							
Evertx	070PG	1346-F		v1.3	v1.1		Tx/Rx																							
Grass Valley	MR-3901-UDC-IP	Rev A1	0.0.0.372	v1.2	v1.0		Tx/Rx																							
Magpie Communications	AMP	0	1.0.0.20	v1.1	v1.0		Tx/Rx																							
Macnica ATD	SA595510	2.0.0	2.0.0	v1.3	v1.1		Tx/Rx																							
Matrox	X-mios Q25	RevA	10.2.010	v1.3	v1.1		Multicast Tx/Rx																							
Matrox	20X LES Q25	RevA	10.2.010	v1.3	v1.1		Multicast Tx/Rx																							
Minuson	Minuson M0	v1.0	v1.4	v1.2	v1.0		Tx/Rx																							
Mobile Beach Systems	Dolphin	N/A	1.16	v1.3	v1.1		Tx/Rx																							
Medtel (Montreal)	Fusion 68 with APP ST2110-SDI Gateway	200	0	v1.2	v1.0		Tx/Rx																							
Medtel (Montreal)	Fusion 88 with APP ST2110-SDI Gateway	200	0	v1.2	v1.0		Tx/Rx																							
Medtel (Montreal)	Multi 8 with APP SDI-ST2110 Gateway	200	0	v1.2	v1.0		Tx/Rx																							
Sony	NDLX-IP900 series	2.1	0.1	v1.3	v1.1		Tx/Rx																							
Sony	HDCU-3000/5000 series	2.5	2.5	v1.3	v1.1		Tx/Rx																							
Sony	AVCC series	1.0	1.0	v1.3	v1.1		Tx/Rx																							
Sony	AVS Series	1.4	1.4	v1.3	v1.1		Tx/Rx																							
Sony	PWS-6000	2.8	2.8	v1.3	v1.1		Tx/Rx																							
Net Insight AB	NG40 (Tx)	A1	1X-users.0	v1.2	v1.0	v1.0	Tx																							
PHABRIX Ltd	PHAB-Encap	1.011.0	0	v1.2	v1.0		Tx																							
Medtel (Montreal)	Multi A10 with APP SDI-ST2110-2E Gateway	100	0.3	v1.2	v1.0		Tx																							
Indigitech	VB440 100G	1.1	6.0.1	v1.3	v1.1		Rx																							
UVS	AS-HE01	1.0	1.2	v1.2	v1.0		Rx																							
Grass Valley	Galardo IP	N/A	11.1.0	v1.2	v1.0		Rx																							
Net Insight AB	NG40 (Rx)	A1	1X-users.0	v1.2	v1.0	v1.0	Rx																							
PHABRIX Ltd	06	0	0.4	v1.3	v1.1		Rx																							
PHABRIX Ltd	PHAB-Disco	1.011.0	0	v1.2	v1.0		Rx																							
Medtel (Montreal)	Multi A10 with APP SDI-ST2110-2D Gateway	100	0.3	v1.2	v1.0		Rx																							
Telestream	PRISM 25G	MP12-25G	2.0.2	v1.2	v1.0		Rx																							
Telestream	PRISM 10G	MP1-10G	1.10.4	v1.2	v1.0		Rx																							

LEGEND

Green square = test passed

Grey square = test failed or not tested

White square = test not applicable



JT-NM Tested March 2020 – AMWA NMOS / JT-NM TR-1001-1 Media Nodes Self-Testing Results

[illegible]

LEGEND

Green square = test passed

Grey square = test failed or not tested

White square = test not applicable



JT-NM Tested March 2020 – AMWA NMOS / JT-NM TR-1001-1 Media Nodes Self-Testing Results

Device Under Test								3. Media Nodes & IS-05																											
Vendor Name	Model	HW Version	SW Version	IS-04 Version	IS-05 Version	IS-08 Version	Tx, Rx, Tx/Rx	3.1 Single sender routing	Test_1: IS-04 parameters for senders comply with constraints	Test_17: Sender patch response schema is valid	Test_18: Sender invalid patch is refused	Test_22: Receiver id on staged sender is changeable	Test_23: Sender transport parameters are changeable	Test_25: Immediate activation of a sender is possible	Test_26: Number of Hagg matches on constraints, staged and active endpoint for senders	3.4 Single Receiver routing	Test_16: Staged parameters for receivers comply with constraints	Test_18: Receiver patch response schema is valid	Test_20: Receiver invalid patch is refused	Test_21: Sender id on staged receiver is changeable	Test_24: Receiver transport parameters are changeable	Test_26: Immediate activation of a receiver is possible	Test_29: Number of Hagg matches on constraints, staged and active endpoint for receivers	3.5 Scheduled activation	Test_27: Relative activation of a sender is possible	Test_28: Relative activation of a receiver is possible	Test_29: Absolute activation of a sender is possible	Test_30: Absolute activation of a receiver is possible	3.6 Bulk Sender routing	Test_34: GET on /bulk/senders returns 405	Test_36: Bulk interface can be used to change destination port on all senders	Test_37: Bulk Receiver routing	Test_35: GET on /bulk/receiver returns 405	Test_37: Bulk interface can be used to change destination port on all receivers	
Axon	NPG1600	0.2	2.7.4	v1.3	v1.1		Tx/Rx																												
Adena/Nestera	SDR IP Gateway	v1.0	v1.1	v1.3	v1.1	v1.0	Tx/Rx																												
Dalet	Brilo	N/A	0.19	v1.3	v1.1	v1.0	Tx/Rx																												
EEG	Alta	2	0W 2.40.0	v1.2	v1.0		Tx/Rx																												
EVS	XT-VIA	0.2	06.3	v1.2	v1.0		Tx/Rx																												
Evertz	Scorpion	Scorpion	1.02	v1.3	v1.1		Tx/Rx																												
Evertz	730	730RPG	1.346-F	v1.3	v1.1		Tx/Rx																												
Grass Valley	KIP-3901-UDC-IP	Rev A1	0.0.0.372	v1.2	v1.0		Tx/Rx																												
Imagine Communications	IMP	4	0.5.0.20	v1.2	v1.0		Tx/Rx																												
Macnica ATD	EASY5330	2.0.0	2.0.0	v1.3	v1.1		Tx/Rx																												
Matrox	K-mips Q25	RevA	10.2.010	v1.3	v1.1		Multicast Tx/Rx																												
Matrox	25K LES D25	RevA	10.2.010	v1.3	v1.1		Multicast Tx/Rx																												
Neovision	Virtuoso M8	v1.0	v1.4	v1.2	v1.0		Tx/Rx																												
Pebble Beach Systems	Dolphin	N/A	1.16	v1.3	v1.1		Tx/Rx																												
Riedel (Montreal)	Fusion 6B with APP S72110-SD Gateway	200	3	v1.2	v1.0		Tx/Rx																												
Riedel (Montreal)	Fusion 1B with APP S72110-SD Gateway	200	3	v1.2	v1.0		Tx/Rx																												
Riedel (Montreal)	Musou B with APP S72110 Gateway	200	3	v1.2	v1.0		Tx/Rx																												
Sony	NDU-X-IP50V series	2.1	2.1	v1.3	v1.1		Tx/Rx																												
Sony	HDCU-3000/5000 series	2.5	2.5	v1.3	v1.1		Tx/Rx																												
Sony	HDCU series	1.0	1.0	v1.3	v1.1		Tx/Rx																												
Sony	KVS Series	1.4	1.4	v1.3	v1.1		Tx/Rx																												
Sony	PWS-4500	2.8	2.8	v1.3	v1.1		Tx/Rx																												
Net Insight AB	N640 (Tx)	A1	0.x-users.0	v1.2	v1.0		Tx																												
PHABRIX Ltd	NetTag - Encap	1.011.0	0	v1.2	v1.0		Tx																												
Riedel (Montreal)	Musou A10 with APP S72110-2E Gateway	100	3.3	v1.2	v1.0		Tx																												
BridgeTech	V8440 100G	1.1	6.0.1	v1.3	v1.1		Rx																												
US	US-NEO	1.0	1.2	v1.2	v1.0		Rx																												
Grass Valley	Kaleido-IP	N/A	11.1.0	v1.2	v1.0		Rx																												
Net Insight AB	N640 (Rx)	A1	0.x-users.0	v1.2	v1.0		Rx																												
PHABRIX Ltd	2x	0	1.4	v1.3	v1.1		Rx																												
PHABRIX Ltd	NetTag - Decap	1.011.0	0	v1.2	v1.0		Rx																												
Riedel (Montreal)	Musou A10 with APP S72110-2D Gateway	100	3.3	v1.2	v1.0		Rx																												
Telestream	PRISM 25G	MP12-25G	2.0.2	v1.2	v1.0		Rx																												
Telestream	PRISM 10G	MP1-10G	1.10.4	v1.2	v1.0		Rx																												

LEGEND

Green square = test passed

Grey square = test failed or not tested

White square = test not applicable



JT-NM Tested March 2020 – AMWA NMOS / JT-NM TR-1001-1 Media Nodes Self-Testing Results

[illegible]

LEGEND

Green square = test passed

Grey square = test failed or not tested

White square = test not applicable



JT-NM Tested March 2020 – AMWA NMOS / JT-NM TR-1001-1 Media Nodes Self-Testing Results

Device Under Test		5. Media Nodes & IS-08																			
Vendor Name	Model	HW Version	SW Version	IS-04 Version	IS-05 Version	IS-08 Version	Tx, Rx, Tx/Rx	1.1 Schema conformance	1.2 Schema conformance	1.3 Schema conformance	1.4 Schema conformance	1.5 Schema conformance	1.6 Schema conformance	1.7 Schema conformance	1.8 Schema conformance	1.9 Schema conformance	1.10 Schema conformance	1.11 Schema conformance	1.12 Schema conformance	1.13 Schema conformance	1.14 Schema conformance
Axon	NPG1600	0.2	2.7.4	v1.1	v1.1	Tx/Rx															
Adams/Needles	SD IP Gateway	v1.0	v1.1	v1.1	v1.0	Tx/Rx															
Balet	Brio	N/A	3.19	v1.3	v1.1	v1.0	Tx/Rx														
EEG	Alfa	2	SW 2.40.0	v1.2	v1.0	Tx/Rx															
ESD	ET-VIA	0.2	10.3	v1.2	v1.0	Tx/Rx															
Exortz	Scorpion	1002	v1.0	v1.1	Tx/Rx																
Eventz	070	07000	1346-F	v1.3	v1.1	Tx/Rx															
Grass Valley	KIP-3801-UDC-IP	Rev A1	3.0.0.372	v1.2	v1.0	Tx/Rx															
Imagine Communications	IMX	4	1.0.0.200	v1.0	v1.0	Tx/Rx															
Medionics ATD	EASYSS10	2.0.0	2.0.0	v1.3	v1.1	Tx/Rx															
Matrox	K-mioS G25	RevA	10.2.010	v1.3	v1.1	Multicast Tx/Rx															
Matrox	DSX-LES G25	RevA	10.2.010	v1.3	v1.1	Multicast Tx/Rx															
Novion	Virtuon-MI	v1.0	v1.4	v1.2	v1.0	Tx/Rx															
Pebble Beach Systems	Dolphin	N/A	1.10	v1.3	v1.1	Tx/Rx															
Randel (Montreal)	Russion 08 with APP ST2110-SDI Gateway	200	1	v1.2	v1.0	Tx/Rx															
Randel (Montreal)	Russion 08 with APP ST2110-SDI Gateway	200	1	v1.2	v1.0	Tx/Rx															
Randel (Montreal)	Muon 18 with APP SDI ST2110 Gateway	200	1	v1.2	v1.0	Tx/Rx															
Sony	NEX-K-IP50V series	0.1	0.1	v1.3	v1.1	Tx/Rx															
Sony	HDCU-3000/5000 series	2.5	2.5	v1.3	v1.1	Tx/Rx															
Sony	HDCU series	1.0	1.0	v1.3	v1.1	Tx/Rx															
Sony	XVS Series	0.4	0.4	v1.3	v1.1	Tx/Rx															
Sony	PWS-4500	0.8	0.8	v1.3	v1.1	Tx/Rx															
Net Insight AB	N640 (Tx)	A1	3.0-users-0	v1.2	v1.0	v1.0	Tx														
PhABRIX Ltd	SciTag - Encap	1.0311.0	2	v1.2	v1.0	Tx															
Randel (Montreal)	Muon A10 with APP SDI ST2110-2B Gateway	200	0.3	v1.2	v1.0	Tx															
BridgeTech	XB440 100G	1.1	0.0.1	v1.3	v1.1	Rx															
QVS	Q5-NEC	1.0	0.2	v1.2	v1.0	Rx															
Grass Valley	Kaleido IP	N/A	11.1.0	v1.2	v1.0	Rx															
Net Insight AB	N640 (Rx)	A1	3.0-users-0	v1.2	v1.0	v1.0	Rx														
PhABRIX Ltd	SciTag	0	0	v1.3	v1.1	Rx															
PhABRIX Ltd	SciTag - Decap	1.0311.0	2	v1.2	v1.0	Rx															
Randel (Montreal)	Muon A10 with APP SDI ST2110-2D Gateway	200	0.3	v1.2	v1.0	Rx															
Telestream	PRISM 25G	MPR2 25G	2.0.2	v1.2	v1.0	Rx															
Telestream	PRISM 10G	MPR-10G	1.10.4	v1.2	v1.0	Rx															

LEGEND

Green square = test passed

Grey square = test failed or not tested

White square = test not applicable



JT-NM Tested March 2020 – AMWA NMOS / JT-NM TR-1001-1 Media Nodes Self-Testing Results

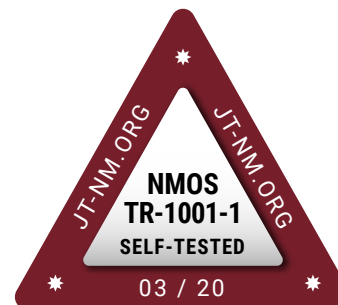
Device Under Test							5. Media Nodes & IS-08													
Vendor Name	Model	HW Version	SW Version	IS-04 Version	IS-05 Version	IS-08 Version	Tx, Rx, Tx/Rx	5.3 API behaviour	Test_07: Channels in the active resource where no input channels are routed have null sets for 'input' and 'channel_numbers'	Test_08: If the device advertises capabilities, the capabilities for each that it is not possible to create a group	Test_11: Inputs have a least one channel represented in their channel resource	Test_12: Outputs have at least one channel represented in their channel resource	Test_13: Attempting to violate routing constraints results in an HTTP 400 response	Test_14: It is not possible to re-order channels when re-assigning to a set of ports	Test_15: It is not possible to make an out-of-block route when block_size is anything other than 1	5.4 Integration with IS-04	Test_01: Actions created in a Device version number increment	Test_02: API is correctly advertised as a control endpoint	Test_03: All Output Source IDs match up to the IS-04 Node ID	Test_04: All Input Source/Receiver IDs match up to the IS-04 Node ID
Atom	NPGL600	2.7	2.7.4	v1.3	v1.3		Tx/Rx													
Advan/Nextera	SDI IP Gateway	v1.0	v1.1	v1.3	v1.3	v1.0	Tx/Rx													
Oratek	Brio	N/A	1.19	v1.3	v1.3	v1.0	Tx/Rx													
EEG	Alta	2	SW 2.40.0	v1.2	v1.0		Tx/Rx													
EVS	XT-VIA	2.2	18.3	v1.2	v1.0		Tx/Rx													
Evertz	Scorpion	Scorpion	100	v1.3	v1.3		Tx/Rx													
Evertz	670	670PFG	1346 F	v1.3	v1.3		Tx/Rx													
Grass Valley	KIP-3901-UDC-IP	Rev A1	1.0.0.372	v1.2	v1.0		Tx/Rx													
Imagine Communications	AMP	4	1.5.0.20	v1.2	v1.0		Tx/Rx													
Machine ATD	EASYSD10	2.0.0	2.0.0	v1.3	v1.3		Tx/Rx													
Matrox	K-mioS Q25	RevA	10.2.010	v1.3	v1.3		Multicast Tx/Rx													
Matrox	MXS L15 D25	RevA	10.2.010	v1.3	v1.3		Multicast Tx/Rx													
Neuvion	Orionase M1	v1.0	v1.4	v1.2	v1.0		Tx/Rx													
Pebble Beach Systems	Dolphin	N/A	1.10	v1.3	v1.3		Tx/Rx													
Riedel (Montreal)	Fusion 68 with APP 572110-SDI Gateway	100	1	v1.2	v1.0		Tx/Rx													
Riedel (Montreal)	Fusion 18 with APP 572110-SDI Gateway	100	1	v1.2	v1.0		Tx/Rx													
Riedel (Montreal)	Muson B with APP 572110-SDI Gateway	100	1	v1.2	v1.0		Tx/Rx													
Sony	NDXL-IP50V series	3.1	2.1	v1.3	v1.3		Tx/Rx													
Sony	HDCU-3000/5000 series	2.5	2.5	v1.3	v1.3		Tx/Rx													
Sony	HDCU series	1.0	1.0	v1.3	v1.3		Tx/Rx													
Sony	XVS series	1.4	1.4	v1.3	v1.3		Tx/Rx													
Sony	PWS-4500	2.8	2.8	v1.3	v1.3		Tx/Rx													
Net insight AB	N640 (Tx)	A1	1X-users.0	v1.2	v1.0		Tx													
PHABRIX Ltd	NetTag - Encap	1.011.0	7	v1.2	v1.0		Tx													
Riedel (Montreal)	Muson A10 with APP 572110-2E Gateway	100	1.3	v1.2	v1.0		Tx													
BridgeTech	VB40 100G	1.1	6.0.1	v1.3	v1.3		Rx													
KVS	K5-NIO	2.0	1.2	v1.2	v1.0		Rx													
Grass Valley	Galileo IP	N/A	11.1.0	v1.2	v1.0		Rx													
Net insight AB	N640 (Rx)	A1	1X-users.0	v1.2	v1.0		Rx													
PHABRIX Ltd	Qx	3	1.4	v1.3	v1.3		Rx													
PHABRIX Ltd	NetTag - Decap	1.011.0	7	v1.2	v1.0		Rx													
Riedel (Montreal)	Muson A10 with APP 572110-2D Gateway	100	1.3	v1.2	v1.0		Rx													
Telestream	PRISM 25G	MP12-25G	2.0.2	v1.2	v1.0		Rx													
Telestream	PRISM 10G	MP1-10G	1.10.4	v1.2	v1.0		Rx													

LEGEND

Green square = test passed

Grey square = test failed or not tested

White square = test not applicable



JT-NM Tested March 2020 – AMWA NMOS / JT-NM TR-1001-1 Registries Self-Testing Results

Device Under Test					6. IS-04 Registries																									
Vendor Name	Model	HW Version	SW Version	IS-04 Version																										
					6.1 Schema Conformance																									
					GET /x-nmos {auto_query_1}																									
					GET /x-nmos/query {auto_query_2}																									
					GET /x-nmos/query/v1.x {auto_query_3}																									
					GET /x-nmos/query/v1.x/devices {auto_query_4}																									
					GET /x-nmos/query/v1.x/devices/{deviceId} {auto_query_5}																									
					GET /x-nmos/query/v1.x/flows {auto_query_6}																									
					GET /x-nmos/query/v1.x/flows/{flowId} {auto_query_7}																									
					GET /x-nmos/query/v1.x/nodes {auto_query_8}																									
					GET /x-nmos/query/v1.x/nodes/{nodeId} {auto_query_9}																									
					GET /x-nmos/query/v1.x/receives {auto_query_10}																									
					GET /x-nmos/query/v1.x/receives/{receiverId} {auto_query_11}																									
					GET /x-nmos/query/v1.x/senders {auto_query_12}																									
					GET /x-nmos/query/v1.x/senders/{senderId} {auto_query_13}																									
					GET /x-nmos/query/v1.x/sources {auto_query_14}																									
					GET /x-nmos/query/v1.x/sources/{sourceId} {auto_query_15}																									
					GET /x-nmos/query/v1.x/subscriptions {auto_query_16}																									
					OPTIONS /x-nmos/query/v1.x/subscriptions {auto_query_17}																									
					GET /x-nmos/query/v1.x/{invalidPath} {404} {auto_query_20}																									
					GET /x-nmos {auto_registration_1}																									
					GET /x-nmos/registration {auto_registration_2}																									
					GET /x-nmos/registration/v1.x {auto_registration_3}																									
					OPTIONS /x-nmos/registration/v1.x/resource {auto_registration_4}																									
					GET /x-nmos/registration/v1.x/{invalidPath} {404} {auto_registration_5}																									

LEGEND

Green square = test passed

Grey square = test failed or not tested

White square = test not applicable



JT-NM Tested March 2020 – AMWA NMOS / JT-NM TR-1001-1 Registries Self-Testing Results

Device Under Test					6. IS-04 Registries																																
Vendor Name	Model	HW Version	SW Version	IS-04 Version																																	
Evertz	MAGNUM	N/A	20.4	v1.2	6.2 Registration	test_03: Registration API accepts and stores a valid Node resource	test_03_1: Registration API responds with correct Location header	test_03_2: Registration API accepts and stores a valid Node resource containing unicode characters	test_04: Registration API rejects an invalid Node resource with a 400 HTTP code	test_05: Registration API accepts and stores a valid Device resource	test_06: Registration API rejects an invalid Device resource with a 400 HTTP code	test_07: Registration API accepts and stores a valid Source resource	test_08: Registration API rejects an invalid Source resource with a 400 HTTP code	test_09: Registration API accepts and stores a valid Flow resource	test_10: Registration API rejects an invalid Flow resource with a 400 HTTP code	test_11: Registration API accepts and stores a valid Sender resource	test_11_1: Registration API accepts and stores a valid Sender resource with null flow_id	test_12: Registration API rejects an invalid Sender resource with a 400 HTTP code	test_13: Registration API accepts and stores a valid Receiver resource	test_14: Registration API rejects an invalid Receiver resource with a 400 HTTP code	test_15: Registration API responds with 200 HTTP code on updating a registered Node	test_16: Registration API responds with 200 HTTP code on updating a registered Device	test_17: Registration API responds with 200 HTTP code on updating a registered Source	test_18: Registration API responds with 200 HTTP code on updating a registered Flow	test_19: Registration API responds with 200 HTTP code on updating a registered Sender	test_20: Registration API responds with 200 HTTP code on updating a registered Receiver	test_26: Registration API responds with 400 HTTP code on posting a resource without parent	test_27: Registration API cleans up Nodes and their sub-resources when a heartbeat doesn't occur for the duration of a fixed timeout period	test_28: Registry removes stale child-resources of an incorrectly unregistered Node	test_30: Registration API accepts heartbeat requests for a Node held in the registry	test_32: Registration API generates 409 code when a conflicting registration exists						
Mellanox	SN2000/SN3000 Series Ethernet Switch	SN2010	Onyx 3.9.0300	v1.3																																	
NEC	NEC-IP-CONTROLLER	N/A	3.0.0-1	v1.2																																	
Sony	PWS-110NM1	N/A	2.3	v1.3																																	

LEGEND

Green square = test passed

Grey square = test failed or not tested

White square = test not applicable



JT-NM Tested March 2020 – AMWA NMOS / JT-NM TR-1001-1 Registries Self-Testing Results

Device Under Test					6. IS-04 Registries																										
Vendor Name	Model	HW Version	SW Version	IS-04 Version	6.3 Basic Querying	test_20: Query API supports websocket subscription request	test_20_1: Query API websocket subscription requests default to the current protocol	test_31: Query API sends correct websocket event messages for UNCHANGED (SYNC), ADDED, MODIFIED and REMOVED	6.4 Advanced Querying	test_21_1: Query API Implements pagination (no query or paging parameters)	test_21_1_1: Query API Implements pagination (when explicitly requested)	test_21_2: Query API Implements pagination (documentation examples)	test_21_3: Query API Implements pagination (edge case)	test_21_4: Query API Implements pagination (requests that require empty responses)	test_21_5: Query API Implements pagination (filters that select disjunctuous resources)	test_21_6: Query API Implements pagination (bad requests)	test_21_7: Query API Implements pagination (updates between paged requests)	test_21_8: Query API Implements pagination (correct encoding of URIs in Link header)	test_21_9: Query API Implements pagination (correct protocol and IP/hostname in Link header)	test_22: Query API implements downgrade queries	test_22_1: Query API subscriptions resource does not support downgrade queries	test_22_2: Query API WebSockets implement downgrade queries	test_23: Query API implements basic query parameters	test_23_1: Query API WebSockets implement basic query parameters	test_24: Query API implements RQL	test_24_1: Query API WebSockets implement RQL	test_25: Query API implements ancestry queries				
Evertz	MAGNUM	N/A	20.4	v1.2																											
Mellanox	SN2000/SN3000 Series Ethernet Switch	SN2010	Onyx 3.9.0300	v1.3																											
NEC	NEC-IP-CONTROLLER	N/A	3.0.0-1	v1.2																											
Sony	PWS-110NM1	N/A	2.3	v1.3																											

LEGEND

Green square = test passed

Grey square = test failed or not tested

White square = test not applicable



JT-NM Tested March 2020 – AMWA NMOS / JT-NM TR-1001-1 Controllers Testing Results

Device Under Test						NMOS Controllers						
						7. NMOS Controllers & Integration with the IS-04 Query Service				8. NMOS Controllers & IS-05 Connection Management		
Vendor Name	Model	HW Version	SW Version	IS-04 Version	IS-05 Version	7.1 Using unicast DNS-SD to find the IS-04 Registry	7.2 Discovering registered resources via the IS-04 Query API	7.3 Updating the list of resources dynamically from the IS-04 Registry	7.4 Discovering a Receiver's Connection Management API capability via the Query API	8.1 Instructing a Receiver to subscribe to a Sender's Flow via IS-05	8.2 Disconnecting a Receiver from a connected Flow via IS-05	8.3 Indicating the state of connections via updates received from the IS-04 Query API
Grass Valley	GV Orbit	N/A	1.0.1	v1.2	v1.0							
Imagine Communications	SDNO	N/A	2.9.9	v1.2	v1.0							
Mellanox	SN2000/SN3000 Series Ethernet Switch	SN2010	Onyx 3.9.0300	v1.3	v1.1							
Riedel Communications	NMOS Explorer	N/A	1.6.3	v1.3	v1.1							
Evertz	MAGNUM	SD	20.4	v1.2	v1.0							
Sony	PWS-110NM1	None	2.3	v1.3	v1.1							

LEGEND

Green square = test passed

Grey square = test failed or not tested

White square = test not applicable

