XAM (eXtensible Access Method) Hands-On Lab for Developers

SNIA Storage Developer Conference

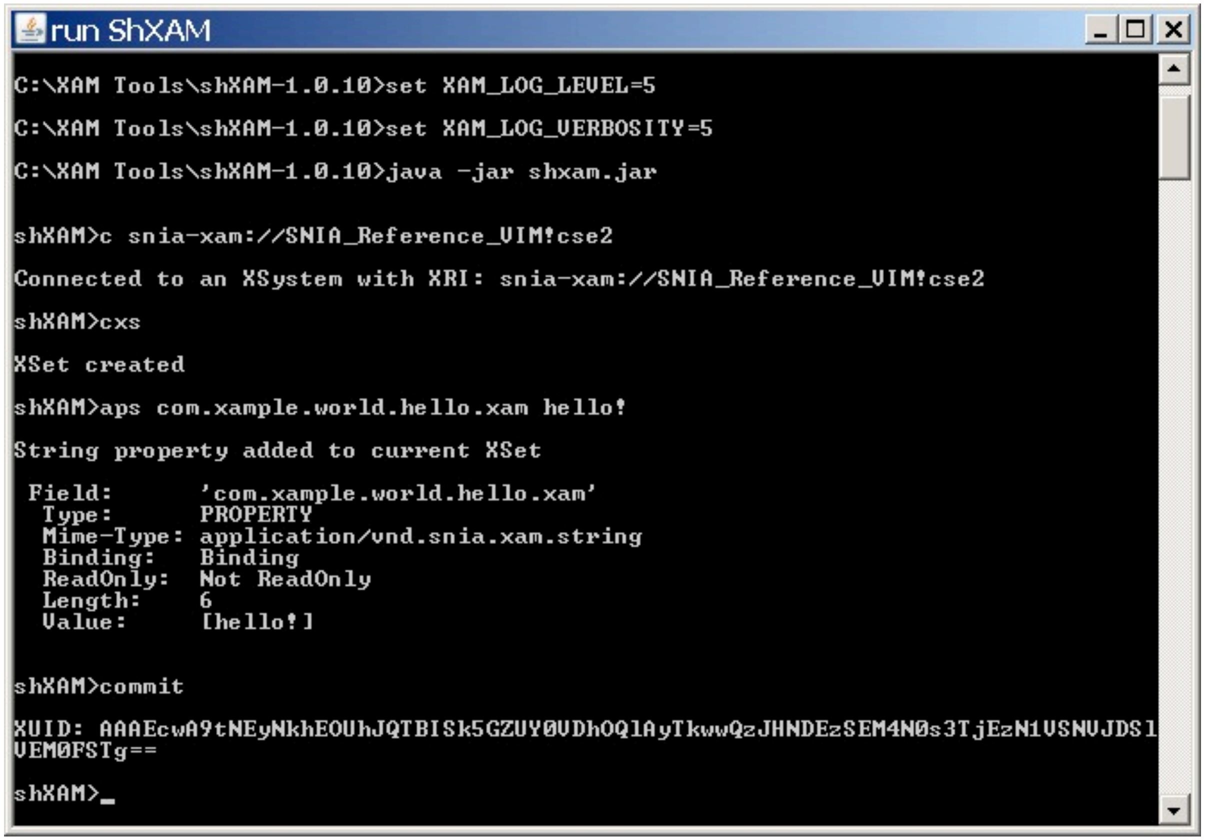
September 2008

Mike Horgan, EMC
horgan_mike@emc.com

Introducing the XAM API

- XAM brings powerful fixed-content storage capabilities to software solutions
- XAM-integrated applications are able to:
  - Store, retrieve, and delete self-describing objects
  - Associate policies with objects
- The XAM API is accessible to both C and Java
- This workshop will use shXAM to demonstrate core features of the XAM API
  - shXAM is a training and debugging tool
  - shXAM exposes a slightly simplified version of the XAM API

Introducing shXAM



```
C:\XAM Tools\shXAM-1.0.10>set XAM_LOG_LEVEL=5

C:\XAM Tools\shXAM-1.0.10>set XAM_LOG_VERBOSITY=5

C:\XAM Tools\shXAM-1.0.10>java -jar shxam.jar


shXAM>c snia-xam://SNIA_Reference_VIM!cse2

Connected to an XSystem with XRI: snia-xam://SNIA_Reference_VIM!cse2

shXAM>cxs

XSet created

shXAM>aps com.xample.world.hello.xam hello!

String property added to current XSet

 Field:      'com.xample.world.hello.xam'
  Type:      PROPERTY
  Mime-Type: application/vnd.snia.xam.string
  Binding:   Binding
  ReadOnly:  Not ReadOnly
  Length:    6
  Value:     [hello!]


shXAM>commit

XUID: AAAEcwA9tNEyNkhEOUhJQTBISk5GZUY0VDhOQlAyTkwwQzJHNDEzSEM4N0s3TjEzN1VSNVJDSl
VEM0FSTg==

shXAM>_
```

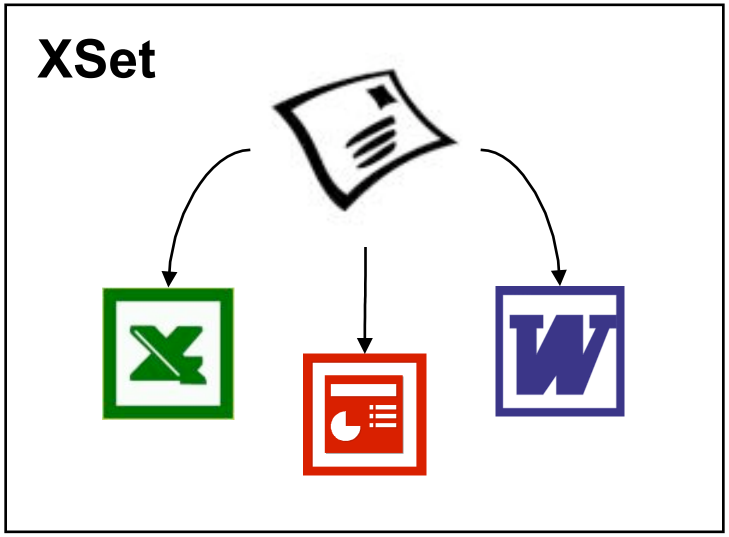
- shXAM is a CLI interface to the XAM API
- Implemented using the XAM Java API 1.0
- Data is stored using snapshot of SNIA Reference VIM

XSystem

- An **XSystem** represents a logical storage device
- An **XSystem Instance** is a XAM API object that represents a connection to a XAM Storage System
- Client applications use an **XSystem Resource Identifier** connect string (**XRI**) to connect to an XSystem
- `connect XRI [name] [password]`
  - Connect to an XSystem
  - Optionally specify data for SASL PLAIN authentication e.g., `connect snia-xam://SNIA_Reference_VIM!localhost`

XSet

- An **XSet** Represents a logical unit of storage on the XSystem
- May reference one or more data objects
  - Multiple medical images pertaining to a single study
  - An email and all associated attachments
- Contains the metadata that describes the stored data
- Retrieved via a XUID
- `createXSet`
  - Create a new XSet object
  - XSet is created in memory, not on storage system!



XAM Fields

- **Fields: Data** and **metadata** associated with an XSet
- **Data** (XAM streams):
  - User-defined objects: emails, office documents, images, etc.
- **Metadata** (XAM properties):
  - Information used to describe the data: filename, patient-name, etc.
  - Simple data types (stypes): Boolean, Int, Float, String, Datetime, XUID
- All fields have an associated MIME type
- An arbitrary number of fields may be associated with an XSet
- Flat hierarchy

XAM Streams (Data)

- **XStream** API mimics POSIX file I/O syntax
- shXAM File-Centric Stream Commands:
  - `addStream fieldName [MIMEType] [binding]`
  - `openStream fieldName {readonly, writeonly, appendonly}`
  - `write fileName`
  - `read fileName [length]`
  - `seek offset {SEEK_SET (0), SEEK_POS (1), SEEK_END (2)}`
  - `tell`
  - `streamClose`
- XStream Convenience Routines
  - `writeStream fileName`
  - `appendStream fileName`
  - `readStream fileName`

XAM Properties (Metadata)

Properties allow applications to create self-describing objects

- Name/Value pair
- shXAM Property Commands:
- Property Creation:
  - `addProperty<TYPE> fieldName value [binding]`
  - where `<TYPE>` is a simple type: (Boolean, Int, Float, String, Datetime, XUID)
- shXAM *state* vs. XAM API

- Property Access:
  - `findField fieldName`
  - `viewFields [match-substr]`
- Property Modification:
  - `modifyValue value`
  - `modifyFieldBinding binding`
- Property Deletion:
  - `deleteField fieldName`

Field Iteration

- `viewFields [match-substr]`
- Substring filter can be applied:
  - `viewFields email_header`
- This will match:
  - com.example.email_app.email_header.from
  - com.example.email_app.email_header.to
  - com.example.email_app.email_header.subject
- Careful application of reverse-dns naming convention can emulate hierarchical field structure.
- Set ‘field pointer’ to a particular field
  - `findField com.example.email_app.email_header.subject`

XUID

- XUID is the permanent name for an XSet
  - Assigned by XAM Storage System on XSet commit
  - XUIDs are globally unique
- XUID native format is binary sequence (10 – 80 bytes)
  - Base64 (RFC 2045) recommended for printable interchange
- XSet's XUID has a strict relationship with the XSet's 'Binding' fields
  - If a 'Binding' field is modified, a new XSet with a new XUID is created on commit
  - The original XSet remains in original form

XUID Format

0	1	2	3	4	5	6	7	8	9	10	…	78	79
reserved (zero)	Vendor OID			reserved (zero)	XUID length	XUID CRC-16		opaque data			…	data end	

Storing, Retrieving, and Deleting

- XSet Management Operations:
  - `createXSet`
  - `openXSet XUID mode`
    - Creates an in-memory XSet from the XSystem
  - `commit`
    - Serializes the in-memory XSet to the XSystem
    - XUID representing the XSet is returned to the client application
  - `deleteXSet XUID`
    - Deletes the specified XSet

shXAM Examples

- Example Command Sequence:
  - createXSet
  - addPropertyBoolean org.snia.xamples.isTemp true
  - addPropertyInt org.snia.xamples.theAnswer 42
  - addPropertyFloat org.snia.xamples.number.euler 2.71828
  - findField org.snia.xamples.isTemp
  - modifyValue false
  - viewFields xample
  - deleteField org.snia.xamples.theAnswer
  - addStream org.snia.xamples.logo image/jpeg
  - writeStream c:\snia_logo.jpg
  - viewFields snia
  - commit
  - **XUID: AAAEcwA9MzIGSE03UTZJUDJGVEVEZUJBTUpDVDBURFVRRzZHNDEzNDMyU05LQjBQRIJNQ1AxRDZJVE9GVg==**

Retention

- Ensure that objects exist until they are eligible for deletion
- **Retention**
  - **StartTime**
  - **Duration**
  - **Enabled**
  - **Binding**
- **Hold**
  - A temporary lock on an object
  - Most commonly applied during litigation activities

shXAM – Applying Retention/Hold

- shXAM Retention Functions
  - `setBaseRetention binding duration`
  - `createRetention binding retentionID`
  - `setRetentionEnabledFlag retentionID binding enabled`
  - `setRetentionDuration retentionID binding duration`
  - `setRetentionStarttime retentionID binding`
- shXAM Hold Functions
  - `holdXSet XUID holdID`
  - `releaseXSet XUID holdID`

Query

- Query [queryString]
- Examples:
  - query
  - query 'select ".xset.xuid"’
  - query 'select ".xset.xuid" where exists (“com.emc.email_app.email_header”)

shXAM Miscellaneous

- `help (?)`
- shXAM command abbreviations
- Field commands for XSystem and XAM Library

shXAM – Cheat Sheet

XSystem

- connect *XRI* (c)
- viewSystemFields [substr] (vsf)
- query [queryString] (q)
- xsetInfo XUID (xsi)

XSet

- createXSet mode (sn)
- openXSet *XUID* mode (so)
- deleteXSet *XUID* (de)
- commit (c)
- viewFields [substr] (vf)

XSet Properties

- addProperty<TYPE> fieldName value [binding]
- findField fieldName (ff)
- modifyValue value (mv)
- modifyFieldValue name value (mfv)

XStream

- addStream fieldName [MIMEtype] [binding] (as)
- openStream name {readonly, writeonly, appendonly} (ev)
- tell
- seek offset whence
- read fileName [length]
- write filename
- writeStream fileName (ws)
- readStream fileName (rs)
- appendStream fileName (wsa)
- streamClose

Retention

- setBaseRetention binding duration (sr)
- createRetention binding retentionID (cr)
- setRetentionEnabledFlag retentionID binding enabled (sre)
- setRetentionDuration retentionID binding duration (srd)
- setRetentionStarttime retentionID binding (srs)
- isXSetRetained XUID (isr)

Hold

- holdXSet XUID holdID (h)
- releaseXSet XUID holdID (r)

XAM C API ←→ shXAM Mapping

shXAM	**XAM API**
connect	XAMLibrary_Connect
viewSystemFields	*XAM Field Iteration**
query	*XAM Query**
createXSet	XSystem_CreateXSet
openXSet	XSystem_OpenXSet
deleteXSet	XSystem_DeleteXSet
commit	XSet_Commit
viewFields	*XAM Field Iteration**
addProperty<TYPE>	XAM_Create<TYPE>
findField	N/A*
modifyFieldValue	XAM_Set<TYPE>
modifyFieldBinding	XAM_SetFieldAsBinding/XAM_SetFieldAsNonbinding
addStream	XAM_CreateXStream

XAM C API ←→ shXAM Mapping (cont.)

shXAM	XAM API
write	XStream_Write*
read	XStream_Read*
seek	XStream_Seek
tell	XStream_Tell
setBaseRetention	XSet_SetBaseRetention
setBaseRetention	XSet_SetBaseRetention
createRetention	XSet_CreateRetention
setRetentionEnabledFlag	XSet_SetRetentionEnabledFlag
setRetentionDuration	XSet_SetRetentionDuration
setRetentionStarttime	XSet_SetRetentionStarttime
isXSetRetained	XSystem_IsXSetRetained
holdXSet	XSystem_HoldXSet
releaseXSet	XSystem_ReleaseXSet

For More Information

- SNIA XAM Home
  - http://www.snia.org/xam/home
- SNIA XAM Google Group (Friendly Q&A! 😃)
  - http://groups.google.com/group/xam-developers-group
- SNIA XAM Initiative
  - http://www.snia.org/apps/org/workgroup/xam/
- SNIA FCAS TWG
  (XAM Technical WorkGroup)
  - http://www.snia.org/apps/org/workgroup/fcastwg/
- SNIA SDK TWG
  (XAM Software Developer Kit Technical Work Group)
  - http://www.snia.org/apps/org/workgroup/xamsdktwg/index.php