Pacific Northwest
NATIONAL LABORATORY

PNNL-30771

Universal Utility Data Exchange (UUDEX) Phase 4 Test Plan

Cybersecurity of Energy Delivery Systems (CEDS) Research and Development

December 2020

SR Mix
JD Welsh
S Raju
CM Schmidt
MJ Rice
SE Harpool
S Yallamraju
S Sridhar

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-OE0000190

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
email: orders@ntis.gov <https://www.ntis.gov/about>
Online ordering: http://www.ntis.gov

Universal Utility Data Exchange (UUDEX) Phase 4 Test Plan

Cybersecurity of Energy Delivery Systems (CEDS) Research and Development

December 2020

SR Mix
JD Welsh
S Raju
CM Schmidt
MJ Rice
SE Harpool
S Yallamraju
S Sridhar

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Revision History

Revision	Date	Deliverable (Reason for Change)	Release #
#	12/10/2020	Initial release	Initial release

Summary

This report Provides a test plan for the Phase 4 UUDEX demonstration.

Acronyms and Abbreviations

ACL	access control list
CA	certificate authority
CN	common name (a field in an X.509 digital certificate)
DOE	U. S. Department of Energy
IETF	Internet Engineering Task Force
ISO	International Standardization Organization
JSON	JavaScript Object Notation, as defined by IETF RFC 7159
OE-417	DOE Electric Emergency Incident and Disturbance Report
PDF	Portable Document Format, as specified in ISO 32000
PSIR	Physical Security Incident Report
RFC	Request for Comment – Used in reference to published IETF standards, which are called RFCs
STIX	Structured Threat Information eXpression, as defined by the OASIS Cyber Threat Intelligence (CTI) Technical Committee
UUDEX	Universal Utility Data Exchange
XML	eXtensible Markup Language, as defined by the W3C

Contents

Figures

No table of figures entries found.

Tables

1.0 Introduction

This report provides the test plan for the UUDEX Phase 4 demonstration

The tests are generally designed to be performed in the order specified allowing previous test results (e.g., created subjects, published data) to be used in subsequent tests. However, the tests may be performed out of order by ensuring pre-requisite steps noted in the introduction paragraph for a specific test are performed. If no pre-requisites are noted, there are none.

Although not required, the tests may be automated to facilitate repeatability and ease of testing, as well as automatic capture and verification of results.

Note: this is a working document. As the UUDEX prototype code continues to be developed, additional tests (including those identified in this version of the Test Plan as “To Be Determined”) may be identified, and some tests may be found to be redundant or unnecessary.

2.0 Test Setup

The following are required to complete the tests in this document:

1. A UUDEX Server instance to contain all subject definitions and data queues associated with the subjects.
2. Diagnostic tools used to verify the test results that allow the review of created subjects and their properties, specifically including the access control list (ACL) and constraints. This may include use of the RabbitMQ console for viewing metadata associated with the content of the individual subject queues, or programmer/administrator access directly to the UUDEX metadata structures.
3. UUDEX Client application software (customized as necessary to perform tests) that is capable of both publish and subscribe functions.
4. A Certificate Authority (CA) that issues the X.509 certificates and that is accessible by the UUDEX instance, and the publisher client and subscriber client applications. Since the tests do not test the specific functionality of the certificate authority, certificates may be self-signed for the test. If a Certificate Authority is used, a separate suite of tests can be developed to test CA functionality (e.g., certificate signing, revocation, expiration, chain of trust, etc.).
5. X.509 formatted digital certificates with the following Common Names (CN) – to be used for ACL definitions:
    a. Alice (valid endpoint participant)
    b. Bob (valid endpoint participant)
    c. Carol (valid endpoint participant)
    d. David (valid endpoint participant)
    e. Eve (valid endpoint participant)
    f. Frank (valid endpoint participant)
    g. Grace (official [government] endpoint participant)
    h. Jack (untrusted endpoint participant)
    i. Lyle (untrusted endpoint participant)
    j. Mallory (malicious endpoint participant)
    k. Oscar (untrusted endpoint participant)
    l. Tester (owner of test subjects created during execution of the test plan)
    m. Tester1 (owner of test subjects created during execution of the test plan)
    n. Walter (UUDEX administrator)
    o. E-ISAC
    p. NERC
6. Create the Default Subject Policy containing the constraints and behaviors as specified in Table 2.1:

Table 2.1: Default Subject Policy Constraints

Constraint Name	Constraint Value
broadestAllowedPublisherAccess	allowAll
broadestAllowedSubscriberAccess	allowAll
broadestAllowedManagerAccess	allowAll
broadestAllowedDiscoveryAccess	allowAll
fullQueueBehavior	PURGE_OLD
deliveryBehavior	DELETE_ON_DELIVERY

All the tests require verification to ensure that the test has completed successfully. The specific steps for verification are different depending on what is being tested. Example verification procedures include:

- For Subject Creation – this is the most difficult verification, since the subject definitions and permissions will be stored internally to the selected message broker system. Most message brokers provide a view into the subject delivery queues to determine if messages are in the queues, but the actual permission and constraint structure of the subject queue may not be available, especially if is controlled by UUDEX processing outside of the message broker (for example, the “PURGE_OLD” constraint, or specific ACL permissions). To verify these structures, administrative access to the UUDEX configuration datastore will be necessary to verify the proper constraints and permissions. Each UUDEX product will need to develop a mechanism to view the contents of the configuration datastore, which may have diagnostic uses beyond the execution of this test plan.
- For Subject Publish – the most straightforward method of verifying that a UUDEX message has been published is to view it in the diagnostic console provided by the selected message broker. The presence of the published message should be viewable before it is consumed by subscribers. For example, the RabbitMQ administration interface provides this capability.
- For Subject Subscription – this is a straightforward verification. The successful act of subscribing means that the UUDEX message has been received by the subscribing application, and can be displayed by the application, or an auxiliary application, such as Adobe Acrobat for ensuring that PDF files are successfully transmitted through UUDEX.

For purposes of these tests, when subjects are created without any constraints, it is assumed that the constraints in Table 2.1 are in place in the Default Subject Policy, since the test is validating other behavior.

in some cases, tests may be combined to expedite testing. For example publish tests in Section 4.0 may be combined with subscribe tests in section 5.0 or Section 6.0.

3.0 Subject Administration

The first set of tests in Section 3.1 involves creating a number of UUDEX Subjects with varying attributes and security settings.

Subject names use the following naming conventions:

- Participant ID: “Tester”
- UUDEX Data Element type ID: “TEST”
- Group key: Test number

Administrative access to the UUDEX internal data structures may be required to verify the creation and configuration of subjects created in these tests.

Note: the subjects created by the tests in Section 3.1 will be used by the tests in Section 3.2 and Section 3.3. Each of the tests in Section 3.1 start by verifying that the subjects do not exist allowing the tests to initially create them. The tests in Section 3.2 verify that the constraints for existing subjects can be modified, while the tests in Section 3.3 verify the discovery process for existing subjects.

Note: these tests only verify the subject creation, modification, or discovery functionality, and do not involve subscription or publication. Tests to verify publishing are contained in Section 4.0; tests to verify subscriptions are contained in Section 5.0 and Section 6.0.

3.1 Subject Creation

These tests verify the behavior of UUDEX Subject Creation processes with different constraints.

3.1.1 Subject Policies

These tests verify the modification of the Default Subject Policy and the creation of a Participant-Data Type Subject Policy, a Participant Subject Policy, and a Data Type Subject Policy.

3.1.1.1 Modify Default Subject Policy

This test verifies that the Default Subject Policy can be modified.

1. Verify the Default Subject Policy specifies the following:
   a. broadestAllowedPublisherAccess: allowAll
   b. broadestAllowedSubscriberAccess: allowAll
   c. broadestAllowedManagerAccess: allowAll
   d. broadestAllowedDiscoveryAccess: allowAll
   e. fullQueueBehavior: PURGE_OLD

f. deliveryBehavior: DELETE_ON_DELIVERY

2. Modify the Default Subject Policy with the following:
   a. broadestAllowedPublisherAccess: allowNone
   b. broadestAllowedSubscriberAccess: allowNone
   c. broadestAllowedManagerAccess: allowNone
   d. broadestAllowedDiscoveryAccess: allowNone
   e. fullQueueBehavior: BLOCK_NEW
   f. deliveryBehavior: RETAIN_ON_DELIVERY
3. Verify the Default Subject Policy has been modified as indicated.

4. Modify the Default Subject Policy with the following:
   a. broadestAllowedPublisherAccess: allowOnly = {Alice, Bob}
   b. broadestAllowedSubscriberAccess: allowOnly = {Carol, David}
   c. broadestAllowedManagerAccess: allowOnly = {Eve, Frank}
   d. broadestAllowedDiscoveryAccess: allowOnly = {Carol, Frank}
5. Verify the Default Subject Policy has been modified as indicated.

6. Modify the Default Subject Policy with the following:
   a. broadestAllowedPublisherAccess: allowExcept = {Alice, Bob}
   b. broadestAllowedSubscriberAccess: allowExcept = {Carol, David}
   c. broadestAllowedManagerAccess: allowExcept = {Eve, Frank}
   d. broadestAllowedDiscoveryAccess: allowExcept = {Carol, Frank}
7. Verify the Default Subject Policy has been modified as indicated.

3.1.1.2 Participant-Data Type Subject Policy

These tests verify the creation and modification of a Participant-Data Type Subject Policy.

Note: the Participant-Data Type Subject Policy should not exist initially, so the verification in Step 1 should not return a valid result.

1. Repeat the tests in Section 3.1.1.1 creating the Participant-Data Type Subject Policy instead of the Default Subject Policy, using the following Participant and Data Type:
   a. Participant: “Tester”
   b. Data Type: “TEST”

3.1.1.3 Participant Subject Policy

These tests verify the creation and modification of a Participant Subject Policy.

Note: the Participant Subject Policy should not exist initially, so the verification in Step 1 should not return a valid result.

1. Repeat the tests in Section 3.1.1.1 creating the Participant Subject Policy instead of the Default Subject Policy, using the following Participant:
   a. Participant: “Tester”

3.1.1.4 Data Type Subject Policy

These tests verify the creation and modification of a Data Type Subject Policy.

Note: the Data Type Subject Policy should not exist initially, so the verification in Step 1 should not return a valid result.

1. Repeat the tests in Section 3.1.1.1 creating the Data Type Subject Policy instead of the Default Subject Policy, using the following Data Type:
   a. Data Type: “TEST”

3.1.1.5 Remove Testing Participant and Data Type Policies and Reset the Default Subject Policy

This test step removes the Participant and Data Type Policy records leaving the UUDEX instance in a default state.

1. Remove the Participant-Data Type Subject Policy, Participant Subject Policy, and Data Type Subject Policy records created for the test in Section 3.1.1.2 through Section 3.1.1.4.
2. Verify that all Participant-Data Type Subject Policy, Participant Subject Policy, and Data Type Subject Policy records created for the test in Section 3.1.1.2 through Section 3.1.1.4 have been deleted.
1. Modify the Default Subject Policy to be as specified in Table 2.1.

3.1.2 Create Subject with Default Constraints

These tests verify subject creation with no performance or access constraints.

Before starting this set of tests, verify that all subjects with Participant “Tester”, Data Type “TEST” and Group Keys starting with “3.1.2” are not present. Remove them if they exist.

3.1.2.1 Create Subject with only Default Subject Policy Constraints

This test verifies that a subject created with no additional constraints is created using the Default Subject Policy constraints.

1. Verify the configuration of the Default Subject Policy is as specified in Table 2.1 (allow creation of all subjects and imposes no constraints or access limits).
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.
3. Create UUDEX Subject “Tester/TEST/T3.1.2.1” with no constraints for size, performance, or access specified.
4. Verify subject was created.
5. Verify the subject constraints match the Default Subject Policy constraints.

3.1.2.2 Create Subject with Participant Subject Policy Constraints

This test verifies proper creation of a subject when a Participant Subject Policy is applicable.

1. Verify the configuration of the Default Subject Policy is as specified in Table 2.1 (allow creation of all subjects and imposes no constraints or access limits).
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.
3. Create UUDEX Subject “Tester/TEST/T3.1.2.2-1” with no constraints for size, performance, or access specified.
4. Verify subject was created.
5. Verify the subject constraints match the Default Subject Policy constraints.

Verify the Participant Subject Policy processing.

6. Create a Participant Subject Policy with the following:
   a. Participant “Tester”
   b. maxQueueSizeKB = 10kB
7. Create UUDEX Subject “Tester/TEST/T3.1.2.2-2” with no constraints for size, performance, or access specified.

8. Verify subject was created.
9. Verify the subject maxQueueSizeKB constraint is 10kB.
10. Verify the other subject constraints match the Default Subject Policy constraints.

Verify the Participant Subject Policy processing only affects the specified participant.

11. Create UUDEX Subject “Tester1/TEST1/T3.1.2.3-3” with no constraints for size, performance, or access specified.
12. Verify subject was created.
13. Verify the subject maxQueueSizeKB constraint is not specified (matches Default Subject Policy constraints).
14. Verify the other subject constraints match the Default Subject Policy constraints.

15. Create UUDEX Subject “Tester/TEST/T3.1.2.2-4” with the following constraint specified:
    a. maxMessageCount = 2
16. Verify subject was created.
17. Verify the subject maxQueueSizeKB constraint is 10kB.
18. Verify the subject maxMessageCount constraint is 2.
19. Verify the other subject constraints match the Default Subject Policy constraints.

20. Remove the Participant Subject Policy created in step 6.

3.1.2.3 Create Subject with Data Type Subject Policy Constraints

This test verifies proper creation of a subject when a Data Type Subject Policy is applicable.

1. Verify the configuration of the Default Subject Policy is as specified in Table 2.1 (allow creation of all subjects and imposes no constraints or access limits).
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.
3. Create UUDEX Subject “Tester/TEST/T3.1.2.3-1” with no constraints for size, performance, or access specified.
4. Verify subject was created.
5. Verify the subject constraints match the Default Subject Policy constraints.

6. Create a Data Type Policy with the following:

   a. Data Type “TEST”
   b. maxQueueSizeKB = 10kB
7. Create UUDEX Subject “Tester/TEST/T3.1.2.3-2” with no constraints for size, performance, or access specified.
8. Verify subject was created.
9. Verify the subject maxQueueSizeKB constraint is 10kB.
10. Verify the other subject constraints match the Default Subject Policy constraints.

Verify the Data Type Policy processing only affects the specified dataType.

11. Create UUDEX Subject “Tester/TEST1/T3.1.2.3-3” with no constraints for size, performance, or access specified.
12. Verify subject was created.
13. Verify the subject maxQueueSizeKB constraint is not specified (matches Default Subject Policy constraints).
14. Verify the other subject constraints match the Default Subject Policy constraints.

15. Create UUDEX Subject “Tester/TEST/T3.1.2.3-4” with the following constraint specified:
   a. maxMessageCount = 2
16. Verify subject was created.
17. Verify the subject maxQueueSizeKB constraint is 10kB.
18. Verify the subject maxMessageCount constraint is 2.
19. Verify the other subject constraints match the Default Subject Policy constraints.

20. Remove the Data Type Policy created in step 6.

3.1.2.4 Create Subject with Participant-Data Type Subject Policy Constraints

This test verifies proper creation of a subject when a Participant-Data Type Subject Policy is applicable.

1. Verify the configuration of the Default Subject Policy is as specified in Table 2.1 (allow creation of all subjects and imposes no constraints or access limits).
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.
3. Create UUDEX Subject “Tester/TEST/T3.1.2.4-1” with no constraints for size, performance, or access specified.

4. Verify subject was created.
5. Verify the subject constraints match the Default Subject Policy constraints.

6. Create a Participant-Data Type Policy with the following:
   a. Participant “Tester”
   b. Data Type “TEST”
   c. maxQueueSizeKB = 10kB
7. Create UUDEX Subject “Tester/TEST/T3.1.2.4-2” with no constraints for size, performance, or access specified.
8. Verify subject was created.
9. Verify the subject maxQueueSizeKB constraint is 10kB.
10. Verify the other subject constraints match the Default Subject Policy constraints.

Verify the Participant-Data Type Policy processing only affects the specified dataType.

11. Create UUDEX Subject “Tester/TEST1/T3.1.2.4-3” with no constraints for size, performance, or access specified.
12. Verify subject was created.
13. Verify the subject maxQueueSizeKB constraint is not specified (matches Default Subject Policy constraints).
14. Verify the other subject constraints match the Default Subject Policy constraints.

Verify the Participant-Data Type Policy processing only affects the specified Participant.

15. Create UUDEX Subject “Tester1/TEST/T3.1.2.4-4” with no constraints for size, performance, or access specified.
16. Verify subject was created.
17. Verify the subject maxQueueSizeKB constraint is not specified (matches Default Subject Policy constraints).
18. Verify the other subject constraints match the Default Subject Policy constraints.

Verify the Participant-Data Type Policy processing only affects the specified Participant / Data Type combination.

19. Create UUDEX Subject “Tester1/TEST1/T3.1.2.4-5” with no constraints for size, performance, or access specified.

20. Verify subject was created.
21. Verify the subject maxQueueSizeKB constraint is not specified (matches Default Subject Policy constraints).
22. Verify the other subject constraints match the Default Subject Policy constraints.

23. Create UUDEX Subject “Tester/TEST/T3.1.2.4-6” with the following constraint specified:
    a. maxMessageCount = 2
24. Verify subject was created.
25. Verify the subject maxQueueSizeKB constraint is 10kB.
26. Verify the subject maxMessageCount constraint is 2.
27. Verify the other subject constraints match the Default Subject Policy constraints.

28. Remove the Data Type Policy created in step 6.

3.1.3 Create Subject with Response Acknowledgement

This test verifies subject creation with acknowledgement.

Before starting this test, verify that all subjects with Participant “Tester”, Data Type “TEST” and Group Keys starting with “3.1.3” are not present. Remove them if they exist.

1. Verify the configuration of the Default Subject Policy is as specified in Table 2.1 (allow creation of all subjects and imposes no constraints or access limits).
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.
3. Create UUDEX Subject “Tester/TEST/T3.1.3-Response” with no constraints for size, performance, or access specified.
4. Verify subject was created.

5. Create UUDEX Subject “Tester/TEST/T3.1.3-1” with and the following metadata in the publish header:
    a. ackRequired = true
    b. replyAddress = “Tester/TEST/ T3.1.3-Response”
6. Verify subject was created.
7. Subscribe to UUDEX Subject “Tester/TEST/T3.1.3-Response”.
8. Retrieve response message.
9. Verify ackReply field is true.

10. Create UUDEX Subject “Tester/TEST/T3.1.3-2” with and the following metadata in the publish header:
    a. ackRequired = false
    b. replyAddress = “Tester/TEST/ T3.1.3-Response”
11. Verify subject was created.
12. Subscribe to UUDEX Subject “Tester/TEST/T3.1.3-Response”.
13. Retrieve response message.
14. Verify ackReply field is false.

3.1.4 Create Subject with Default Message Size Performance Restrictions

The following tests verify subject creation with message size constraints.

Before starting this test, verify that all subjects with Participant “Tester”, Data Type “TEST” and Group Keys starting with “3.1.4” are not present. Remove them if they exist.

1. Modify the Default Subject Policy to contain a performance constraint maxQueueSizeKB of 100kB, a fullQueueBehavior constraint of “PURGE_OLD” and no other constraints.
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.
3. Create UUDEX Subject “Tester/TEST/T3.1.4-1” with no constraints for size, performance, or access specified.
4. Verify subject was created.
5. Verify performance constraints match system default of 100kB.
6. Verify fullQueueBehavior performance matches Default Subject Policy (PURGE_OLD).
7. Verify lack of access constraints.

8. Modify the Default Subject Policy to contain only a performance constraint maxQueueSizeKB of 100kB the fullQueueBehavior parameter set to “BLOCK_NEW”, and no other constraints or constraints.
9. Create UUDEX Subject “Tester/TEST/T3.1.4-2” with no constraints for size, performance, or access specified.
10. Verify subject was created.
11. Verify performance constraints match system default of 100kB.
12. Verify fullQueueBehavior performance restriction of BLOCK_NEW.
13. Verify lack of access constraints.

14. Modify the Default Subject Policy to contain only a performance constraint maxQueueSizeKB of 100kB the fullQueueBehavior parameter set to “PURGE_OLD”, and no other constraints or constraints.
15. Create UUDEX Subject “Tester/TEST/T3.1.4-3” with no constraints for size, performance, or access specified.
16. Verify subject was created.
17. Verify performance constraints match system default of 100kB.
18. Verify fullQueueBehavior performance restriction of PURGE_OLD.
19. Verify lack of access constraints.

20. Modify the Default Subject Policy to be as specified in Table 2.1.

3.1.5 Create Subject with Default Queue Size Performance Restrictions

The following tests verify subject creation with queue size constraints.

Before starting this test, verify that all subjects with Participant “Tester”, Data Type “TEST” and Group Keys starting with “3.1.5” are not present. Remove them if they exist.

1. Modify the Default Subject Policy to contain only a performance constraint maxMessageCount of 2, fullQueueBehavior of “PURGE_OLD”, and no other constraints or constraints.
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.
3. Create UUDEX Subject “Tester/TEST/T3.1.5-1” with no constraints for size, performance, or access specified.
4. Verify subject was created.
5. Verify performance constraints match system default of 2 messages.
6. Verify fullQueueBehavior performance matches the Default Subject Policy (PURGE_OLD).
7. Verify lack of access constraints.

8. Modify the Default Subject Policy to contain only a performance constraint maxMessageCount of 2, the fullQueueBehavior parameter set to “BLOCK_NEW”, and no other constraints or constraints.
9. Create UUDEX Subject “Tester/TEST/T3.1.5-2” with no constraints for size, performance, or access specified.
10. Verify subject was created.
11. Verify performance constraints match system default of 2 messages.
12. Verify fullQueueBehavior performance restriction of BLOCK_NEW.

13. Verify lack of access constraints.

14. Modify the Default Subject Policy to contain only a performance constraint maxMessageCount of 2, the fullQueueBehavior parameter set to “PURGE_OLD”, and no other constraints or constraints.
15. Create UUDEX Subject “Tester/TEST/T3.1.5-3” with no constraints for size, performance, or access specified.
16. Verify subject was created.
17. Verify performance constraints match system default of 2 messages.
18. Verify fullQueueBehavior performance restriction of PURGE_OLD.
19. Verify lack of access constraints.

20. Modify the Default Subject Policy to be as specified in Table 2.1.

3.1.6 Create Subject with Default Access Constraints and Verify Administrator Approvals

The following tests verify subject creation with default access constraints.

Note: the specific mechanics of administrator actions are implementation dependent, and are not specified in this test plan. This may be a manual process.

Before starting this test, verify that all subjects with Participant “Tester”, Data Type “TEST” and Group Keys starting with “3.1.6” are not present. Remove them if they exist.

1. Modify the Default Subject Policy to contain only access constraint of “ALLOW” for all subjects.
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.
3. Create UUDEX Subject “Tester/TEST/T3.1.6-1” with no constraints for size, performance, or access specified.
4. Verify subject was created.

5. Modify the Default Subject Policy to contain only access constraint of “REVIEW” for all subjects.
6. Create UUDEX Subject “Tester/TEST/T3.1.6-2” with no constraints for size, performance, or access specified.
7. Verify a request was sent to the UUDEX administrator to approve the subject creation request.
8. UUDEX Administrator approves creation request.

9. Verify subject was created.

10. Create UUDEX Subject “Tester/TEST/T3.1.6-3” with no constraints for size, performance, or access specified.
11. Verify a request was sent to the UUDEX administrator to approve the subject creation request.
12. UUDEX Administrator denies creation request.
13. Verify subject was not created.

14. Modify the Default Subject Policy to be as specified in Table 2.1..

3.1.7 Create Subject with Performance Constraints (no default constraints)

This set of tests is performed with the Default Subject Policy as specified in Table 2.1 (allow creation of all subjects and imposes no constraints or access limits).

Before starting this test, verify that all subjects with Participant “Tester”, Data Type “TEST” and Group Keys starting with “3.1.7” are not present. Remove them if they exist.

3.1.7.1 Size Limit

This test verifies subject creation with maximum queue message size constraints specified in the subject creation request.

1. Modify the Default Subject Policy to contain no access or performance constraints, and a fullQueueBehavior to “PURGE_OLD”.
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.
3. Create UUDEX Subject “Tester/TEST/T3.1.7.1” with a limit on the total size of stored messages of 100kB.
4. Verify the subject was created.
5. Verify constraint restriction of 100kB.
6. Verify fullQueueBehavior performance matches the Default Subject Policy (PURGE_OLD) (implicit PURGE_OLD).
7. Verify no other constraint constraints.
8. Verify lack of access constraints.

3.1.7.2 Queue Length Limit

This test verifies subject creation with maximum number of queued messages constraints specified in the subject creation request.

1. Create UUDEX Subject “Tester/TEST/T3.1.7.2” with a limit on the total number of stored messages of 2.
2. Verify the subject was created.
3. Verify constraint restriction of 2 messages.
4. Verify fullQueueBehavior performance matches the Default Subject Policy (PURGE_OLD) (implicit PURGE_OLD).
5. Verify no other constraint constraints.
6. Verify lack of access constraints.

3.1.7.3 Maximum Priority

This test verifies subject creation with queue maximum priority constraints specified in the subject creation request.

7. Create UUDEX Subject “Tester/TEST/T3.1.7.3” with specified maximum priority of 3.
8. Verify the subject was created.
9. Verify constraint restriction with a maximum priority of 3.
10. Verify fullQueueBehavior performance matches the Default Subject Policy (PURGE_OLD) (implicit PURGE_OLD).
11. Verify no other constraint constraints.
12. Verify lack of access constraints.

3.1.7.4 Full Queue Block Behavior

This test verifies subject creation with block on full queue constraints specified in the subject creation request.

1. Create UUDEX Subject “Tester/TEST/T3.1.7.4” specifying that messages should be blocked if the queue length or queue size is exceeded and with a limit on the total size of stored messages of 100kB.
2. Verify the subject was created.
3. Verify fullQueueBehavior performance of BLOCK_NEW.
4. Verify constraint restriction of 100kB.

5. Verify no other constraint constraints.
6. Verify lack of access constraints.

3.1.7.5 Full Queue Delete Behavior

This test verifies subject creation with delete on full queue constraints specified in the subject creation request.

1. Create UUDEX Subject “Tester/TEST/T3.1.7.5” specifying that old messages should be deleted if the queue length or queue size is exceeded and with a limit on the total size of stored messages of 100kB.
2. Verify the subject was created.
3. Verify fullQueueBehavior performance of PURGE_OLD (explicit PURGE_OLD).
4. Verify constraint restriction of 100kB.
5. Verify no other constraint constraints.
6. Verify lack of access constraints.

3.1.8 Create Subject with Performance Constraints (with default constraints)

This set of tests is performed with the Default Subject Policy containing constraints. These tests verify that the resulting subjects are created with the more restrictive of either the Default Subject Policy or the specific request.

Before starting this test, verify that all subjects with Participant “Tester”, Data Type “TEST” and Group Keys starting with “3.1.8” are not present. Remove them if they exist.

3.1.8.1 Modify Default Subject Policy

This test modifies the Default Subject Policy to contain specific constraints that will be either enforced or overridden in the following tests.

7. Verify the Default Subject Policy is configured as described in Table 2.1.

1. Modify the Default Subject Policy with the following constraints:
   a. Maximum size of 100 kB
   b. Maximum queue length of 3
   c. Maximum priority of 3
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.

Note: there are two tests and results for each test in this section.

3.1.8.2 Size Limit

This test verifies the ability to constrain the total message size of the message queue with the lesser of the Default Subject Policy or the specific subject creation request constraints.

1. Verify subject policies as described in Section 3.1.8.1.
2. Create UUDEX Subject “Tester/TEST/T3.1.8.2-1” with a limit on the total size of stored messages of 200 kB.
3. Verify the subject was created.
4. Verify constraint restriction of 100kB.
5. Verify lack of access constraints.

6. Create UUDEX Subject “Tester/TEST/T3.1.8.2-2” with a limit on the total size of stored messages of 50 kB.
7. Verify the subject was created.
8. Verify constraint restriction of 50kB.
9. Verify lack of access constraints.

3.1.8.3 Queue Length Limit

This test verifies the ability to constrain the number of messages in the queue with the lesser of the Default Subject Policy or the specific subject creation request constraints.

1. Verify subject policies as described in Section 3.1.8.1.
2. Create UUDEX Subject “Tester/TEST/T3.1.8.3-1” with a limit on the total number of stored messages of 5.
3. Verify the subject was created.
4. Verify constraint restriction of 3 messages.
5. Verify lack of access constraints.

6. Create UUDEX Subject “Tester/TEST/T3.1.8.3-2” with a limit on the total number of stored messages of 2.
7. Verify the subject was created.
8. Verify constraint restriction of 2 messages.
9. Verify lack of access constraints.

3.1.8.4 Maximum Priority

This test verifies the ability to constrain the maximum priority of the message queue with the lesser of the Default Subject Policy or the specific subject creation request constraints.

1. Verify subject policies as described in Section 3.1.8.1.
2. Create UUDEX Subject “Tester/TEST/T3.1.8.4-1” with maximum priority of 2.
3. Verify the subject was created.
4. Verify constraint restriction priority of 3.
5. Verify lack of access constraints.

6. Create UUDEX Subject “Tester/TEST/T3.1.8.4-2” with maximum priority of 4.
7. Verify the subject was created.
8. Verify constraint restriction priority of 4.
9. Verify lack of access constraints.

3.1.9 Create Subject with Access Constraints (no default constraints)

Access constraints, in the form ACL specifications, are used to restrict which participants can create, view, publish to, or subscribe to subjects.

This set of tests validates the access constraint processing when there are no constraints specified in the Default Subject Policy.

Before starting this test, verify that all subjects with Participant “Tester”, Data Type “TEST” and Group Keys starting with “3.1.10” are not present. Remove them if they exist.

Note: there are multiple tests and results for each test in this section.

3.1.9.1 Broadest Publish Access

This test verifies the Broadest Publish Access constraint with no Default Subject Policy constraints and additional constraints specified in the subject creation request.

1. Verify the configuration of the Default Subject Policy is as specified in Table 2.1 (allow creation of all subjects and imposes no constraints or access limits).
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.
3. Create UUDEX Subject “Tester/TEST/T3.1.9.1-1” with broadestAllowedPublisherAccess set to allowAll.
4. Verify the subject was created.

5. Verify lack of constraint restriction.
6. Verify access constraints set to allow publish access to all.

7. Create UUDEX Subject “Tester/TEST/T3.1.9.1-2” with broadestAllowedPublisherAccess set to allowNone.
8. Verify the subject was created.
9. Verify lack of constraint restriction.
10. Verify access constraints set to allow publish access to no one.

11. Create UUDEX Subject “Tester/TEST/T3.1.9.1-3” with broadestAllowedPublisherAccess set to allowExcept = {Jack}.
12. Verify the subject was created.
13. Verify lack of constraint restriction.
14. Verify access constraints set to allow publish access to all except Jack.

15. Create UUDEX Subject “Tester/TEST/T3.1.9.1-4” with broadestAllowedPublisherAccess set to allowExcept = {Lyle, Mallory}.
16. Verify the subject was created.
17. Verify lack of constraint restriction.
18. Verify access constraints set to allow publish access to all except Lyle and Mallory.

19. Create UUDEX Subject “Tester/TEST/T3.1.10.1-5” with broadestAllowedPublisherAccess set to allowOnly = {Alice}.
20. Verify the subject was created.
21. Verify lack of constraint restriction.
22. Verify access constraints set to allow publish access to only Alice.

23. Create UUDEX Subject “Tester/TEST/T3.1.10.1-6” with broadestAllowedPublisherAccess set to allowOnly = {Bob, Carol}.
24. Verify the subject was created.
25. Verify lack of constraint restriction.
26. Verify access constraints set to allow publish access to only Bob and Carol.

3.1.9.2 Broadest Subscribe Access

This test verifies the Broadest Subscribe Access constraint with no Default Subject Policy constraints and additional constraints specified in the subject creation request.

1. Verify the configuration of the Default Subject Policy is as specified in Table 2.1 (allow creation of all subjects and imposes no constraints or access limits).
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.
3. Create UUDEX Subject “Tester/TEST/T3.1.9.2-1” with broadestAllowedSubscriberAccess set to allowAll.
4. Verify the subject was created.
5. Verify lack of constraint restriction.
6. Verify access constraints set to allow subscribe access to all.

7. Create UUDEX Subject “Tester/TEST/T3.1.10.2-2” with broadestAllowedSubscriberAccess set to allowNone.
8. Verify the subject was created.
9. Verify lack of constraint restriction.
10. Verify access constraints set to allow subscribe access to no one.

11. Create UUDEX Subject “Tester/TEST/T3.1.10.2-3” with broadestAllowedSubscriberAccess set to allowExcept = {Jack}.
12. Verify the subject was created.
13. Verify lack of constraint restriction.
14. Verify access constraints set to allow subscribe access to all except Jack.

15. Create UUDEX Subject “Tester/TEST/T3.1.10.2-4” with broadestAllowedSubscriberAccess set to allowExcept = {Lyle, Mallory}.
16. Verify the subject was created.
17. Verify lack of constraint restriction.
18. Verify access constraints set to allow a subscribe access to all except Lyle and Mallory.

19. Create UUDEX Subject “Tester/TEST/T3.1.10.2-5” with broadestAllowedSubscriberAccess set to allowOnly = {Alice}.
20. Verify the subject was created.
21. Verify lack of constraint restriction.
22. Verify access constraints set to subscribe access to allow only Alice.

23. Create UUDEX Subject “Tester/TEST/T3.1.10.2-6” with broadestAllowedSubscriberAccess set to allowOnly = {Bob, Carol}.

24. Verify the subject was created.
25. Verify lack of constraint restriction.
26. Verify access constraints set to subscribe access to allow only Bob and Carol.

3.1.9.3 Broadest Management Access

This test verifies the Broadest Management Access constraint with no Default Subject Policy constraints and additional constraints specified in the subject creation request.

1. Verify the configuration of the Default Subject Policy is as specified in Table 2.1 (allow creation of all subjects and imposes no constraints or access limits).
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.
3. Create UUDEX Subject “Tester/TEST/T3.1.10.3-1” with broadestAllowedManagerAccess set to allowAll.
4. Verify the subject was created.
5. Verify lack of constraint restriction.
6. Verify access constraints to allow management access to all.

7. Create UUDEX Subject “Tester/TEST/T3.1.10.3-2” with broadestAllowedManagerAccess set to allowNone.
8. Verify the subject was created.
9. Verify lack of constraint restriction.
10. Verify access constraints to allow management access to no one.

11. Create UUDEX Subject “Tester/TEST/T3.1.10.3-3” with broadestAllowedManagerAccess set to allowExcept = {Jack}.
12. Verify the subject was created.
13. Verify lack of constraint restriction.
14. Verify access constraints to allow management access to all except Jack.

15. Create UUDEX Subject “Tester/TEST/T3.1.10.3-4” with broadestAllowedManagerAccess set to allowExcept = {Lyle, Mallory}.
16. Verify the subject was created.
17. Verify lack of constraint restriction.
18. Verify access constraints to allow management access to all except Lyle and Mallory.

19. Create UUDEX Subject “Tester/TEST/T3.1.10.3-5” with broadestAllowedManagerAccess set to allowOnly = {Alice}.
20. Verify the subject was created.
21. Verify lack of constraint restriction.
22. Verify access constraints to allow management access to only Alice.

23. Create UUDEX Subject “Tester/TEST/T3.1.10.3-6” with broadestAllowedManagerAccess set to allowOnly = {Bob, Carol}.
24. Verify the subject was created.
25. Verify lack of constraint restriction.
26. Verify access constraints to allow management access to only Bob and Carol.

3.1.9.4 Broadest Discovery Access

This test verifies the Broadest Discovery Access constraint with no Default Subject Policy constraints and additional constraints specified in the subject creation request.

1. Verify the configuration of the Default Subject Policy is as specified in Table 2.1 (allow creation of all subjects and imposes no constraints or access limits).
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.
3. Create UUDEX Subject “Tester/TEST/T3.1.9.4-1” with broadestAllowedDiscoveryAccess set to allowAll.
4. Verify the subject was created.
5. Verify lack of constraint restriction.
6. Verify access constraints to allow discovery access to all.

7. Create UUDEX Subject “Tester/TEST/T3.1.9.4-2” with broadestAllowedDiscoveryAccess set to allowNone.
8. Verify the subject was created.
9. Verify lack of constraint restriction.
10. Verify access constraints to allow discovery access to no one.

11. Create UUDEX Subject “Tester/TEST/T3.1.9.4-3” with broadestAllowedDiscoveryAccess set to allowExcept = {Jack}.
12. Verify the subject was created.
13. Verify lack of constraint restriction.

14. Verify access constraints to allow discovery access to all except Jack.

15. Create UUDEX Subject “Tester/TEST/T3.1.9.4-4” with broadestAllowedDiscoveryAccess set to allowExcept = {Lyle, Mallory}.
16. Verify the subject was created.
17. Verify lack of constraint restriction.
18. Verify access constraints to allow discovery access to all except Lyle and Mallory.

19. Create UUDEX Subject “Tester/TEST/T3.1.9.3-5” with broadestAllowedDiscoveryAccess set to allowOnly = {Alice}.
20. Verify the subject was created.
21. Verify lack of constraint restriction.
22. Verify access constraints to allow discovery access to only Alice.

23. Create UUDEX Subject “Tester/TEST/T3.1.10.3-6” with broadestAllowedDiscoveryAccess set to allowOnly = {Bob, Carol}.
24. Verify the subject was created.
25. Verify lack of constraint restriction.
26. Verify access constraints to allow discovery access to only Bob and Carol.

3.1.10 Create Subject with Access Constraints (with Default Subject Policy Constraints)

Access constraints, in the form ACL specifications, are used to restrict which participants can create, view, publish to, or subscribe to subjects.

This set of tests validates the access constraint processing when there are Default Subject Policy constraints supplied that limit the individual subject creation requests.

Before starting this test, verify that all subjects with Participant “Tester”, Data Type “TEST” and Group Keys starting with “3.1.10” are not present. Remove them if they exist.

Note: there are multiple sub-tests and results for each test in this section.

3.1.10.1 Modify Default Subject Policies

This test modifies the Default Subject Policy with specific constraints used in these tests.

1. Verify the Default Subject Policy is configured as described in Table 2.1.
2. Modify the Default Subject Policy with the following constraints:

a. "broadestAllowedPublisherAccess": allowOnly = {Alice, Bob}
b. "broadestAllowedSubscriberAccess": allowOnly = {Carol, David}
c. "broadestAllowedManagerAccess: allowOnly = {Eve, Frank}
d. “BroadestAllowedDiscoveryAccess: allowOnly = {Carol, Frank}

3. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.

Note: there are multiple tests and results for each test in this section.

3.1.10.2 Broadest Publish Access

This test verifies subjects containing specific Broadest Publish Access constrains are properly created when the Default Subject Policy contains additional constraints.

4. Verify subject policies as described in Section 3.1.10.1.
5. Create UUDEX Subject “Tester/TEST/T3.1.10.2-1” with broadestAllowedPublisherAccess set to allowAll.
6. Verify the subject was created.
7. Verify lack of constraint restriction.
8. Verify access constraints to allow publish access to only Alice and Bob

9. Create UUDEX Subject “Tester/TEST/T3.1.11.2-2” with broadestAllowedPublisherAccess set to allowNone.
10. Verify the subject was created.
11. Verify lack of constraint restriction.
12. Verify access constraints to allow publish access to no one.

13. Create UUDEX Subject “Tester/TEST/T3.1.11.2-3” with broadestAllowedPublisherAccess set to allowExcept = {Jack}.
14. Verify the subject was created.
15. Verify lack of constraint restriction.
16. Verify access constraints to allow publish access to only Alice and Bob.

17. Create UUDEX Subject “Tester/TEST/T3.1.11.2-4” with broadestAllowedPublisherAccess set to allowExcept = {Lyle, Mallory}.
18. Verify the subject was created.
19. Verify lack of constraint restriction.
20. Verify access constraints to allow publish access to only Alice and Bob.

21. Create UUDEX Subject “Tester/TEST/T3.1.11.2-5” with broadestAllowedPublisherAccess set to allowExcept = {Alice}.
22. Verify the subject was created.
23. Verify lack of constraint restriction.
24. Verify access constraints to allow publish access to only Bob.

25. Create UUDEX Subject “Tester/TEST/T3.1.11.2-6” with broadestAllowedPublisherAccess set to allowOnly = {Alice}.
26. Verify the subject was created.
27. Verify lack of constraint restriction.
28. Verify access constraints to allow publish access to only Alice.

29. Create UUDEX Subject “Tester/TEST/T3.1.11.2-7” with broadestAllowedPublisherAccess set to allowOnly = {Bob, Carol}.
30. Verify the subject was created.
31. Verify lack of constraint restriction.
32. Verify access constraints to allow publish access to only Bob.

33. Create UUDEX Subject “Tester/TEST/T3.1.11.2-8” with broadestAllowedPublisherAccess set to allowOnly = {Carol}.
34. Verify the subject was created.
35. Verify lack of constraint restriction.
36. Verify access constraints to allow publish access to no one.

3.1.10.3 Broadest Subscribe Access

This test verifies subjects containing specific Broadest Subscribe Access constrains are properly created when the Default Subject Policy contains additional constraints.

1. Verify subject policies as described in Section 3.1.11.1.
2. Create UUDEX Subject “Tester/TEST/T3.1.11.3-1” with broadestAllowedSubscriberAccess set to allowAll.
3. Verify the subject was created.
4. Verify lack of constraint restriction.
5. Verify access constraints to allow subscribe access to only Carol and David.

6. Create UUDEX Subject “Tester/TEST/T3.1.11.3-2” with broadestAllowedSubscriberAccess set to allowNone.
7. Verify the subject was created.
8. Verify lack of constraint restriction.
9. Verify access constraints to allow subscribe access to no one.

10. Create UUDEX Subject “Tester/TEST/T3.1.11.3-3” with broadestAllowedSubscriberAccess set to allowExcept = {Jack}.
11. Verify the subject was created.
12. Verify lack of constraint restriction.
13. Verify access constraints to allow subscribe access to only Carol and David.

14. Create UUDEX Subject “Tester/TEST/T3.1.11.3-4” with broadestAllowedSubscriberAccess set to allowExcept = {Lyle, Mallory}.
15. Verify the subject was created.
16. Verify lack of constraint restriction.
17. Verify access constraints to allow subscribe access to only Carol and David.

18. Create UUDEX Subject “Tester/TEST/T3.1.10.3-5” with broadestAllowedSubscriberAccess set to allowExcept = {Carol}.
19. Verify the subject was created.
20. Verify lack of constraint restriction.
21. Verify access constraints to allow publish access to only David.

22. Create UUDEX Subject “Tester/TEST/T3.1.11.3-6” with broadestAllowedSubscriberAccess set to allowOnly = {Carol}.
23. Verify the subject was created.
24. Verify lack of constraint restriction.
25. Verify access constraints to allow subscribe access to only Carol.

26. Create UUDEX Subject “Tester/TEST/T3.1.11.3-7” with broadestAllowedSubscriberAccess set to allowOnly = {Carol, Alice}.
27. Verify the subject was created.
28. Verify lack of constraint restriction.
29. Verify access constraints to allow subscribe access to only Carol.

30. Create UUDEX Subject “Tester/TEST/T3.1.11.3-8” with broadestAllowedSubscriberAccess set to allowOnly = {Bob}.
31. Verify the subject was created.
32. Verify lack of constraint restriction.
33. Verify access constraints to allow subscribe access to no one.

3.1.10.4 Broadest Management Access

This test verifies subjects containing specific Broadest Management Access constrains are properly created when the Default Subject Policy contains additional constraints.

1. Verify subject policies as described in Section 3.1.11.1.
2. Create UUDEX Subject “Tester/TEST/T3.1.11.4-1” with broadestAllowedManagerAccess set to allowAll.
3. Verify the subject was created.
4. Verify lack of constraint restriction.
5. Verify access constraints to allow manage access to only Eve and Frank.
6. Create UUDEX Subject “Tester/TEST/T3.1.11.4-2” with broadestAllowedManagerAccess set to allowNone.
7. Verify the subject was created.
8. Verify lack of constraint restriction.
9. Verify access constraints to allow manage access to no one.
10. Create UUDEX Subject “Tester/TEST/T3.1.11.4-3” with broadestAllowedManagerAccess set to allowExcept = {Jack}.
11. Verify the subject was created.
12. Verify lack of constraint restriction.
13. Verify access constraints to allow manage access to only Eve and Frank.
14. Create UUDEX Subject “Tester/TEST/T3.1.11.4-4” with broadestAllowedManagerAccess set to allowExcept = {Lyle, Mallory}.
15. Verify the subject was created.
16. Verify lack of constraint restriction.
17. Verify access constraints to allow manage access to only Eve and Frank.

18. Create UUDEX Subject “Tester/TEST/T3.1.11.4-5” with broadestAllowedManagerAccess set to allowExcept = {Eve}.
19. Verify the subject was created.
20. Verify lack of constraint restriction.
21. Verify access constraints to allow publish access to only Frank.

22. Create UUDEX Subject “Tester/TEST/T3.1.11.4-6” with broadestAllowedManagerAccess set to allowOnly = {Eve}.
23. Verify the subject was created.
24. Verify lack of constraint restriction.
25. Verify access constraints to allow manage access to only Eve.

26. Create UUDEX Subject “Tester/TEST/T3.1.11.4-7” with broadestAllowedManagerAccess set to allowOnly = {Eve, Carol}.
27. Verify the subject was created.
28. Verify lack of constraint restriction.
29. Verify access constraints to allow manage access to only Eve.

30. Create UUDEX Subject “Tester/TEST/T3.1.11.4-8” with broadestAllowedManagerAccess set to allowOnly = {Carol}.
31. Verify the subject was created.
32. Verify lack of constraint restriction.
33. Verify access constraints to allow manage access to no one.

3.1.10.5 Broadest Discovery Access

This test verifies subjects containing specific Broadest Discovery Access constrains are properly created when the Default Subject Policy contains additional constraints.

1. Verify subject policies as described in Section 3.1.11.1.
2. Create UUDEX Subject “Tester/TEST/T3.1.11.5-1” with broadestAllowedDiscoveryAccess set to allowAll.
3. Verify the subject was created.
4. Verify lack of constraint restriction.
5. Verify access constraints to allow discovery access to only Carol and Frank.

6. Create UUDEX Subject “Tester/TEST/T3.1.11.5-2” with broadestAllowedDiscoveryAccess set to allowNone.
7. Verify the subject was created.
8. Verify lack of constraint restriction.
9. Verify access constraints to allow discovery access to no one.

10. Create UUDEX Subject “Tester/TEST/T3.1.11.5-3” with broadestAllowedDiscoveryAccess set to allowExcept = {Jack}.
11. Verify the subject was created.
12. Verify lack of constraint restriction.
13. Verify access constraints to allow manage access to only Carol and Frank.

14. Create UUDEX Subject “Tester/TEST/T3.1.11.5-4” with broadestAllowedDiscoveryAccess set to allowExcept = {Lyle, Mallory}.
15. Verify the subject was created.
16. Verify lack of constraint restriction.
17. Verify access constraints to allow manage access to only Carol and Frank.

18. Create UUDEX Subject “Tester/TEST/T3.1.11.5-5” with broadestAllowedDiscoveryAccess set to allowExcept = {Carol}.
19. Verify the subject was created.
20. Verify lack of constraint restriction.
21. Verify access constraints to allow publish access to only Frank.

22. Create UUDEX Subject “Tester/TEST/T3.1.11.5-6” with broadestAllowedDiscoveryAccess set to allowOnly = {Carol}.
23. Verify the subject was created.
24. Verify lack of constraint restriction.
25. Verify access constraints to allow manage access to only Carol.

26. Create UUDEX Subject “Tester/TEST/T3.1.11.5-7” with broadestAllowedDiscoveryAccess set to allowOnly = {Carol, Eve}.
27. Verify the subject was created.
28. Verify lack of constraint restriction.
29. Verify access constraints to allow manage access to only Carol.

30. Create UUDEX Subject “Tester/TEST/T3.1.11.5-8” with broadestAllowedDiscoveryAccess set to allowOnly = {Alice}.
31. Verify the subject was created.
32. Verify lack of constraint restriction.
33. Verify access constraints to allow manage access to no one.

3.1.11 Create Subject with Access Constraints (with Participant default constraints)

Access constraints, in the form ACL specifications, are used to restrict which participants can create, view, publish to, or subscribe to subjects.

This set of tests validates the access constraint processing when there are Participant Subject Policy constraints that limit the individual subject creation requests.

Before starting this test, verify that all subjects with Participant “Tester”, Data Type “TEST” and Group Keys starting with “3.1.12” are not present. Remove them if they exist.

Note: there are multiple tests and results for each test in this section.

3.1.11.1 Modify Default Subject Policy

This test modifies the Default Subject Policy with specific constraints used in these tests.

1. Verify the configuration of the Default Subject Policy is as specified in Table 2.1 (allow creation of all subjects and imposes no constraints or access limits).
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.
3. Create a Participant Subject Policy with the following constraints:
   a. "broadestAllowedPublisherAccess": allowOnly= {Alice, Bob}
   b. "broadestAllowedSubscriberAccess": allowOnly= {Carol, David}
   c. "broadestAllowedManagerAccess: allowOnly= {Eve, Frank}

Note: there are multiple tests and results for each test in this section.

3.1.11.2 Tests

These tests verify creation of subjects with specific constraints in the request when the Participant Subject Policy contains additional constraints.

Repeat all tests in Section 3.1.10.

3.1.12 Create Subject with Access Constraints (with Data Type default constraints)

Access constraints, in the form ACL specifications, are used to restrict which participants can create, view, publish to, or subscribe to subjects.

This set of tests validates the access constraint processing when there are Data Type Subject Policy constraints that limit the individual subject creation requests.

Before starting this test, verify that all subjects with Participant “Tester”, Data Type “TEST” and Group Keys starting with “3.1.12” are not present. Remove them if they exist.

Note: there are multiple tests and results for each test in this section.

3.1.12.1 Modify Default Subject Policy

This test modifies the Default Subject Policy with specific constraints used in these tests.

1. Verify the configuration of the Default Subject Policy is as specified in Table 2.1 (allow creation of all subjects and imposes no constraints or access limits).
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”
3. Create a Data Type Subject Policy with the following constraints:
   a. "broadestAllowedPublisherAccess": allowOnly= {Alice, Bob}
   b. "broadestAllowedSubscriberAccess": allowOnly= {Carol, David}
   c. "broadestAllowedManagerAccess: allowOnly= {Eve, Frank}

Note: there are multiple tests and results for each test in this section.

3.1.12.2 Tests

These tests verify creation of subjects with specific constraints in the request when the Data Type Subject Policy contains additional constraints.

Repeat all tests in Section 3.1.10.

3.1.13 Create Subject with Access Constraints (with Participant and Data Type default constraints)

Access constraints, in the form ACL specifications, are used to restrict which participants can create, view, publish to, or subscribe to subjects.

This set of tests validates the access constraint processing when there are Participant and Data Type Subject Policy constraints supplied that limit the individual subject creation requests.

Before starting this test, verify that all subjects with Participant “Tester”, Data Type “TEST” and Group Keys starting with “3.1.13” are not present. Remove them if they exist.

Note: there are multiple tests and results for each test in this section.

3.1.13.1 Modify Default Subject Policy

This test modifies the Default Subject Policy with specific constraints used in these tests.

1. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”
3. Create a Participant-Data Type Subject Policy with the following constraints:
   a. "broadestAllowedPublisherAccess": allowOnly= {Alice, Bob}
   b. "broadestAllowedSubscriberAccess": allowOnly= {Carol, David}
   c. "broadestAllowedManagerAccess: allowOnly= {Eve, Frank}

Note: there are multiple tests and results for each test in this section.

3.1.13.2 Tests

These tests verify creation of subjects with specific constraints in the request when the Participant-Data Type Subject Policy contains additional constraints.

Repeat all tests in Section 3.1.10.

3.1.14 Create Subject with Access Constraints for other Participants or Data Types

This set of tests verifies that the subject creation requests with Participants or Data Types other than those specified in Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records are created with the Default Subject Policy constraints.

1. Repeat the tests in Section 3.1.11, Section 3.1.12, and 3.1.13 specifying Participant “Tester1” rather than “Tester”.
2. Verify that the subjects are created with the Default Subject Policy constraints.
3. Repeat the tests in Section 3.1.11, Section 3.1.12, and 3.1.13 specifying Data Type “TEST1” rather than “TEST”.
4. Verify that the subjects are created with the Default Subject Policy constraints.
5. Repeat the tests in Section 3.1.11, Section 3.1.12, and 3.1.13 specifying Participant “Tester1” rather than “Tester” and Data Type “TEST1” rather than “TEST”.

6. Verify that the subjects are created with the Default Subject Policy constraints.

3.2 Subject Modification

These tests will verify the process for modifying existing subjects, using subject already created in previous tests in Section 3.1.

----- TESTS TO BE DETERMINED -----

3.3 Subject Discovery

These tests involve the ability to query a UUDEX Server for information about subjects. The subjects to be queried were created in the tests described in Section 3.1,

3.3.1 Discover Subject with no Constraints

----- TESTS TO BE DETERMINED -----

3.3.2 Discover Subject with Access Constraints – Permitted

----- TESTS TO BE DETERMINED -----

3.3.3 Discover Subject with Access Constraints – Denied

----- TESTS TO BE DETERMINED -----

4.0 Stand-alone Publishing

This set of tests involved publishing data to UUDEX subjects. Verification of the published data should be independently performed, for example by observing the message queues using an administrative interface.

Note: the tests in this section only verify the publish action. While subscription actions may be used to verify the publish fiction, they are not the subject of these tests. Testing of the subscription function is performed in the tests in Section 5.0 and Section 6.0.

4.1 Publish to Subject with No Constraints

This test verifies that a subject created with no constraints on size or performance will accept any number of messages of any size from any participant.

1. Create UUDEX Subject “Tester/TEST/T4.1” with no constraints for size, performance, or access specified.
2. Verify subject is created with no constraints.
3. Publish an arbitrary message to the subject.
4. Verify the message was successfully published.
5. Publish a haphazardly selected number of messages from a set of haphazardly selected participants of varying message lengths.
6. Verify that all messages are successfully published.

4.2 Publish with Client Response

This test verifies that messages can be published with acknowledgement provided by the UUDEX client that the message has been received. This function is similar to an email “read receipt” process.

NOTE: these tests assume a configuration with a UUDEX Publish client and two UUDEX Subscribe clients each on their own compute node. These nodes are referred to as “UUDEX Publish Node”, “UUDEX Subscribe Node 1”, and “UUDEX Subscribe Node 2” respectively in the test steps.

1. Create UUDEX Subject “Tester/TEST/T4.2-Response”.
2. Verify subject is created.
3. Create UUDEX Subject “Tester/TEST/T4.2-1”.
4. Verify subject is created.
5. On “UUDEX Subscribe Node 1”, subscribe to the subject “Tester/TEST/T4.2-1”.
6. On “UUDEX Subscribe Node 2”, subscribe to the subject “Tester/TEST/T4.2-1”.

7. On “UUDEX Publish Node”, publish an arbitrary message to subject “Tester/TEST/T4.2-1” specifying the following metadata in the header:
   a. asyncReplyFlag = true
   b. replyAddress = “Tester/TEST/T4.2-Response”
8. Verify the message was successfully published.
9. On “UUDEX Subscribe Node 1”, verify receipt of the message.
10. On “UUDEX Subscribe Node 2”, verify receipt of the message.

11. On “UUDEX Publish Node”, subscribe to the subject “Tester/TEST/T4.2-Response” and process all queued messages.
12. Retrieve the message (there should be two).
13. Verify two acknowledge responses are received.
14. Verify one message contains the following:
    a. Verb = “ACKNOWLEDGED”
    b. correlationID = the messageID of the message published in step 7
    c. Origin = Other Node 1
15. Verify the other message contains the following:
    a. Verb = “ACKNOWLEDGED”
    b. correlationID = the messageID of the message published in step 7
    c. Origin = Other Node 2

4.3 Publish to Subject with Performance Constraints

These tests verify the performance constraints (max number of messages, max size of messages)

4.3.1 Subject with Queue Total Size Constraints

These tests verify that subjects with total message queue size performance constraints behave as expected.

4.3.1.1 Block Publish

This test verifies the blocking of excessive messages when the total size of the message queue is full. Excessive publish requests should not be allowed – they should return an error.

1. Create UUDEX Subject “Tester/TEST/T4.3.1.1” with a maxQueueSizeKB of 10kB, and the fullQueueBehavior parameter set to “BLOCK_NEW”.

2. Verify subject is created with the requested parameters.
3. Publish a 3 kB message to the subject internally identified as “Message 1”.
4. Verify the message is successfully published.
5. Publish another 3 kB message to the subject internally identified as “Message 2”.
6. Verify the message is successfully published (2 messages, total queue size is 6 kB).
7. Publish another 3 kB message to the subject internally identified as “Message 3”.
8. Verify the message is successfully published (3 messages, total queue size is 9kB).
9. Publish another 3 kB message to the subject internally identified as “Message 4”.
10. Verify the message is not successfully published (publish call should return an error to the calling application).

4.3.1.2 Delete Oldest

This test verifies the deletion of old messages in the queue when the total size of the message queue is full. There should be no failure for excessive publish requests.

1. Create UUDEX Subject “Tester/TEST/T4.3.1.2” with a maxQueueSizeKB of 10kB, and the fullQueueBehavior parameter set to “PURGE_OLD”.
2. Verify subject is created with the requested parameters.
3. Publish a 3 kB message to the subject internally identified as “Message 1”.
4. Verify the message is successfully published.
5. Publish another 3 kB message to the subject internally identified as “Message 2”.
6. Verify the message is successfully published (2 messages, identified as “Message 1” and “Message 2”, total queue size is 6 kB).
7. Publish another 3 kB message to the subject internally identified as “Message 3”.
8. Verify the message is successfully published (3 messages, identified as “Message 1”, “Message 2”, and “Message 3”, total queue size is 9kB).
9. Publish another 3 kB message to the subject internally identified as “Message 4”.
10. Verify the message is successfully published (3 messages, identified as “Message 2”, “Message 3”, and “Message 4”, total queue size is 9kB).

4.3.2 Subject with Queue Message Number Constraints

These tests verify that subjects with number of messages in the queue performance constraints behave as expected.

4.3.2.1 Block Publish

This test verifies the blocking of excessive messages when number of messages in the queue indicates it is full. Excessive publish requests should not be allowed – they should return an error.

1. Create UUDEX Subject “Tester/TEST/T4.3.2.1” with a maxMessageCount of 3, and the fullQueueBehavior parameter set to “BLOCK_NEW”.
2. Verify subject is created with the requested parameters.
3. Publish a message to the subject internally identified as “Message 1”.
4. Verify the message is successfully published.
5. Publish another message to the subject internally identified as “Message 2”.
6. Verify the message is successfully published (2 messages, identified as “Message 1” and “Message 2”).
7. Publish another message to the subject internally identified as “Message 3”.
8. Verify the message is successfully published (3 messages, identified as “Message 1”, “Message 2”, and “Message 3”).
9. Publish another message to the subject internally identified as “Message 4”.
10. Verify the message is not successfully published (publish call should return an error to the calling application).

4.3.2.2 Delete Oldest

This test verifies the deletion of old messages when number of messages in the queue indicates it is full. There should be no failure for excessive publish requests.

1. Create UUDEX Subject “Tester/TEST/T4.3.2.2” with a maxMessageCount of 3, and the fullQueueBehavior parameter set to “PURGE_OLD”.
2. Verify subject is created with the requested parameters.
3. Publish a message to the subject internally identified as “Message 1”.
4. Verify the message is successfully published.
5. Publish another message to the subject internally identified as “Message 2”.
6. Verify the message is successfully published (2 messages, identified as “Message 1” and “Message 2”).
7. Publish another message to the subject internally identified as “Message 3”.
8. Verify the message is successfully published (3 messages, identified as “Message 1”, “Message 2”, and “Message 3”).
9. Publish another message to the subject internally identified as “Message 4”.

10. Verify the message is successfully published (3 messages, identified as “Message 2”, “Message 3”, and “Message 4”).

4.4 Publish to Subject with Access Constraints

These tests verify that the access constraints (and allowances) on subjects work as expected.

These tests use the following UUDEX subjects:

- “Tester/TEST/T4.4-1” with the following attributes:
  a. Owner “Alice”
  b. Publish allowOnly “Bob”
- “Tester/TEST/T4.4-2” with the following attributes:
  c. Owner “Alice”
  d. Publish allowExcept “Carol”

4.4.1 Implicit Publish Allowed

This test verifies that the owner of the subject is implicitly allowed to publish to the subject.

1. Use UUDEX subject ”Tester/TEST/T4.4-1”.
2. Publish a message from participant “Alice”.
3. Verify message was published.

4.4.2 Publish Allowed

This test verifies explicit and implicit the publish allow.

1. Use UUDEX subject ”Tester/TEST/T4.4-1”.
2. Publish a message from participant “Bob”.
3. Verify message was published.

4. Use UUDEX subject ”Tester/TEST/T4.4-2”.
5. Publish a message from participant “Bob”.
6. Verify message was published.

4.4.3 Publish Denied

This test verifies explicit and implicit the publish deny processing

1. Use UUDEX subject “Tester/TEST/T4.4-1”.
2. Publish a message from participant “Carol”.
3. Verify message was not published – check for error.

4. Use UUDEX subject “Tester/TEST/T4.4-2”.
5. Publish a message from participant “Carol”.
6. Verify message was not published – check for error.

5.0 Stand-alone Subscribing (Data Delivery)

This set of tests involves subscribing to data in UUDEX subjects. Subjects used were created in the tests described in Section 3.1, Subscribed data may be data published in Section 4.0, or may be published specifically for this test.

In this test, the subscribe function returns the data to the caller for immediate processing.

5.1 Subscribe to Subject with No Constraints

This test verifies the ability to subscribe to subjects that have no access constraints and received published messages.

1. Create UUDEX Subject “Tester/TEST/T5.1” with no access constraints.
2. Subscribe to UUDEX Subject “Tester/TEST/T5.1” from participant “Alice”.
3. Publish data to UUDEX Subject “Tester/TEST/T5.1” from participant “Bob”.
4. Verify “Alice” received data published by “Bob”.

5.2 Subscribe to Subject with Access Constraints

These tests use the following UUDEX subjects which must be created:

- “Tester/TEST/T5.2-1” with the following attributes:
  a. Owner “Alice”
  b. Subscribe allowOnly “Bob”
  c. Publish allowAll
- “Tester/TEST/T5.2-2” with the following attributes:
  d. Owner “Alice”
  e. Subscribe allowExcept “Carol”
  f. Publish allowAll

5.2.1 Implicit Subscribe Allowed

This test verifies the ability for the owner of a subject to implicitly subscribe to it

1. Use UUDEX subject ”Tester/TEST/T5.2-1”.
2. Subscribe from participant “Alice”.
3. Publish a message from participant “David”.
4. Retrieve message by participant “Alice”.

5.2.2 Subscribe Allowed

This test verifies the ability of specific users to subscribe implicitly or explicitly to subjects and retrieve messages.

1. Use UUDEX subject ”Tester/TEST/T5.2-1”.
2. Subscribe from participant “Bob”.
3. Publish a message from participant “David”.
4. Retrieve message by participant “Bob”.

5. Use UUDEX subject ”Tester/TEST/T5.2-2”.
6. Subscribe from participant “Bob”.
7. Publish a message from participant “David”.
8. Retrieve message by participant “Bob”.

5.2.3 Subscribe Denied

This test verifies the ability to explicitly or implicitly deny subscribe access to a subject.

1. Use UUDEX subject ”Tester/TEST/T5.2-1”.
2. Subscribe from participant “Carol”.
3. Verify subscribe request denied.

4. Use UUDEX subject ”Tester/TEST/T5.2-2”.
5. Subscribe from participant “Carol”.
6. Verify subscribe request denied.

5.2.4 Subscription Fulfillment – Queue Priority

----- TEST TO BE DETERMINED -----

6.0 Stand-alone Subscribing (Data Notification)

This set of tests involves subscribing to data in UUDEX subjects. Subscribed data may be data published in Section 4.0, or may be published specifically for this test.

In this test, the subscribe function returns an indication that data is available in the Subject but requires a separate action by the subscribing application to retrieve the data.

6.1 Subscribe to Subject with No Constraints

This test verifies the ability to receive publish notifications to a subject.

1. Create UUDEX Subject “Tester/TEST/T6.1”.
2. Subscribe to UUDEX Subject “Tester/TEST/T6.1” from participant “Alice”.
3. Publish data to UUDEX Subject “Tester/TEST/T6.1” from participant “Bob”.
4. Verify “Alice” received a publication notification about Bob’s publication.
5. Verify Alice can retrieve Bob’s message.

6.2 Subscribe to Subject with Access Constraints

These tests use the following UUDEX subjects which must be created:

- “Tester/TEST/T6.2-1” with the following attributes:
  a. Owner “Alice”
  b. Subscribe allowOnly “Bob”
  c. Publish allowAll
- “Tester/TEST/T6.2-2” with the following attributes:
  d. Owner “Alice”
  e. Subscribe allowExcept “Carol”
  f. Publish allowAll

6.2.1 Subscribe Allowed

This test verifies the ability of specific users to subscribe implicitly or explicitly to subjects and retrieve messages.

6. Use UUDEX subject ”Tester/TEST/T6.2-1”.
7. Subscribe from participant “Bob”.

8. Publish a message from participant “David”.
9. Verify “Bob” received publish notification about David’s publication.
10. Verify Bob can retrieve David’s message.

11. Use UUDEX subject ”Tester/TEST/T6.2-2”.
12. Subscribe from participant “Bob”.
13. Publish a message from participant “David”.
14. Verify “Bob” received publish notification about David’s publication.
15. Verify Bob can retrieve David’s message.

7.0 Query / Response Testing

This set of tests involves testing the query capabilities of the UUDEX server.

7.1 Get List of Available Subjects (no constraints)

----- TEST TO BE DETERMINED -----

7.2 Get List of Available Subjects (with constraints)

----- TEST TO BE DETERMINED -----

7.3 Keyword Search

This test performs a keyword search of messages in a subject with specific keywords. The messages must already exist in the subject. These tests are generic versions of tests performed in Section 9.0.

----- TEST TO BE DETERMINED -----

8.0 Performance Stress Tests

This set of tests involves sending large amounts of data through UUDEX. For this test, “large” is tested as both a large number of small messages, and a smaller number of large messages. A separate test is specified that sends both large and small messages.

8.1 Large Number of Messages

----- TEST TO BE DETERMINED -----

8.2 Large Messages

----- TEST TO BE DETERMINED -----

8.3 Variable Messages

----- TEST TO BE DETERMINED -----

9.0 Application Tests

This set of tests involved using UUDEX Publish and Subscribe functions to transfer data from one application to another and verify its receipt.

9.1 PowerWorld Data Exchange

This test simulates the exchange of power system measurement information between two organizations. Each organization is simulated on a different computer node using the PowerWorld power flow solver using the same power system model. Power system data Is exchanged between a simulated Transmission Operator (on node PW1), and its Reliability Coordinator (on node PW2).

Data is extracted from a running PowerWorld instance (PW1), published to a UUDEX subject. An application associated with a separate PowerWorld instance (PW2) then subscribes to the UUDEX subject, receives the data, and inserts it into the PW2 instance. Values changed in the PW1 instance should show up in the PW2 instance. Power flow results should be the same in both instances of PowerWorld after the data transfer and execution of the PowerWorld simulation.

1. Verify the configuration of the Default Subject Policy is as specified in Table 2.1 (allow creation of all subjects and imposes no constraints or access limits).
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.
3. Create the “Tester/TEST/PW-App” UUDEX Subject with no constraints for size, performance, or access specified.
4. Verify operational status of PW1 by changing a value and allowing PowerWorld to re-solve the power flow and display the results.
5. Verify operational status of PW2 by changing a different value and allowing PowerWorld to re-solve the power flow and display the results.
6. Start the data extraction and data receipt processes (python scripts). A subset of all available PowerWorld data is transferred from PW1 to PW2.
   a. Make note the list of available transferred values.
   b. Make note the frequency of transfer (e.g., 10 seconds).
7. Start the data receiving application.
   a. Verify it is subscribing to the “Tester/TEST/PW-App” subject.
8. Select a point from the available list of transferred values in PW1, and manually change it.
   a. Verify that PW1 re-solves the power flow.
9. Monitor UUDEX to ensure that the changed value is published to UUDEX.
10. Wait until the configured periodic time for data transfer has passed (e.g., 10 seconds).
11. Monitor UUDEX to verify that the changed value is subscribed by the receiving application.
12. Verify the modified values properly arrives and is inserted into the PW2 instance.

13. Allow PW2 to re-solve the power flow.
14. Verify that the re-solved power flows are the same in PW1 and PW2.

15. Change the order of the points in the message being transferred.
16. Verify that the points still are properly inserted into the PW dataset.

17. Add or remove points from the message being transferred.
18. Verify that the points still are properly inserted into the PW dataset.

9.2 OE-417 and Physical Incident Reporting

OE-417 reports may be published as either XML or PDF format. XML formats are more compact, and more readily processed by parsers and database applications by the subscribing application. A special application may be required for both publishing and subscribing to properly format the files prior to publishing them to UUDEX, or to re-convert the UUDEX content to a file after subscribing to the data.

Physical Security Incident Reports (PSIR) are native JSON format and require no additional processing for use by UUDEX.

9.2.1 OE-417 Data Exchange (XML data)

This test verifies publishing DOE-OE-417 reports using XML format with no access constraints.

1. Verify the configuration of the Default Subject Policy is as specified in Table 2.1 (allow creation of all subjects and imposes no constraints or access limits).
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.
3. Create the “Tester/TEST/OE-417” UUDEX Subject with no constraints for size, performance, or access specified.
4. Create an OE-417 report form in PDF using the Adobe Acrobat OE-417 report form.
5. Export the data to XML format using the Adobe Acrobat program.
6. Publish the OE-417 XML report specifying XML format.
7. Subscribe to the data on a different node.
8. Save the XML data to a file.
9. Import the XML data into the PDF version of the OE-417 using Adobe Acrobat and verify its content.

9.2.2 OE-417 Report Exchange (PDF)

This test verifies publishing DOE-OE-417 reports using PDF format with no access constraints.

1. Verify the configuration of the Default Subject Policy is as specified in Table 2.1 (allow creation of all subjects and imposes no constraints or access limits).
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.
3. Create or use the “Tester/TEST/OE-417” UUDEX Subject from Section 9.2.1 with no constraints for size, performance, or access specified.
4. Create or use the PDF OE-417 report from Section 9.2.1.
5. Save the PDF form.
6. Publish the OE-417 PDF report specifying PDF format.
7. Subscribe to the data on a different node.
8. Extract and save the data to a file.
9. Verify the file is a valid PDF file and contains the correct data.

9.2.3 OE-417 Data Exchange (with constraints)

This test verifies that OE-417 reports can be published securely to a UUDEX instance and only retrieved (subscribed) by authorized participants.

1. Modify the “Tester/TEST/OE-417” UUDEX Subject from Section 9.2.1 to add the following access constraints:
   a. broadestAllowedPublisherAccess to allowAll
   b. broadestAllowedSubscriberAccess to allowOnly={E-ISAC, NERC}
2. Create a PDF OE-417 report, or use the PDF OE-417 report from Section 9.2.2.
3. Save the PDF form.
4. Publish the OE-417 PDF report specifying PDF format by participant Alice.
5. Subscribe to the data by participant E-ISAC.
6. Extract and save the data to a file.
7. Verify the file is a valid PDF file and contains the correct data.

8. Publish the OE-417 PDF report specifying PDF format by participant Alice.
9. Subscribe to the data by participant Bob.
10. Verify that the subscribe operation fails.

9.2.4 OE-417 Data Exchange (with submission acknowledgement)

This test verifies the acknowledgement function of the DOE-OE-417 report submission.

1. Create or use the “Tester/TEST/OE-417” UUDEX Subject from Section 9.2.1 or as modified in Section 9.2.3.
2. Create or use the PDF OE-417 report from Section 9.2.2.
3. Save the PDF form.
4. Create the “Tester/TEST/Alice-Reply” UUDEX Subject with the following constraints:
   a. broadestAllowedPublisherAccess to allowOnly {E-ISAC, NERC}.
   b. broadestAllowedSubscriberAccess to allowOnly Alice.
5. Publish the OE-417 PDF report to UUDEX subject “Tester/TEST/OE-417” by participant Alice, specifying PDF format, and the following metadata in the publish header:
   a. asyncReplyFlag = true
   b. replyAddress = “Tester/TEST/Alice-Reply”.
6. Subscribe to UUDEX subject “Tester/TEST/Alice-Reply” by participant Alice.
7. Subscribe to UUDEX subject “Tester/TEST/OE-417” by participant E-ISAC.
8. Extract and save the OE-417 PDF report to a file.
9. Verify the file is a valid PDF file and contains the correct data.
10. Publish a response to UUDEX subject “Tester/TEST/Alice- Reply” by participant E-ISAC, with no data payload, and the following metadata:
    a. correlationID = messageID (UUID) from initial publish in step 5.
    b. verb = “ACKNOWLEDGED”.
11. Data returned to Participant Alice.
12. Participant Alice verifies the correlationID and verb.

9.2.5 Physical Security Incident Event

This test verifies the publishing of Physical Security Incident Event reports with no access constraints.

1. Verify the configuration of the Default Subject Policy is as specified in Table 2.1 (allow creation of all subjects and imposes no constraints or access limits).
2. Verify that no Participant-Data Type Subject Policy, Participant Subject Policy, or Data Type Subject Policy records exist for UUDEX Participant “Tester” or data type “TEST”.

3. Create the “Tester/TEST/PSIR” UUDEX Subject with no constraints for size, performance, or access specified.
4. Generate a Physical Security Incident Report JSON document.
5. Format the UUDEX Physical Security Incident Report message.
6. Publish the PSIR report message.
7. Subscribe to the data.
8. Save the PSIR document from the UUDEX PSIR message.
9. View the PSIR and verify its contents.

Note: the website https://json-editor.github.io/json-editor/ may be of assistance in generating and displaying JSON documents using their JSON Schema files.

Note: this test may also be repeated with acknowledgement processing similar to that in Test 9.2.4.

9.3 Security and Event Notifications

These tests perform a keyword searches of messages in a subject with specific keywords. The messages must already exist in the subject.

----- TEST TO BE DETERMINED -----

9.3.1 Threat and Vulnerability Reporting (STIX)

----- TEST TO BE DETERMINED -----

9.3.2 Patch Notifications

----- TEST TO BE DETERMINED -----

9.4 Power System Model Updates

----- TEST TO BE DETERMINED -----

9.4.1 Full Model Updates

----- TEST TO BE DETERMINED -----

9.4.2 Partial Model Updates

----- TEST TO BE DETERMINED -----

9.5 RCIS

----- TEST TO BE DETERMINED -----

PNNL-30771

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)

www.pnnl.gov