

Scoping Document for Launch In Context subprofile

Duane Baldwin, John Crandall, Martine Wedlake

July 9, 2008

Target SMI-S Release: 1.5

1 Description

Add Launch-in-Context (LIC) support in SMI-S via the addition of a new LIC subprofile that leverages the Access Points subprofile. This new subprofile will be developed in the Core TWG and will be part of the SMI-S 1.5 Common Profiles book.

Providing solutions to storage customers that enable the integration of various tools, SRMs, and element managers they use to run their business will promote the adoption of SMI-S and provide real value to the customer. Launch-in-Context is one way to provide a basic level of integration and seamless access to storage management tools & applications essential to the customer.

1.1 Short Description

Add Launch-in-Context (LIC) support in SMI-S via the addition of a new LIC subprofile that leverages the Access Points subprofile.

1.2 Feature Mapping Requirements

Identify if you believe this scoping document addresses any requirements from the Feature Mapping document. An example table is shown below:

Req #	Requirement	Capability

1.3 Specific Objectives within Scope

- Represent LIC points with meta-data to explain how to establish launch (e.g., context, parameters, etc.)
- Link LIC point to the management domain
- Provide concrete representation of LIC capabilities
-

2 Dependencies

2.1 TWG Dependencies

No direct dependencies outside of Core. But being a common subprofile, various device types may choose to implement.

2.2 DMTF WG Dependencies

Currently none. New classes are being proposed as SNIA classes.

2.3 Other Standards Dependencies

None.

2.4 Author Dependencies

2.5

Book	Clause / Profile	Author	Agreement

For the last column, the encoding is:

- **No** – means that the champion (author of this scoping document) **does not** have agreement from the Author.
- **Yes** – means that the champion (author of this scoping document) **does** have agreement from the Author.
- **Transfer** – means that the champion will accept having the clause / profile transferred to him / her, AND the author has agreed to the transfer.

NOTE: The Author column is to be filled out with the name of the current author as identified on the Editors and Authors page. For a list of the current Authors of SMI-S see <http://www.snia.org/members/smis/EditorsAndAuthors>

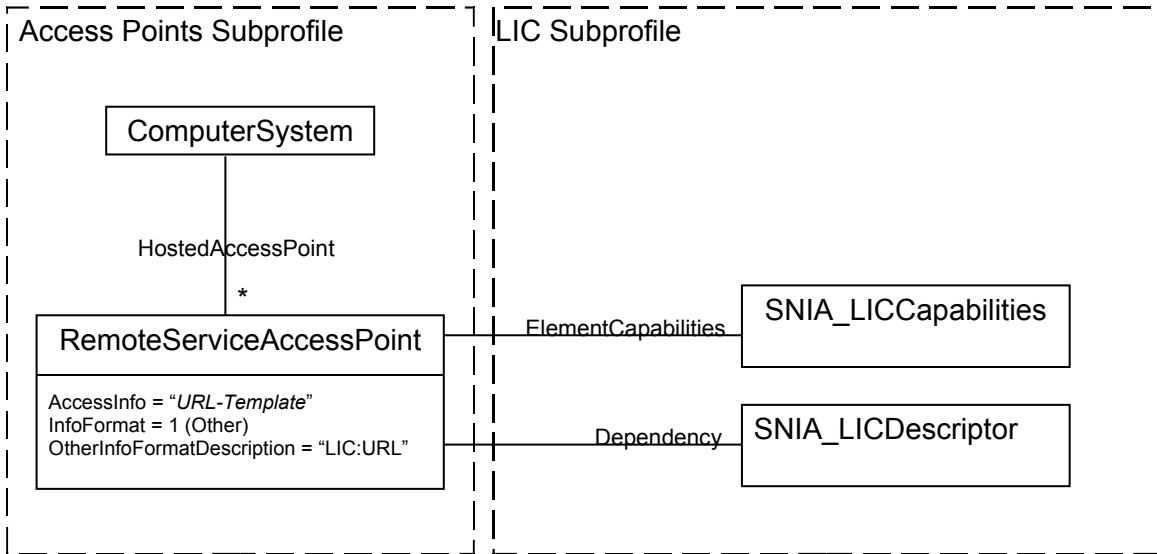
3 Modeling Requirements/Detailed Description

3.1 Overview

Add Launch-in-Context (LIC) support in SMI-S via the addition of a new LIC subprofile that leverages the Access Points subprofile. Goals:

- Represent LIC points with meta-data to explain how to establish launch (e.g., context, parameters, etc.)
- Link LIC point to the management domain
- Provide concrete representation of LIC capabilities

3.2 Instance Diagram:



3.3 RemoteServiceAccessPoint:

Represents a single launch point into GUI where additional launch points will be represented by separate instances. Only URL-based LIC entry points are supported currently; other entry points will be added later (e.g., direct invocation).

•

Required Parameters:

AccessInfo is the URL template of the launch point. It is formatted like a normal URL, except that named variables may be inserted within the URL that will be filled in dynamically by the client. These named variables have the syntax `${name}` where the name is represented as a parameter within the `SNIA_LICDescriptor`. For example:

```
http://host1:8020/DisplayHosts?host="${hostID}"
```

InfoFormat will be set to 1 (Other).

OtherInfoFormatDescription will be set to "LIC:URL" to represent a Launch-in-Context Template URL.

3.4 SNIA_LICCapabilities:

Extends from CIM_Capabilities with list of flags indicating the capabilities of the launch point using a uint16 array called SupportedAction.

The MOF is represented below:

```

    • //

// =====
// SNIA_LICCapabilities
// =====

[Description(
    "Describes the launch-in-context capabilities of a launch point.")]
class SNIA_LICCapabiliites : CIM_Capabilities {

    [Description(
        "Enumeration indicating what operations are accesible through "
        "the launch point."),
    ValueMap {
        "1", "2", "3", "4", "5", "6", "7", "8", "9", "10",
        "11", "12", "13", "14", "15", "16", "17", "18", "19", "20",
        "21", "22", "23", "24", "25", "26", "27", "28", "29", "30",
        "31", "32", "33", "34", "35", "36", "37", "38", "39", "40",
        "41", "42", "43", "44", "45", "46", "47", "48", "49", "50",
        "51", "52..999", "1000..10000", ".."};

    Values {

        "Manage Volume", // 1
        "View Volume", // 2
        "Create Volume", // 3
        "Modify Volume", // 4
        "Delete Volume", // 5

        "Manage Storage Pool", // 6
        "View Storage Pool", // 7
        "Create Storage Pool", // 8
        "Modify Storage Pool", // 9
        "Delete Storage Pool", // 10

        "Manage Extent", // 11
        "View Extent", // 12
        "Create Extent", // 13
        "Modify Extent", // 14
        "Delete Extent", // 15

        "Manage Host Port", // 16
        "View Host Port", // 17
    }
}

```

SMI Scoping Document

```
"Create Host Port", // 18
"Modify Host Port", // 19
"Delete Host Port", // 20

"Manage Host Mapping", // 21
"View Host Mapping", // 22
"Create Host Mapping", // 23
"Modify Host Mapping", // 24
"Delete Host Mapping", // 25

"Manage Replica", // 26
"View Replica", // 27
"Create Replica", // 28
"Modify Replica", // 30
"Delete Replica", // 31

"Manage SynchronizationGroup", // 32
"View SynchronizationGroup", // 33
"Create SynchronizationGroup", // 34
"Modify SynchronizationGroup", // 35
"Delete SynchronizationGroup", // 36

"Manage Synchronization", // 37
"View Synchronization", // 38
"Modify Synchronization", // 39

"Manage Statistics", // 40
"View Volume Statistics", // 41
"View Disk Drive Statistics", // 42
"View Computer System Statistics", // 43
"View Front-End Ports Statistics", // 44
"View Back-End Ports Statistics", // 45
"View Remote Replica Group Statistics", // 46

"View Logs", // 47
"Upgrade Firmware/Software", // 48
"Reboot/Reset Device", // 49
"Configure Device", // 50
"Manage Power Supply", // 51
"SNIA Reserved", // 52..999
"Vendor Reserved", // 1000..10000
"SNIA Reserved"}]
uint16 SupportedActions[];
};
```

3.5 SNIA_LICDescriptor:

The SNIA_LICDescriptor represents the catalogue of possible parameters that can be included within the URL-Template of the RemoteServiceAccessPoint.

SMI Scoping Document

It is recommended that there be a single instance with all parameters identified; however, it is permitted to have different instances associated to different RemoteServiceAccessPoint instances if desired.

The MOF representation of the SNIA_LICDescriptor is as follows:

```
//
// =====
// SNIA_LICDescriptor
// =====
//

[Description(
    "Describes the launch-in-context parameters for a launch point.")]
class SNIA_LICDescriptor : CIM_ManagedElement {
    [Key, Description (
        "Within the scope of the instantiating Namespace, InstanceID "
        "opaquely and uniquely identifies an instance of this class. ")]
    string InstanceID;

    [Description(
        "The name of the parameter."),
    ArrayType("Indexed")]
    string ParameterName[];

    [Description(
        "The description of the parameter."),
    ArrayType("Indexed")]
    string ParameterDescription[];

    [Description(
        "The data type of the parameter."),
    ArrayType("Indexed"),
    ValueMap {"0", "1", "2", "3", "4", "5", ".."},
    Values {"uint", "sint", "string", "bool", "real", "datetime",
        "SNIA Reserved"}]
    uint32 ParameterType[];

    [Description(
        "The lower range for uint Data"),
    ArrayType("Indexed")]
    uint32 ParameterDataUintMin[];

    [Description(
        "The upper range for uint Data"),
    ArrayType("Indexed")]
    uint32 ParameterDataUintMax[];
}
```

SMI Scoping Document

```
[Description(
    "The lower range for sint Data"),
    ArrayType("Indexed")]
uint32 ParameterDataSintMin[];

[Description(
    "The upper range for sint Data"),
    ArrayType("Indexed")]
uint32 ParameterDataSintMax[];

[Description(
    "The lower range for real Data"),
    ArrayType("Indexed")]
real32 ParameterDataRealMin[];

[Description(
    "The upper range for real Data"),
    ArrayType("Indexed")]
real32 ParameterDataRealMax[];

[Description(
    "The allowable pattern for string Data using Java ",
    "regular expressions."),
    ArrayType("Indexed")]
string ParameterDataStringPattern[];

[Description(
    "The lower range for datetime Data"),
    ArrayType("Indexed")]
datetime ParameterDataDatetimeMin[];

[Description(
    "The upper range for datetime Data"),
    ArrayType("Indexed")]
datetime ParameterDatetimeMax[];
};

•
•
```

3.6

4 Use cases

The following are the anticipated primary uses cases that will drive development of the functionality. In all cases, the SMI-S Client Program (CP) wishes to launch to the device's element manager to extend capabilities not found within the CP itself.

4.1 Managing Advanced Device Functions

In this example, the CP wishes to perform an advanced function on the storage device that is not within the scope of SMI-S. For example, a hypothetical storage device may have advanced power management functions (e.g., spin down disk drives, shut down fans/CPUs, etc.) that can be enabled and configured.

To discover these capabilities, the CP will communicate with the device CIM agent to determine the existence of LIC points (remote service access points). From these LIC points, the CP will determine the capabilities by traversing to the associated SNIA_LICCapabilities instances. Upon examination of the capabilities, the CP will discover that the element manager is capable of “Manage Power Supply” from the SupportedActions array. In response, the CP will make a button or option available in the appropriate place in the GUI to launch to the power supply LIC point of the element manager.

4.2 Showing Advanced Properties

In this example, the base SMI-S properties for a given object will be augmented with vendor specific properties. For example, a storage volume within an array device may have several properties not found in SMI-S (e.g., the name of the person who created the volume). Using the same discovery approach as above, the CP will determine that the element manager has an internal list volumes capability and can make an option or button available in the GUI to launch to the detailed volume panel.

4.3 Enabling Advanced Wizards

Sometimes, customers want to launch to element managers not because of vendor extended functions or properties; but because of the way the element manager performs a given task. For example, the element manager may provide an advanced provisioning wizard that is tuned to the specific hardware (e.g., creation of storage pools, volumes and mapping/masking might be done all in one panel). These CP can embed these LIC points within their application as needed to augment/enhance their own provisioning implementations.

5 Resources for Development of the functionality

5.1 TWG Resources

Estimate 4-6 Core TWG meetings

5.2 Champion / Author Resources

-
-
-

30-40 hours.

5.3 Projected Vendor Implementations

Identify by name and role, the vendors that are expected to be implementers of the materials covered by the scoping document.

5.3.1 Client Vendors

IBM, Brocade, Aperi, LSI.

-

5.3.2 Provider Vendors

- IBM – Array, Storage Virtualizer
- Brocade – Fabric, Switch
- LSI – Array.

The “Product type” information should be supplied for “common” elements (clauses, packages or subprofiles) to make it clear where these elements would show up in the implementations.

6 Conclusion

Launch-in-Context extends the capabilities of Access Point component profile to allow CIM Providers to publish URL entry points for accessing specialized functions within the device user interface. These specialized functions have a context that is discoverable from the client so that the client can find the best entry point for a given task.