Errata in "Information Management – Extensible Access Method (XAM) – Part 2: C API" v1.0 SNIA FCAS TWG, June 16, 2009 – Approved Version

Summary of errata:

- 5.3.4.3 XSystem_Close: Open XStream also blocks close
- 5.3.6.5 XSystem_AccessXSet: Permission checks
- 5.3.8.3.1 XSet_CreateRetention: Add two error cases and clarify a Note
- 5.3.8.3.2 XSet_SetRetentionEnabledFlag: Add an error case
- 5.3.8.3.3 XSet_ApplyRetentionEnabledPolicy: Add an error case and clarify another one
- 5.3.8.3.4 XSet_SetRetentionDuration: Add an error case
- 5.3.8.3.5 XSet_ApplyRetentionDurationPolicy: Add an error case and clarify another one
- 5.3.8.3.6 XSet_SetRetentionStarttime: Add an error case
- 5.3.8.3.7 XSet_SetBaseRetention: Clarify binding behavior
- 5.3.8.3.7 XSet_SetBaseRetention: Fix the units of duration
- 5.3.8.7.1 XSet_GetActualRetentionDuration: Add error cases
- 5.3.8.7.2 XSet_GetActualRetentionEnabled: Add error cases
- 5.3.8.7.3 XSet_GetActualAutoDelete: Add error cases
- 5.3.8.7.4 XSet_GetActualShred: Add error cases
- 5.3.10.3.3 XASync_GetXOPID: Add an error case
- 5.3.10.3.4 XASync_GetStatus: Add an error case
- 5.3.10.3.5 XASync_GetXSet: Add an error case
- 5.3.10.3.6 XASync_GetXStream: Add an error case
- 5.3.10.3.7 XASync_GetXUID: Add an error case
- 5.3.10.3.8 XASync_GetBytesRead: Add an error case
- 5.3.10.3.9 XASync_GetBytesWritten: Add an error case

NOTE: All of the above changes also apply to the corresponding VIM interfaces

- Section 4.2.4.1, Table 1 – Field stypes - Minimum XUID length improperly stated
- 8 Sections:.xset.xuidtime
- Section A.2: .xsystem.job.list.query
- Section 5.4.1: Add additional log fields
- Section 5.4.2, Table 3: .xsystem.job.xam.job.query.continuence.supported
- Section 5.4.2: .xsystem.job.job.xam.job.query.continuence.supported
- Section 5.4.2: SASL mechanism list is a list of booleans, not strings
- Section A.2: .xsystem.job.query.continuence.supported
- Section A.2: .xsystem.job.query.level1.supported
- Section A.2: .xsystem.job.query.level2.supported
- Section A.2: .xset.retention.list.base and .xset.retention.list.event
- Section A.2: .xsystem.retention.duration.policy.list and .xsystem.retention.enabled.policy.list
- Section C.2 (new): Base64 conversion methods
- Section D: Add API mappings for base64 conversion methods

Errata

5.3.4.3 XSystem_Close: Open XStream also blocks close

The final error condition is:

- There are open XSets.

Change "XSets" to "XSets or XStreams".

Also apply the above change to 6.2.4.4 VIM_XSystem_Close.

5.3.6.5 XSystem_AccessXSet: Permission checks

The specification of the inMode Parameter is:

- inMode: The value is the bitwise OR of the access 'permissions' to be checked (R_OK for read permission, WU_OK for write-user permission, WS_OK for write-system permission, D_OK for delete - in addition there are composite permissions W_OK (WU_OK|WS_OK) and for ALL_OK (R_OK|W_OK|D_OK)).

Replace that specification with:

- inMode: The value is the bitwise OR of the access 'permissions' to be checked (R_OK for read permission, WU_OK for write-user permission, WS_OK for write-system permission, D_OK for delete, H_OK for hold, RE_OK for retention event, J_OK for job and JC_OK for job commit). In addition there are composite permissions W_OK (WU_OK|WS_OK), RW_OK (R_OK|W_OK) and ALL_OK (RW_OK|D_OK|H_OK|RE_OK|J_OK|JC_OK).

Also apply the above change to 6.2.6.5 VIM_XSystem_AccessXSet.

5.3.8.3.1 XSet_CreateRetention: Add an error case

Add two error cases:

> The retention identifier is "base"
>
> The retention identifier is "event" and the binding input parameter is FALSE.

In the Note: in the description, change:

> Changing this field from binding to nonbinding (or vice versa)

to:

> Creating a binding set of retention criteria

Also apply the above changes to 6.2.7.3.1 VIM_XSet_CreateRetention.

5.3.8.3.2 XSet_SetRetentionEnabledFlag: Add an error case

Add an error case:

> The retention identifier is “base”

Also apply the above change to 6.2.7.3.2 VIM_XSet_SetRetentionEnabledFlag.

5.3.8.3.3 XSet_ApplyRetentionEnabledPolicy: Add an error case and clarify another one

Add an error case:

> The retention identifier is “base”

The final error case is:

> Enabled is being set to after it was set to true

Change it to:

> The applied policy has the effect of disabling retention for this retention ID after it was previously enabled.

Also apply the above changes to 6.2.7.3.3 VIM_XSet_ApplyRetentionEnabledPolicy.

5.3.8.3.4 XSet_SetRetentionDuration: Add an error case

Add an error case:

> The retention identifier is “base”

Also apply the above changes to 6.2.7.3.4 VIM_XSet_SetRetentionDuration.

5.3.8.3.5 XSet_ApplyRetentionDurationPolicy: Add an error case and clarify another one

Add an error case:

> The retention identifier is “base”

The final error case is:

> The field already exists on the XSet, and the specified duration value is less than the existing duration value.

Change it to:

> The applied policy has the effect of decreasing the duration for this retention ID.

Also apply the above changes to 6.2.7.3.5 VIM_XSet_ApplyRetentionDurationPolicy.

5.3.8.3.6 XSet_SetRetentionStarttime: Add an error case

Add an error case:

> The retention identifier is “base”.

Also apply the above changes to 6.2.7.3.6 VIM_XSet_SetRetentionStarttime.

5.3.8.3.7 XSet_SetBaseRetention: Clarify binding behavior

The last sentence in the first paragraph of the Description is:

> These fields will have their binding attribute set according to the binding flag that is set by the application.

Replace that sentence with:

> The *.xset.retention.base.duration* field will have its binding attribute set according to the binding flag that is set by the application. The *.xset.retention.list.base* is always a binding field.

The two Notes at the end of the Description are:

> **Note:** Changing this field from binding to nonbinding (or vice versa) will result in a new XSet being created and a new XUID being assigned on a successful commit.
>
> **Note:** When an XSet instance containing the field *.xset.retention.list.base* is first committed, the field *.xset.retention.base.starttime* will be created and have its value set to *.xset.xuidtime*.

In the first Note, replace "this field" with "*.xset.retention.base.duration*" (italicized).
In the second Note, add "as a binding field" after "will be created" and change "*.xset.xuidtime*" to "*.xset.time.xuid*" (italicized).

Also apply the above changes to 6.2.7.3.7 VIM_XSet_SetBaseRetention.

5.3.8.3.7 XSet_SetBaseRetention: correct units for duration

Change parameter description from:

> inDuration is a xam_int containing the amount of time (measured in minutes from the time of commit) to retain the XSet. Zero indicates no retention, while a negative one (-1) indicates infinite retention.

to:

> inDuration is a xam_int containing the amount of time (measured in milliseconds from the time of commit) to retain the XSet. Zero indicates no retention, while a negative one (-1) indicates infinite retention.

Also apply the above changes to 6.2.7.3.7 VIM_XSet_SetBaseRetention.

5.3.8.7.1 XSet_GetActualRetentionDuration: Add error cases

Add error cases:

- The XSet instance was imported and contains a retention duration policy that does not exist
- The XSet instance was imported and contains a retention duration policy that does not match the policy in the XSystem

Also apply the above changes to 6.2.7.7.1 VIM_XSet_GetActualRetentionDuration.

5.3.8.7.2 XSet_GetActualRetentionEnabled: Add error cases

Add error cases:

- The XSet instance was imported and contains a retention enabled policy that does not exist
- The XSet instance was imported and contains a retention enabled policy that does not match the policy in the XSystem

Also apply the above changes to 6.2.7.7.2 VIM_XSet_GetActualRetentionEnabled.

5.3.8.7.3 XSet_GetActualAutoDelete: Add error cases

Add error cases:

- The XSet instance was imported and contains an auto-delete policy that does not exist
- The XSet instance was imported and contains an auto-delete policy that does not match the policy in the XSystem

Also apply the above changes to 6.2.7.7.3 VIM_XSet_GetActualAutoDelete.

5.3.8.7.4 XSet_GetActualShred: Add error cases

Add error cases:

- The XSet instance was imported and contains a shred policy that does not exist
- The XSet instance was imported and contains a shred policy that does not match the policy in the XSystem

Also apply the above changes to 6.2.7.7.4 VIM_XSet_GetActualShred.

5.3.10.3.3 XASync_GetXOPID: Add an error case

Add an error case:

The operation was programmatically halted

Also apply the above change to 6.2.10.3.3VIM_XAsync_GetXOPID.

5.3.10.3.4 XASync_GetStatus: Add an error case

Add an error case:

> The operation was programmatically halted

Also apply the above change to 6.2.10.3.4VIM_XAsync_GetStatus.

5.3.10.3.5 XASync_GetXSet: Add an error case

Add an error case:

> The operation was programmatically halted

Also apply the above change to 6.2.10.3.5VIM_XAsync_GetXSet.

5.3.10.3.6 XASync_GetXStream: Add an error case

Add an error case:

> The operation was programmatically halted

Also apply the above change to 6.2.10.3.6VIM_XAsync_GetXStream.

5.3.10.3.7 XASync_GetXUID: Add an error case

Add an error case:

> The operation was programmatically halted

Also apply the above change to 6.2.10.3.7 VIM_XAsync_GetXUID.

5.3.10.3.8 XASync_GetBytesRead: Add an error case

Add an error case:

> The operation was programmatically halted

Also apply the above change to 6.2.10.3.8VIM_XAsync_GetBytesRead.

5.3.10.3.9 XASync_GetBytesWritten: Add an error case

Add an error case:

> The operation was programmatically halted

Also apply the above change to 6.2.10.3.9VIM_XAsync_GetBytesWritten.

Section 4.2.4.1, Table 1 – Field stypes

Length (in bytes) cell of **XUID** row is improperly stated as **8 - 80** bytes. The minimum XUID length is 9; thus the cell should contain **9 - 80**.

8 Sections: .xset.xuidtime

Replace .xset.xuidtime with .xset.time.xuid in the following sections

- Section 5.3.8.3.7
- Section 5.3.8.3.8
- Section 6.2.7.3.7
- Section 6.2.7.3.8
- Section A.4 XSet_SetBaseRetention
- Section A.4 XSet_ApplBaseRetentionPolicy
- Section B.1 XSet_SetBaseRetention
- Section B.1 XSet_ApplBaseRetentionPolicy

Section A.2: .xsystem.job.list.query

Replace *.xsystem.job.list.query* with *.xsystem.job.list.xam.job.query*

Section 5.4.1: Add additional log fields

Add the following fields to Table 2:

Field Name	stype	MIME type
.xam.log.append	xam_boolean	application/vnd.snia.xam.boolean
.xam.log.max.rollovers	xam_int	application/vnd.snia.xam.int
.xam.log.max.size	xam_int	application/vnd.snia.xam.int

And add the following field descriptions below the table:

.xam.log.append: indicates whether to append to an existing log file (TRUE) or overwrite (FALSE). The default value is FALSE.

.xam.log.max.rollovers: indicated the number of previous log files to retain when starting a new log file. The default value is 1.

.xam.log.max.size: indicates the maximum size in bytes that a log file may reach before a new log file is started. The default value is 1GB (2^30 = 1,073,741,824 bytes).

Section 5.4.2, Table 3: .xsystem.job.xam.job.query.continuence.supported

Replace *.xsystem.job.xam.job.query.continuence.supported with .xsystem.job.xam.job.query.continuance.supported*

Section 5.4.2: .xsystem.job.job.xam.job.query.continuence.supported

Replace *.xsystem.job.job.xam.job.query.continuence.supported with .xsystem.job.xam.job.query.continuance.supported*

Section 5.4.2: SASL Mechanism List

The elements of the SASL mechanism list are booleans, not strings. In Table 3, for the "*.xsystem.auth.SASLmechanism.list.<mechanism>*" row, change the stype from "xam_string" to "xam_boolean" and change the MIME type from "application/vnd.snia.xam.string" to "application/vnd.snia.xam.boolean".

Section A.2: .xsystem.job.query.continuence.supported

Replace *.xsystem.job query.continuence.supported with .xsystem.job.xam.job.query.continuance.supported*

Section A.2: .xsystem.job.query.level1.supported

Replace *.xsystem.job query.level1.supported with .xsystem.job.xam.job.query.level1.supported*

Section A.2: .xsystem.job.query.level2.supported

Replace *.xsystem.job query.level2.supported with .xsystem.job.xam.job.query.level2.supported*

Section A.2: .xset.retention.list.base and .xset.retention.list.event

After the line:

```
static const char* const XAM_XSET_RETENTION_LIST = ".xset.retention.list";
```

add the following 2 lines

```
static const char* const XAM_XSET_RETENTION_LIST_BASE = ".xset.retention.list.base";
static const char* const XAM_XSET_RETENTION_LIST_EVENT = ".xset.retention.list.event";
```

Section A.2: .xsystem.retention.duration.policy.list and .xsystem.retention.enabled.policy.list

After the line and the blank line that follows it:

```
static const char* const XAM_XSYSTEM_SHRED_POLICY_LIST = ".xsystem.deletion.shred.policy.list";
```

add the following 2 lines

```
static const char* const XAM_XSYSTEM_RETENTION_DURATION_POLICY_LIST =
        ".xsystem.retention.duration.policy.list";
static const char* const XAM_XSYSTEM_RETENTION_ENABLED_POLICY_LIST =
        ".xsystem.retention.enabled.policy.list ";
```

Section C.2: Base64 conversion methods

Add the following new Section C.2 and subsections

C.2 Base64 Conversion

To store XUID values in printable formats, it is recommended that applications base64 encode them.

C.2.1 base64_encode

Syntax prototype:

```
void
base64_encode (const char *inSrcBuf, int inSrcLen, char *outDstBuf, int *outDstLen);
```

Parameters:

- inSrcBuf is a pointer to a character string to be encoded in base64.
- inSrcLen is the length of the input character string.
- outDstBuf is a pointer to a buffer where the base64-encoded output is to be placed.
- outDstLen is the length of the base64-encoded output

  Note: In order to avoid overwriting other data, make sure that outDstBuf is at least (inSrcLen+2)/3 * 4 bytes long

Error conditions:

- None

C.2.2 base64_decode

Syntax prototype:

```
void
base64_decode (const char *inSrcBuf, int inSrcLen, char *outDstBuf, int *outDstLen);
```

Parameters:

- inSrcBuf is a pointer to a character string to be decoded from base64.
- inSrcLen is the length of the input character string.
- outDstBuf is a pointer to a buffer where the decoded output is to be placed.
- outDstLen is the length of the decoded output

  Note: In order to avoid overwriting other data, make sure that outDstBuf is at least (inSrcLen+3)/4 * 3 bytes long

Error conditions:

- None

Section D: Base64 conversion method mappings

In the final 2 rows of Table D.1:

- Change the " Methods in C API Spec" entry for XUIDToString from "N/A" to "base64_encode".
- Change the " Methods in C API Spec" entry for StringToXUID from "N/A" to "base64_decode".