SNIA Extensible Access Method (XAM) Reviewer's Guide

Summary of Changes from Version 0.6 to 1.0

Disclaimer

This Reviewers' Guide is provided as a convenience to those reviewing Extensible Access Method (XAM) v1.0. This Reviewers' Guide is a good faith effort to identify the substantive changes that have been made since the most recent IP review of Extensible Access Method (XAM) v0.6, but SNIA does not guarantee that all changes are identified. The obligations of SNIA Members are defined by the SNIA IP Policy and are not limited to those parts of the material that have changed since the last review. The SNIA does not represent that all substantive changes are listed or described fully and accurately in the Reviewers' Guide. The reviewing SNIA Member is fully responsible for its response, whether or not the Reviewers' Guide provides complete and accurate information regarding the substantive changes.

Document Names

The names of both of the specification documents in version 0.6 have been changed to meet standardization requirements. The JAVA API is a new document.

Part 1: Architecture Specification

The chapter numbers changed as a consequence of reorganization of the Architecture Specification to meet standardization requirements:

- The Foreword, Introduction and Chapter 1 (Scope) have been added since the 0.6 version
- Chapter 2 (References) was Annex A in the 0.6 version
- Chapter 3 (Terms and Conventions) was sections 2.1 and 2.2 of the 0.6 version
- Chapter 4 (Business Overview) has been added since the 0.6 version
- Chapter 5 (Overview of the XAM Architecture) was the remainder of Chapter 2 in the 0.6 version.
    - The portion of chapter 5 before Section 5.1 is based on Section 2.1 of the 0.6 version.
    - Sections 5.1 through 5.8 are based on Sections 2.4 through 2.11 of the 0.6 version (chapter number increases by 3, section number decreases by 3).
- Chapters 6 through 11 were Chapters 3 through 8 of the 0.6 version (chapter number increases by 3).
- Annex A (Toolkit) was Annex B in the 0.6 version
- Annex B (Canonical XSet Interchange Format) was Annex C in the 0.6 Version

Chapters 1 through 4

Chapters 2 and 3 are normative (contain implementation requirements), Chapters 1 and 4 are informative. The significant changes in chapters 2 and 3 are:

- Significant additions to References in chapter 2
- Significant expansion of list of defined Terms in Section 3.1
- Addition of typographical conventions in Section 3.2

Chapter 5 - Overview of the XAM Architecture

Chapter 5 has been extensively edited. As Chapter 5 is an overview, the changes made to it generally reflect the detailed changes in the following chapters that correspond to the overviews in Chapter 5.

Chapter 6 - XAM Objects and Common Operations

This was chapter 3 in the 0.6 version. It contains extensive editorial changes, many of which are minor technical clarifications. The following is a summary of technical changes:

- Section 6.3.4 - added specification of the CRC polynomial used in XUIDs.
- Section 6.3.5:
    - Added check that name of a newly created field must be valid for the field creation to succeed.
    - Forbid checking that reserved fields in XUIDs are zero. This should ensure that these fields can be used in the future.
    - Added required syntax checks for MIME type, and specified that the validity of the MIME type is not to be checked.

- Section 6.4.1 - added explanations of property operations.
- Section 6.4.2 - this section and the containsField method are new.

- Section 6.5 - expanded specification of relationship of secondary objects to primary objects, including:
    - Creation restriction - secondary object can only be created by factory method in primary object.
    - Closure restriction - secondary object must be closed before closing primary object.
- Section 6.5.1 - Added asynchronous read and write methods to XStream, also added abandon method.
- Section 6.5.2 - XIterator. The GetFieldIterator method has been renamed to openFieldIterator. The hasNext method has been added.

- Section 6.6.1 - extensive changes to the reader and writer XStream finite state machines. These changes include support for the new methods (cf. 6.5.1).
- Section 6.6.1.1 - added Read Corrupt state to the XStream FSM for readers.
- Section 6.6.2 - extensive changes to the XIterator finite state machine. These include support for the new hasNext method.

Chapter 7 - XAM Library and XSystems

This was Chapter 4 in the 0.6 version. Extensive editorial changes have been made, many of which are minor technical clarifications. The following technical changes have been made:

- Section 7.1.3 - XAM.Connect call renamed to XAMLibrary.connect
- Section 7.1.4 - XAM Library log level field contents have been specified, log path and verbosity fields have been added. The debug level field has been removed.

- Section 7.2.1 - VIM initialization specification added (last paragraph).
- Section 7.2.2 - Extensively changed, requires detailed review. The primary changes are addition of an XSet copy method in both synchronous and asynchronous forms and the addition of the isXSetRetained method.
- Section 7.2.3 - extensive changes to XSystem field names and details. Most of this is caused by changes elsewhere in the spec (e.g., retention fields) or oversight in the 0.6 version (e.g., access fields were omitted in 0.6).
- Section 7.2.3 - lower limit on maxFieldsPerXSet is now 2^14 instead of 2^16.
- Section 7.3 - The authentication state machine in Section 7.3.1 is basically unchanged from the 0.6 version (e.g., the state diagram figure has not changed), but there are numerous small changes and clarifications in the detailed specification of the transitions in Table 20. These small changes and clarifications have corresponding effects on Sections 7.3.2 through 7.3.3.3.

Chapter 8 - XSet Operations

This was Chapter 5 in the 0.6 version. Extensive editorial changes have been made, many of which are minor technical clarifications. The following technical changes have been made:

- Section 8.2.3 - changed names of fields, added XUID time field and clarified contents of create time field.
- Section 8.2.4 - this section is new.
- Section 8.4 - added XSystem.copyXSet method. Most method names have changed, slightly.
- Section 8.5 - This section contains extensive changes and should be thoroughly reviewed. The restricted and unrestricted state machines have changed due to the addition of the copyXSet method and a restricted mode of the createXSet method.
- Section 8.8 - This section contains extensive changes, and should be thoroughly reviewed. The description of the Canonical XSet Interchange Format changes in Annex B (see below) may assist the reviewer.
- Section 8.8.5 - this section is new
- Section 8.10 - added an asynchronous XSet copy method. There are numerous small changes to the XAsync methods in Section 8.10.1 and to the XAsync state machine in Section 8.10.2

Chapter 9 - XSet Management

This was Chapter 6 in the 0.6 version. Extensive editorial changes have been made, many of which are minor technical clarifications. The following technical changes have been made:

- Addition of (multiple) application defined XSet retention criteria to the XSet retention discipline (primarily Section 9.2.1 and Section 9.3.2, but there are some related changes in other sections, including Section 9.1.2 and Section 9.3.3):
    - An application may define one or more named XSet retention criteria, using retention value and policy management properties, in addition to the previously specified XSet base (formerly known as minimum) and event retention criteria.
    - A named XSet retention criteria is indicated by a .xset.retention.list.<retention identifier> xam_string property.
    - A committed XSet shall contain .xset.retention.list.base and xset.retention.list.event properties
    - A complete XSet retention criteria includes 2 elements: a retention criteria enablement flag, an end time which is specified as a start time combined with a millisecond duration. These retention criteria elements have specified behavior similar to the previously specified event-based retention criteria with respect to value and policy management hierarchy, update restrictions, and methods.
    - A committed XSet shall contain .xset.retention.base.enabled and .xset.retention.base.starttime properties.
    - XSet.setBaseRetention and XSet.applyBaseRetention methods have been added to preserve previously specified base (aka minimum) retention behavior
    - The "Get Actual" methods for XSet retention have been extended into XSet.getActualRetentionDuration and XSet.getActualRetentionEnabled methods which require an application to supply a <retention identifier>
- Added XSystem.isXSetRetained method (Section 9.2.1.1)
- Addition of XSet shredding to the XSet deletion discipline (primarily Section 9.2.2 and Section 9.3.2, but there are some related changes in other sections, including Section 9.1.1 and Section 9.3.3).
    - XSet shredding defined to be the destruction of the binary recording of an XSet once the XSet is deleted from the XSystem. The method of destruction is outside the scope of XAM. XSet shredding support is optional for XAM Storage System vendors to implement.
    - XSystem support for XSet shredding is indicated in the .xsystem.deletion.shred xam_boolean property
    - Specific XSet shredding is indicated by the following properties in the XSet policy management hierarchy
        - .xset.deletion.shred
        - .xset.deletion.shred.policy
        - .xset. management.policy
- Added XSet.getActualShred method (Section 9.3.6)

Chapter 10 - Query

This was Chapter 7 in the 0.6 version. Extensive editorial changes have been made, many of which are minor technical clarifications. The following technical changes have been made:

- Property names must be quoted with Double Quotes, for example:
  `select ".xset.xuid" where ".xset.time.commit" > date('2008-03-17T00:00')`
- All the example queries have been changed to reflect changes in various predefined field names in the XAM system
- Escape codes quoting and arbitrary characters have been added to string expressions.
- IEEE-754 Floating point concepts for INF and NAN have been introduced to the query behavior. (primarily Section 10.4.3)
- The formal ABNF grammar has changed to accommodate the previously mentioned changes, and to correct defects identified during the review period. (Section 10.6)

Chapter 11 - Security

This was Chapter 8 in the 0.6 version. Extensive editorial changes have been made, many of which are minor technical clarifications. The following technical changes have been made:

- Authorization Granules (Section 11.3.1.1)
  - Numerous updates to Authorization granules to reflect changes elsewhere in XAM Architecture and API.
  - Write-user granule renamed to write-application.
  - Delete granule added to separate out authorization of XSystem.deleteXSet
  - Job-query and Job-query-commit granules expanded to cover all jobs, not just the query job (the only job defined in the XAM v1 standard is the query job; all others are vendor-specific). The word "query" was removed from the granule names as a consequence.
  - Retention-event granule expanded to cover creation of new retention identifiers beyond "base" and "event".
- Job-only XSet expanded to cover all jobs, not just query. (Section 11.3.1.1.3)
- Authorization Roles (Section 11.3.1.2)
  - The "query" roles were expanded to cover all jobs, and the word"query" was removed from their names.
  - The xam-unrestricted role was renamed to xam-create
  - The xam-user role was renamed to xam-job-commit
  - The xam-create-delete role was added to encompass the delete granule.
- Access Control (Section 11.3.2)
  - The create-delete granule was removed from access control. Create-delete actions are now controlled solely by authorization.
  - The write-user granule was renamed to write-application.
- The following three items were added to clarify aspects of Access Control (Section 11.3.2) that were implicit in the 0.6 version. In all cases, the clarification is not a technical change to the intent of the 0.6 specification:

- XSystem.AccessXSet is used to determine whether an XSet is visible in a XAM session. If an XSet cannot be accessed for reading, it's not visible.
- The authorization granule requirements and finite state machine effects of the access control policy methods have now been specified.
- There is a clear statement that XSet Access Control is optional to support.

Annex A - Toolkit

The entire Toolkit is now mandatory to support (Normative). It was optional to support (Informative) in 0.6.

A close method has been added to the XUIDIterator (Section A.2).

Two boolean methods have been added to determine whether a field is a property vs. an XStream (Section A.4).

Annex B - Canonical XSet Interchange Format

This was in a preliminary form in the 0.6 version. Extensive work has been done to complete its design and specification, including:

- Added copyright SNIA notice
- Added xs:import of namespaces
- Added comment section describing general format of Canonical XSet XML
- Added xs:any attribute to FieldAttrs attribute group to allow vendors to add attributes (see Section 8.8.5 for rules on how this may be used)
- Added complexType PropertyType to encapsulate typing of Property values. This was added to allow the use of XSD built-in types (xs:int, xs:float, xs:string, xs:dateTime, and xs:boolean) to describe the type of the value. This allows proper expression of infinity (Inf) and not-a-number (NaN) as defined by XAM QL and IEEE-754. Removed the <value> element from the property complex type for this reason.
- Added policy section to allow export/import of XSystem policies. The policy may have a name (e.g. for a policy list) and zero or more policy properties.
- Added xstream minOccurs="0". If no minOccurs is specified, the default is one. Since an XSet may have no XStreams, minOccurs must be set to zero
- Dropped xset "offset" attribute. Was a holdover from the original design. Offset is handled in the table of contents section of the Canonical XSet.
- Added xs:any attribute to xstream element to allow vendors to add attributes (see Section 8.8.5 for rules on how this may be used)

Part 2: C API and Part 3: Java API

This C API document has been extensively changed since 0.6, with many technical details filled in. Most of the changes either add to incomplete specifications in the 0.6 version or respond to changes in the Architecture specification since the 0.6 version. The JAVA API document is new; it did not exist in the 0.6 version of the XAM documents.