TCG Storage
Security Subsystem Class: Opal

Specification Version 1.00
Revision 3.00

4 February, 2010

Contact: admin@trustedcomputinggroup.org

Copyright © TCG 2010

Disclaimers, Notices, and License Terms

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

Without limitation, TCG disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification and to the implementation of this specification, and TCG disclaims all liability for cost of procurement of substitute goods or services, lost profits, loss of use, loss of data or any incidental, consequential, direct, indirect, or special damages, whether under contract, tort, warranty or otherwise, arising in any way out of use or reliance upon this specification or any information herein.

This document is copyrighted by Trusted Computing Group (TCG), and no license, express or implied, is granted herein other than as follows: You may not copy or reproduce the document or distribute it to others without written permission from TCG, except that you may freely do so for the purposes of (a) examining or implementing TCG specifications or (b) developing, testing, or promoting information technology standards and best practices, so long as you distribute the document with these disclaimers, notices, and license terms.

Contact the Trusted Computing Group at www.trustedcomputinggroup.org for information on specification licensing through membership agreements.

Any marks and brands contained herein are the property of their respective owners.

Change History

Version 1.00	Date	Description
Rev 1.00	27 January, 2009	First publication
Rev 2.00	20 April, 2009	Changed TCG Storage Architecture Core Specification reference and Opal SSC specification numbering
Rev 3.00	18 December, 2009	Corrected the definition of LockingEnabled bit Clarified Revert when Manufactured-Inactive

TABLE OF CONTENTS

Tables

1 Introduction

1.1 Document Purpose

The Storage Workgroup specifications provide a comprehensive architecture for putting Storage Devices under policy control as determined by the trusted platform host, the capabilities of the Storage Device to conform with the policies of the trusted platform, and the lifecycle state of the Storage Device as a Trusted Peripheral.

1.2 Scope and Intended Audience

This specification defines the Opal Security Subsystem Class (SSC). Any SD that claims OPAL SSC compatibility SHALL conform to this specification.

The intended audience for this specification is both trusted Storage Device manufacturers and developers that want to use these Storage Devices in their systems.

1.3 Key Words

Key words are used to signify SSC requirements.

The Key Words "**SHALL**", "**SHALL NOT**", "**SHOULD**," and "**MAY**" are used in this document. These words are a subset of the RFC 2119 key words used by TCG, and have been chosen since they map to key words used in T10/T13 specifications. These key words are to be interpreted as described in [1].

In addition to the above key words, the following are also used in this document to describe the requirements of particular features, including tables, methods, and usages thereof.

- **Mandatory (M):** When a feature is Mandatory, the feature SHALL be implemented. A Compliance test SHALL validate that the feature is operational.
- **Optional (O):** When a feature is Optional, the feature MAY be implemented. If implemented, a Compliance test SHALL validate that the feature is operational.
- **Excluded (X):** When a feature is Excluded, the feature SHALL NOT be implemented. A Compliance test SHALL validate that the feature is not operational.
- **Not Required (N)** When a feature is Not Required, the feature MAY be implemented. No Compliance test is required.

1.4 Document References

[1]. IETF RFC 2119, 1997, "Key words for use in RFCs to Indicate Requirement Levels"

[2]. Trusted Computing Group (TCG), "TCG Storage Architecture Core Specification", Version 2.00

[3]. NIST, FIPS-197, 2001, "Advanced Encryption Standard (AES)"

[4]. [INCITS T10/1731-D], "Information technology - SCSI Primary Commands - 4 (SPC-4)"

[5]. [ANSI INCITS 452-2008], "Information technology - AT Attachment 8 - ATA/ATAPI Command Set (ATA8-ACS)"

[6]. Trusted Computing Group (TCG), "TCG Storage Storage Interface Interactions Specification", Version 1.00

1.5 Document Precedence

In the event of conflicting information in this specification and other documents, the precedence for requirements is:

1. This specification
2. Storage Interface Interactions Specification [6]
3. TCG Storage Architecture Core Specification [2]

1.6 SSC Terminology

This section provides special definitions that are not defined in the Core Specification.

Table 1 Opal SSC Terminology

Term	Definition
Manufactured SP	A Manufactured SP is an SP that was create and preconfigured during the SD manufacturing process
N/A	Not Applicable.
Original Factory State	The original state of an SP when it was created in manufacturing, including its table data, access control settings, and life cycle state. Each Manufactured SP has its own Original Factory State. Original Factory State applies to Manufactured SPs only.
Vendor Unique (VU)	These values are unique to each SD manufacturer. Typically VU is used in table cells.
MM MM	The LSBs of a User Authority object’s UID (hexadecimal) as well as the corresponding C_PIN credential object’s UID (hexadecimal)
NN NN	The LSBs of a Locking object’s UID (hexadecimal) as well as the corresponding K_AES_128/K_AES_256 object’s UID (hexadecimal)
XX XX	The LSBs of an Admin Authority object’s UID (hexadecimal) as well as the corresponding C_PIN credential object’s UID (hexadecimal)

1.7 Legend

The following legend defines SP table cell coloring coding. This color coding is informative only. The table cell content is normative.

Table 2 SP Table Legend

Table Cell Legend	R-W	Value	Access Control	Comment
Arial-Narrow	Read-only	Opal SSC specified	Fixed	• Cell content is Read-Only. • Access control is fixed. • Value is specified by the Opal SSC
Arial Narrow bold-under	Read-only	VU	Fixed	• Cell content is Read-Only. • Access Control is fixed. • Values are Vendor Unique (VU). A minimum or maximum value may be specified.
Arial-Narrow	Not Defined	(N)	Not Defined	• Cell content content is (N). • Access control is not defined. • Any text in table cell is informative only. • A `Get` MAY omit this column from the method response.
Arial Narrow bold-under	Write	Preconfigured, user personalizable	Preconfigured, user personalizable	• Cell content is writable. • Access control is personalizable • `Get` Access Control is not described by this color coding
Arial-Narrow	Write	Preconfigured, user personalizable	Fixed	• Cell content is writable. • Access control is fixed. • `Get` Access Control is not described by this color coding

2 Opal SSC Overview

2.1 Opal SSC Use Cases and Threats

Begin Informative Content

The Opal SSC is an implementation profile for Storage Devices built to:

- Protect the confidentiality of stored user data against unauthorized access once it leaves the owner's control (involving a power cycle and subsequent deauthentication)
- Enable interoperability between multiple SD vendors

An Opal SSC compliant SD:

- Facilitates feature discoverability
- Provides some user definable features (e.g. access control, locking ranges, user passwords, etc.)
- Supports Opal SSC unique behaviors (e.g. communication, table management)

This specification addresses a limited set of use cases. They are:

- Deploy Storage Device & Take Ownership: the Storage Device is integrated into its target system and ownership transferred by setting or changing the Storage Device's owner credential.
- Activate or Enroll Storage Device: LBA ranges are configured and data encryption and access control credentials (re)generated and/or set on the Storage Device. Access control is configured for LBA range unlocking .
- Lock & Unlock Storage Device: unlocking of one or more LBA ranges by the host and locking of those ranges under host control via either an explicit lock or implicit lock triggered by a reset event. MBR shadowing provides a mechanism to boot into a secure pre-boot authentication environment to handle device unlocking.
- Repurpose & End-of-Life: erasure of data within one or more LBA ranges and reset of locking credential(s) for Storage Device repurposing or decommissioning.

End Informative Content

2.2 Security Providers (SPs)

An Opal SSC compliant SD SHALL support at least two Security Providers (SPs):

1) Admin SP
2) Locking SP

The Locking SP MAY be created by the SD manufacturer.

2.3 Interface Communication Protocol

An Opal SSC compliant SD SHALL implement the synchronous communications protocol as defined in Section 3.3.4.

This communication protocol operates based upon configuration information defined by:

1) The values reported via Level 0 Discovery (Section 3.1.1)
2) The combination of the host's communication properties and the TPer's communication properties (see Properties Method Section 4.1.1.1)

2.4 Cryptographic Features

An Opal SSC compliant SD SHALL implement Full Disk Encryption for all host accessible user data stored on media. AES-128 or AES-256 SHALL be supported (see [3]).

2.5 Authentication

An Opal SSC compliant SD SHALL support password authorities and authentication.

2.6 Table Management

This specification defines the mandatory tables and mandatory/optional table rows delivered by the SD manufacturer. The creation or deletion of tables after manufacturing is outside the scope of this specification. The creation or deletion of table rows post-manufacturing is outside the scope of this specification.

2.7 Access Control & Personalization

Initial access control policies are preconfigured at SD manufacturing time on manufacturer created SPs. An Opal SSC compliant SD SHALL support personalization of certain Access Control Elements of the Locking SP.

2.8 Issuance

The Locking SP MAY be present in the SD when the SD leaves the manufacturer. The issuance of SPs is outside the scope of this specification.

2.9 SSC Discovery

Refer to [2] for details (see section 3.1.1).

3 Opal SSC Features

3.1 Security Protocol 1 Support

3.1.1 Level 0 Discovery (M)

Refer to [2] for more details.

An Opal SSC compliant SD SHALL return the following Level 0 response:

- Level 0 Discovery Header
- TPer Feature Descriptor
- Locking Feature Descriptor
- Opal SSC Feature Descriptor

3.1.1.1 Level 0 Discovery Header

Table 3 Level 0 Discovery Header

Bit **Byte**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	(MSB)	Length of Parameter Data						
1								
2								
3								(LSB)
4	(MSB)	Data structure revision						
5								
6								
7								(LSB)
8	(MSB)	Reserved						
…								
15								(LSB)
16	(MSB)	Vendor Specific						
…								
47								(LSB)

• Length of parameter data	= VU
• Data structure revision	= 0x00000001 or any version that supports the defined features in this SSC
• Vendor Specific	= VU

3.1.1.2 TPer Feature (Feature Code = 0x0001)

Table 4 Level 0 Discovery - TPer Feature Descriptor

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Byte								
0	(MSB)	Feature Code						
1								(LSB)
2	Version				Reserved			
3	Length							
4	Reserved	ComID Mgmt Supported	Reserved	Streaming Supported	Buffer Mgmt Supported	ACK/NAK Supported	Async Supported	Sync Supported
5 - 15	Reserved							

- Feature Code = 0x0001
- Version = 0x1 or any version that supports the defined features in this SSC
- Length = 0x0C
- ComID Mgmt Supported = VU
- Streaming Supported = 1
- Buffer Mgmt Supported = VU
- ACK/NACK Supported = VU
- Async Supported = VU
- Sync Supported = 1

3.1.1.3 Locking Feature (Feature Code = 0x0002)

** = the present current state of the respective feature

Table 5 Level 0 Discovery - Locking Feature Descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	Feature Code						
1								(LSB)
2	Version				Reserved			
3	Length							
4	Reserved		MBR Done	MBR Enabled	Media Encryption	Locked	Locking Enabled	Locking Supported
5 - 15	Reserved							

- Feature Code = 0x0002
- Version = 0x1 or any version that supports the defined features in this SSC
- Length = 0x0C
- MBR Done = **
- MBR Enabled = **
- Media Encryption = 1
- Locked = **
- Locking Enabled = See 3.1.1.3.1
- Locking Supported = 1

3.1.1.3.1 LockingEnabled Definition

The definition of the LockingEnabled bit is changed from [2] as follows:

The LockingEnabled bit SHALL be set to one if an SP that incorporates the Locking template is any state other than Nonexistent or Manufactured-Inactive; otherwise the LockingEnabled bit SHALL be set to zero.

3.1.1.4 Opal SSC Feature (Feature Code = 0x0200)

Table 6 Level 0 Discovery - Opal SSC Feature Descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	Feature Code						
1								(LSB)
2	Version				Reserved			
3	Length							
4	(MSB)	Base ComID						
5								(LSB)
6	(MSB)	Number of ComIDs						
7								(LSB)
8	Reserved for future common SSC parameters							Range Crossing
9 - 19	Reserved for future common SSC parameters							

• Feature Code	= 0x0200
• Version	= 0x1 or any version that supports the defined features in this SSC
• Length	= 0x10
• Base ComID	= VU
• Number of ComIDs	= 0x0001 (minimum value)
• Range Crossing	= VU[1]

Note 1: Range Crossing Values:

- 0 = The SD supports commands addressing consecutive LBAs in more than one LBA range if all the LBA ranges addressed are unlocked. See Section 4.3.5
- 1 = The SD terminates commands addressing consecutive LBAs in more than one LBA range. See Section 4.3.5

3.2 Security Protocol 2 Support

3.2.1 ComID Management

ComID management support is reported in Level 0 Discovery. Statically allocated ComIDs are also discoverable via the Level 0 Discovery response.

3.2.2 Stack Protocol Reset (O)

An Opal SSC compliant SD MAY support the Stack Protocol Reset command. Refer to [2] for details.

3.3 Communications

3.3.1 Communication Properties

The TPer SHALL support the minimum communication buffer size as defined in Section 4.1.1.1. For each ComID, the physical buffer size SHALL be reported to the host via the `Properties` method.

The TPer SHALL terminate any IF-SEND command whose transfer length is greater than the reported MaxComPacketSize size for the corresponding ComID. For details, reference "Invalid Transfer Length parameter on IF-SEND" in [6].

Data generated in response to methods contained within an IF-SEND command payload subpacket (including the required ComPacket / Packet / Subpacket overhead data) SHALL fit entirely within the response buffer. If the method response and its associated protocol overhead do not fit completely within the response buffer, the TPer

1) SHALL terminate processing of the IF-SEND command payload,
2) SHALL NOT return any part of the method response if the Sync Protocol is being used, and
3) SHALL return an empty response list with a TCG status code of RESPONSE_OVERFLOW in that method's response status list.

3.3.2 Supported Security Protocols

The TPer SHALL support:

- IF-RECV commands with a Security Protocol values of 0x00, 0x01.
- IF-SEND commands with a Security Protocol values of 0x01.
- IF-RECV commands with a Security Protocol values of 0x02 when Protocol Stack Reset is supported
- IF-SEND commands with a Security Protocol values of 0x02 when Protocol Stack Reset is supported

3.3.3 ComIDs

For the purpose of communication using Security Protocol 0x01, the TPer SHALL:

- support at least one statically allocated ComID for Synchronous Protocol communication.
- have the ComID Extension values = 0x0000 for all statically allocated ComIDs.
- keep all statically allocated ComIDs in the Active state.

When the TPer receives an IF-SEND or IF-RECV with an inactive or unsupported ComID, the TPer SHALL either:

- terminate the command as defined in [6] with “Other Invalid Command parameter”, or
- follow the requirements defined in [2] for “Inactive or Unsupported ComID parameter on IF-SEND” or “Inactive or Unsupported ComID parameter on IF-RECV”.

3.3.4 Synchronous Protocol

The TPer SHALL support the Synchronous Protocol. Refer to [2] for details.

3.3.4.1 Payload Encoding

3.3.4.1.1 Stream Encoding Modifications

The TPer SHALL support tokens listed in Table 7. If an unsupported token is encountered, the TPer SHALL treat this as a streaming protocol violation and return an error per the definition in section 3.3.4.1.3.

Table 7 Supported Tokens

Acronym	Meaning
	Tiny atom
	Short atom
	Medium atom
	Long atom
SL	Start List
EL	End List
SN	Start Name
EN	End Name
CALL	Call
EOD	End of Data
EOS	End of session
ST	Start transaction
ET	End of transaction
MT	Empty atom

The TPer SHALL support tha above token atoms with the B bit set to 0 or 1 and the S bit set to 0.

3.3.4.1.2 TCG Packets

Within a single IF-SEND/IF-RECV command, the TPer SHALL support a ComPacket containing one Packet, which contains one Subpacket. The Host MAY discover TPer support of capabilites beyond this requirement in the parameters returned in response to a `Properties` method.

The TPer MAY ignore Credit Control Subpackets sent by the host. The host MAY discover TPer support of Credit Management with Level 0 Discovery. For more details refer to Section 3.1.1 Level 0 Discovery (M)

The TPer MAY ignore the AckType and Acknowledgement fields in the Packet header on commands from the host and set these fields to zero in its responses to the host. The host MAY discover TPer support of the TCG packet acknowledgement/retry mechanism with Level 0 Discovery. For more details refer to Section 3.1.1 Level 0 Discovery (M)

The TPer MAY ignore packet sequence numbering and not enforce any sequencing behavior. Refer to [2] for details on discovery of packet sequence numbering support.

3.3.4.1.3 Payload Error Response

The TPer SHALL respond according to the following rules if it encounters a streaming protocol violation:

- If the error is on Session Manager or is such that the TPer cannot resolve a valid session ID from the payload (i.e. errors in the ComPacket header or Packet header), then the TPer SHALL discard the payload and immediately transition to the “Awaiting IF-SEND” state.
- If the error occurs after the TPer has resolved the session ID, then the TPer SHALL abort the session and MAY prepare a `CloseSession` method for retrieval by the host.

3.3.5 Storage Device Resets

3.3.5.1 Interface Resets

Interface resets that generate TCG reset events are defined in [6].

Interface initiated TCG reset events SHALL result in:

1. All open sessions SHALL be aborted;
2. All uncommitted transactions SHALL be aborted;
3. All pending session startup activities SHALL be aborted;
4. All TCG command and response buffers SHALL be invalidated;
5. All related method processing SHALL be aborted;
6. For each ComID, the state of the synchronous protocol stack SHALL transition to “Awaiting IF-SEND” state;
7. No notification of these events SHALL be sent to the host.

3.3.6 Protocol Stack Reset Commands (O)

An IF-SEND containing a Protocol Stack Reset Command MAY be supported.

Refer to [2] for details.

4 Opal SSC-compliant Functions and SPs

4.1 Session Manager

4.1.1 Methods

4.1.1.1 Properties (M)

An Opal compliant SD SHALL support the `Properties` method. The requirements for support of the various TPer and Host properties, and the requirements for their values, are shown in Table 8.

Table 8 Properties Requirements

Property Name	TPer Property Requirements and Values Reported	Host Property Requirements and Values Accepted
MaxComPacketSize	(M) 2048 minimum	(M) Initial Assumption: 2048 Minimum allowed: 2048 Maximum allowed: VU
MaxResponseComPacketSize	(M) 2048 minimum	(N) Although this is a legal host property, there is no requirement for the TPer to use it. The TPer MAY ignore this host property and not list it in the HostProperties result of the `Properties` method response.
MaxPacketSize	(M) 2028 minimum	(M) Initial Assumption: 2028 Minimum allowed: 2028 Maximum allowed: VU
MaxIndTokenSize	(M) 1992 minimum	(M) Initial Assumption: 1992 Minimum allowed: 1992 Maximum allowed: VU
MaxPackets	(M) 1 minimum	(M) Initial Assumption: 1 Minimum allowed: 1 Maximum allowed: VU
MaxSubpackets	(M) 1 minimum	(M) Initial Assumption: 1 Minimum allowed: 1 Maximum allowed: VU
MaxMethods	(M) 1 minimum	(M) Initial Assumption: 1 Minimum allowed: 1 Maximum allowed: VU
MaxSessions	(M) 1 minimum	N/A – not a host property
MaxAuthentications	(M) 2 minimum	N/A – not a host property
MaxTransactionLimit	(M) 1 minimum	N/A – not a host property
DefSessionTimeout	(M) VU	N/A – not a host property

4.1.1.2 StartSession (M)

An Opal-compliant SD SHALL support the following parameters for the `StartSession` method:

- HostSessionID
- SPID
- Write = support for “True" is (M), support for "False" is (N)
- HostChallenge
- HostSigningAuthority

4.1.1.3 SyncSession (M)

An Opal-compliant SD SHALL support the following parameters for the `SyncSession` method:

- HostSessionID
- SPSessionID

4.1.1.4 CloseSession (O)

An Opal-Compliant SD MAY support the `CloseSession` method.

4.2 Admin SP

The Admin SP includes the Base Template and the Admin Template.

4.2.1 Base Template Tables

All tables included in the following subsections are mandatory.

4.2.1.1 SPInfo (M)

Table 9 Admin SP - SPInfo Table Preconfiguration

UID	SPID	Name	Size	SizeInUse	SPSessionTimeout	Enabled
00 00 00 02 00 00 00 01	00 00 02 05 00 00 00 01	"Admin"				T

4.2.1.2 SPTemplates (M)

*ST1 = this version number or any version number that complies with this SSC.

Table 10 Admin SP - SPTemplates Table Preconfiguration

UID	TemplateID	Name	Version
00 00 00 03 00 00 00 01	00 00 02 04 00 00 00 01	"Base"	00 00 00 02 *ST1
00 00 00 03 00 00 00 02	00 00 02 04 00 00 00 02	"Admin"	00 00 00 02 *ST1

4.2.1.3 Table (M)

Table 11 Admin SP - Table Table Preconfiguration

UID	Name	CommonName	TemplateID	Kind	Column	NumColumns	Rows	RowsFree	RowBytes	LastID	MinSize	MaxSize
00 00 00 01 00 00 00 01	"Table"			Object								

UID	Name	CommonName	TemplateID	Kind	Column	NumColumns	Rows	RowsFree	RowBytes	LastID	MinSize	MaxSize
00 00 00 01 00 00 00 02	“SPInfo”			Object								
00 00 00 01 00 00 00 03	“SPTemplates”			Object								
00 00 00 01 00 00 00 06	"MethodID"			Object								
00 00 00 01 00 00 00 07	"AccessControl"			Object								
00 00 00 01 00 00 00 08	"ACE"			Object								
00 00 00 01 00 00 00 09	"Authority"			Object								
00 00 00 01 00 00 00 0B	"C_PIN"			Object								
00 00 00 01 00 00 02 01	"TPerInfo"			Object								
00 00 00 01 00 00 02 04	"Template"			Object								
00 00 00 01 00 00 02 05	"SP"			Object								

4.2.1.4 MethodID (M)

The following table contains Optional rows as designated by (O).
*MT1 = refer to section 5.2.1 for details on the requirements for supporting `Activate`.
.

Table 12 Admin SP - MethodID Table Preconfiguration

UID	Name	CommonName	TemplateID
00 00 00 06 00 00 00 08	"Next"		
00 00 00 06 00 00 00 0D	"GetACL"		
00 00 00 06 00 00 00 16	"Get"		
00 00 00 06 00 00 00 17	"Set"		
00 00 00 06 00 00 02 02 (O)	"Revert"		
00 00 00 06 00 00 02 03 *MT1	"Activate"		

TCG Copyright 2010

4.2.1.5 AccessControl (M)

The following table contains Optional rows identified by (O)

*AC1 = TT TT TT TT is a shorthand for the LSBs of the Table object UIDs

*AC2 = TT TT TT TT is a shorthand for the LSBs of the SPTemplates object UIDs

*AC3 = TT TT TT TT is a shorthand for the LSBs of the MethodID object UIDs

*AC4 = TT TT TT TT is a shorthand for the LSBs of the ACE object UIDs

*AC5 = TT TT TT TT is a shorthand for the LSBs of the Authority object UIDs

*AC6 = TT TT TT TT is a shorthand for the LSBs of the Template object UIDs

*AC7 = TT TT TT TT is a shorthand for the LSBs of the SP object UIDs

*AC8 = refer to section 5.2.1 for details on the requirements for supporting `Activate`

Notes:

- The `InvokingID`, `MethodID` and `GetACLACL` columns are a special case. Although they are marked as Read-Only with fixed access control, the access control for invocation of the `Get` method is (N).
- The `ACL` column is readable only via the `GetACL` method.

Table 13 Admin SP - AccessControl Table Preconfiguration

Table association - Informative text	UID	InvokingID	InvokingID Name - Iinformative text	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
Table																
		00 00 00 01 00 00 00 00	Table	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 01 TT TT TT TT *AC1	TableObj	Get		ACE_Anybody				ACE_Anybody						
SPInfo																
		00 00 00 02 00 00 00 01	SPInfoObj	Get		ACE_Anybody				ACE_Anybody						

Table association - Informative text	UID	InvokingID	InvokingID Name - Iinformative text	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
SPTemplates																
		00 00 00 03 00 00 00 00	SPTemplates	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 03 TT TT TT TT *AC2	SPTemplatesObj	Get		ACE_Anybody				ACE_Anybody						
MethodID																
		00 00 00 06 00 00 00 00	MethodID	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 06 TT TT TT TT *AC3	MethodIDObj	Get		ACE_Anybody				ACE_Anybody						
ACE																
		00 00 00 08 00 00 00 00	ACE	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 08 TT TT TT TT *AC4	ACEObj	Get		ACE_Anybody				ACE_Anybody						
Authority																
		00 00 00 09 00 00 00 00	Authority	Next		ACE_Anybody				ACE_Anybody						

Table association - Informative text	UID	InvokingID	InvokingID Name - Iinformative text	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 09 TT TT TT TT *AC5	AuthorityObj	Get		ACE_Anybody				ACE_Anybody						
		00 00 00 09 00 00 00 03	Makers	Set		ACE_Makers_Set_Enabled				ACE_Anybody						
C_PIN																
		00 00 00 0B 00 00 00 00	C_PIN	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 0B 00 00 00 01	C_PIN_SID	Get		ACE_C_PIN_SID_Get_NOPIN				ACE_Anybody						
		00 00 00 0B 00 00 00 01	C_PIN_SID	Set		ACE_C_PIN_SID_Set_PIN				ACE_Anybody						

Table association - Informative text	UID	InvokingID	InvokingID Name - Iinformative text	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 0B 00 00 84 02	C_PIN_MSID	Get		ACE_C_PIN_MSID_Get_PIN				ACE_Anybody						
TPerInfo																
		00 00 02 01 00 03 00 01	TPerInfoObj	Get		ACE_Anybody				ACE_Anybody						
Template																
		00 00 02 04 00 00 00 00	Template	Next		ACE_Anybody				ACE_Anybody						
		00 00 02 04 TT TT TT TT *AC6	TemplateObj	Get		ACE_Anybody				ACE_Anybody						
SP																
		00 00 02 05 00 00 00 00	SP	Next		ACE_Anybody				ACE_Anybody						
		00 00 02 05 TT TT TT TT *AC7	SPObj	Get		ACE_Anybody				ACE_Anybody						

Table association - Informative text	UID	InvokingID	InvokingID Name - Iinformative text	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
(O)		00 00 02 05 TT TT TT TT *AC7	SPObj	Revert		ACE_SP_SID				ACE_Anybody						
*AC8		00 00 02 05 TT TT TT TT *AC7	SPObj	Activate		ACE_SP_SID				ACE_Anybody						

4.2.1.6 ACE (M)

The following table contains Optional rows designated with (O).

*ACE1 = This row is (M) if the TPer supports either `Activate` or `Revert`, and (N) otherwise.

Table 14 Admin SP - ACE Table Preconfiguration

Table Assocuiation - Informative text	UID	Name	CommonName	BooleanExpr	Columns
BaseACEs					
	00 00 00 08 00 00 00 01	"ACE_Anybody"		Anybody	All
	00 00 00 08 00 00 00 02	"ACE_Admin"		Admins	All
Authority					
	00 00 00 08 00 03 00 01	"ACE_Makers_Set_Enabled"		SID	Enabled
C_PIN					
	00 00 00 08 00 00 8C 02	"ACE_C_PIN_SID_Get_NOPIN"		Admins OR SID	UID, CharSet, TryLimit, Tries, Persistence
	00 00 00 08 00 00 8C 03	"ACE_C_PIN_SID_Set_PIN"		SID	PIN
	00 00 00 08 00 00 8C 04	"ACE_C_PIN_MSID_Get_PIN"		Anybody	UID, PIN
SP					

Table Assocuiation - Informative text	UID	Name	CommonName	BooleanExpr	Columns
*ACE1	00 00 00 08 00 03 00 02	"ACE_SP_SID"		SID	All

4.2.1.7 Authority (M)

Table 15 Admin SP - Authority Table Preconfiguration

UID	Name	CommonName	IsClass	Class	Enabled	Secure	HashAndSign	PresentCertificate	Operation	Credential	ResponseSign	ResponseExch	ClockStart	ClockEnd	Limit	Uses	Log	LogTo
00 00 00 09 00 00 00 01	"Anybody"		F	Null	T	None	None	F	None	Null	Null	Null						
00 00 00 09 00 00 00 02	"Admins"		T	Null	T	None	None	F	None	Null	Null	Null						
00 00 00 09 00 00 00 03	"Makers"		T	Null	T	None	None	F	None	Null	Null	Null						
00 00 00 09 00 00 00 06	"SID"		F	Null	T	None	None	F	Password	C_PIN_SID	Null	Null						

4.2.1.8 C_PIN (M)

Table 16 Admin SP - C_PIN Table Preconfiguration

UID	Name	CommonName	PIN	CharSet	TryLimit	Tries	Persistence
00 00 00 0B 00 00 00 01	"C_PIN_SID"		MSID	Null	VU	VU	FALSE
00 00 00 0B 00 00 84 02	"C_PIN_MSID"		MSID				

The PIN column value of C_PIN_SID is set to the PIN column value of C_PIN_MSID in OFS

4.2.2 Base Template Methods

Refer to Section 4.2.1.4 for supported Base template methods.

4.2.3 Admin Template Tables

4.2.3.1 TPerInfo (M)

*TP1 = this version or any version that supports the defined features in this SSC.

*TP2 = The SSC column is a list of names and SHALL have “Opal” as one of the list elements.

Table 17 Admin SP - TPerInfo Table Preconfiguration

UID	Bytes	GUDID	Generation	Firmware Version	ProtocolVersion	SpaceForIssuance	SSC
00 00 02 01 00 03 00 01					1 *TP1		[“Opal”] *TP2

4.2.3.2 Template (M)

The following table contains an Optional row as designated by (O).

*T1 = refer to section 5.1 for Interface Control details.

Table 18 Admin SP - Template Table Preconfiguration

UID	Name	Revision Number	Instances	MaxInstances
00 00 02 04 00 00 00 01	"Base"	1	VU	VU
00 00 02 04 00 00 00 02	"Admin"	1	1	1
00 00 02 04 00 00 00 06	"Locking"	1	1	1
00 00 02 04 00 00 00 07 (O) *T1	"Interface Control"	1	1	1

4.2.3.3 SP (M)

*SP1 = This row only exists in the Admin SP's OFS when the Locking SP is created by the manufacturer.

Table 19 Admin SP - SP Table Preconfiguration

UID	Name	ORG	EffectiveAuth	DateOfIssue	Bytes	LifeCycle	Frozen
00 00 02 05 00 00 00 01	"Admin"					Manufactured	FALSE
00 00 02 05 00 00 00 02 *SP1	"Locking"					Manufactured-Inactive OR Manufactured	FALSE

4.2.4 Admin Template Methods

Refer to 4.2.1.4 for Admin SP supported methods.

4.3 Locking SP

4.3.1 Base Template Tables

All tables defined with (M) in section titles are mandatory.

4.3.1.1 SPInfo (M)

Table 20 Locking SP - SPInfo Table Preconfiguration

UID	SPID	Name	Size	SizeInUse	SPSessionTimeout	Enabled
00 00 00 02 00 00 00 01	00 00 02 05 00 00 00 02	"Locking"				T

4.3.1.2 SPTemplates (M)

*SP1 = This version number or any number that supports the defined features in this SSC
*SP2 = refer to section 5.1 for Interface Control details

Table 21 Locking SP - SPTemplates Table Preconfiguration

UID	TemplateID	Name	Version
00 00 00 03 00 00 00 01	00 00 02 04 00 00 00 01	"Base"	00 00 00 02 *SP1
00 00 00 03 00 00 00 02	00 00 02 04 00 00 00 06	"Locking"	00 00 00 02 *SP1
00 00 00 03 00 00 00 03 (O) *SP2	00 00 02 04 00 00 00 07	"Interface Control"	00 00 00 02 *SP1

4.3.1.3 Table (M)

The following table contains Optional rows designated with (O).

TT1 = only one of the two K_AES table is required

*TT2 = refer to section 5.1 for Interface Control details

Table 22 Locking SP - Table Table Preconfiguration

UID	Name	CommonName	TemplateID	Kind	Column	NumColumns	Rows	RowsFree	RowBytes	LastID	MinSize	MaxSize
00 00 00 01 00 00 00 01	"Table"			Object								
00 00 00 01 00 00 00 02	"SPInfo"			Object								

UID	Name	CommonName	TemplateID	Kind	Column	NumColumns	Rows	RowsFree	RowBytes	LastID	MinSize	MaxSize
00 00 00 01 00 00 00 03	"SPTemplates"			Object								
00 00 00 01 00 00 00 06	"MethodID"			Object								
00 00 00 01 00 00 00 07	"AccessControl"			Object								
00 00 00 01 00 00 00 08	"ACE"			Object								
00 00 00 01 00 00 00 09	"Authority"			Object								
00 00 00 01 00 00 00 0B	"C_PIN"			Object								
00 00 00 01 00 00 08 01	"LockingInfo"			Object								
00 00 00 01 00 00 08 02	"Locking"			Object								
00 00 00 01 00 00 08 03	"MBRControl"			Object								
00 00 00 01 00 00 08 04	"MBR"			Byte			0x08000000 min					
00 00 00 01 00 00 08 05 *TT1	"K_AES_128"			Object								
00 00 00 01 00 00 08 06 *TT1	"K_AES_256"			Object								
00 00 00 01 00 00 0C 01 (O) *TT2	"RestrictedCommands"			Object								
00 00 00 01 00 00 10 01	"DataStore"			Byte			0x00000400 min					

4.3.1.4 Type (N)

The Type table is (N) by Opal. The following types as defined by [2] SHALL meet the following requirements:
:

- The "boolean_ACE" type (00000005 000040E) SHALL include the OR boolean operator.
- The "AC_element" type (00000005 00000801) SHALL support at least 9 entries (4 User authorities and 1 Admin authority.)

4.3.1.5 MethodID (M)

*MT1 = refer to section 5.2.3 for details on the requirements for supporting RevertSP.

Table 23 Locking SP - MethodID Table Preconfiguration

UID	Name	CommonName	TemplateID
00 00 00 06 00 00 00 08	"Next"		
00 00 00 06 00 00 00 0D	"GetACL"		
00 00 00 06 00 00 00 10	"GenKey"		
00 00 00 06 00 00 00 11 *MT1	"RevertSP"		
00 00 00 06 00 00 00 16	"Get"		
00 00 00 06 00 00 00 17	"Set"		

4.3.1.6 AccessControl (M)

The following table contains Optional rows designated with (O).

*AC1 = refer to section 5.2.3 for details on the requirements for supporting RevertSP

*AC2 = TT TT TT TT is a shorthand for the LSBs of the Table object UIDs

*AC3 = TT TT TT TT is a shorthand for the LSBs of the SPTemplates object UIDs

*AC4 = TT TT TT TT is a shorthand for the LSBs of the MethodID object UIDs

*AC5 = TT TT TT TT is a shorthand for the LSBs of the ACE object UIDs

*AC6 = only K_AES_128 or K_AES_256 related rows mandatory

*AC7 = TT TT TT TT is a shorthand for the LSB of the Authority object UIDs

*AC8 = TT TT TT TT is a shorthand for the LSBs of the RestrictedCommands object UIDs

Notes:

- The `InvokingID`, `MethodID` and `GetACLACL` columns are a special case. Although they are marked as Read-Only with fixed access control, the access control for invocation of the `Get` method is (N).
- The `ACL` column is readable only via the `GetACL` method.

Table 24 Locking SP - AccessControl Table Preconfiguration

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
SP																

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
*AC1		00 00 00 00 00 00 00 01	ThisSP	RevertSP		ACE_Admin				ACE_Anybody						
Table																
		00 00 00 01 00 00 00 00	Table	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 01 TT TT TT TT *AC2	TableObj	Get		ACE_Anybody				ACE_Anybody						
SPInfo																
		00 00 00 02 00 00 00 01	SPInfoObj	Get		ACE_Anybody				ACE_Anybody						
SPTemplates																
		00 00 00 03 00 00 00 00	SPTemplates	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 03 TT TT TT TT *AC3	SPTemplatesObj	Get		ACE_Anybody				ACE_Anybody						
MethodID																
		00 00 00 06 00 00 00 00	MethodID	Next		ACE_Anybody				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 06 TT TT TT TT *AC4	MethodIDObj	Get		ACE_Anybody				ACE_Anybody						
ACE																
		00 00 00 08 00 00 00 00	ACE	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 08 TT TT TT TT *AC5	ACEObj	Get		ACE_ACE_Get_All				ACE_Anybody						
		00 00 00 08 00 03 80 00	ACE_ACE_Get_All	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 90 00	ACE_Authority_Get_All	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
*AC6		00 00 00 08 00 03 B0 00	ACE_K_AES_128_GlobalRange_GenKey	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
*AC6		00 00 00 08 00 03 B0 01	ACE_K_AES_128_Range1_GenKey	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
*AC6		00 00 00 08 00 03 B0 00 (+NNNN)	ACE_K_AES_128_RangeNNNN _GenKey	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
*AC6		00 00 00 08 00 03 B8 00	ACE_K_AES_256_GlobalRange_GenKey	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
*AC6		00 00 00 08 00 03 B8 01	ACE_K_AES_256_Range1_GenKey	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
*AC6		00 00 00 08 00 03 B8 00 (+NNNN)	ACE_K_AES_256_RangeNNNN _GenKey	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 D0 00	ACE_Locking_GlobalRange_Get_RangeStartToActiveKey	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 08 00 03 D0 01	ACE_Locking_Range1_Get_RangeStartToActiveKey	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 D0 00 (+NNNN)	ACE_Locking_RangeNNNN_Get_RangeStartToActiveKey	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 E0 00	ACE_Locking_GlobalRange_Set_RdLocked	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 08 00 03 E0 01	ACE_Locking_Range1_Set_RdLocked	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 E0 00 (+NNNN)	ACE_Locking_RangeNNNN_Set_RdLocked	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 E8 00	ACE_Locking_GlobalRange_Set_WrLocked	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 08 00 03 E8 01	ACE_Locking_Range1_Set_WrLocked	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 E8 00 (+NNNN)	ACE_Locking_RangeNNNN_Set_WrLocked	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 F8 01	ACE_MBRControl_Set_Done	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 08 00 03 FC 00	ACE_DataStore_Get_All	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 FC 01	ACE_DataStore_Set_All	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
Authority																
		00 00 00 09 00 00 00 00	Authority	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 09 TT TT TT TT *AC7	AuthorityObj	Get		ACE_Authority_Get_All				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 09 00 01 00 02	Admin2	Set		ACE_Authority_Set_Enabled				ACE_Anybody						
		00 00 00 09 00 01 00 00 (+XX XX)	AdminXXXX	Set		ACE_Authority_Set_Enabled				ACE_Anybody						
		00 00 00 09 00 03 00 01	User1	Set		ACE_Authority_Set_Enabled				ACE_Anybody						
		00 00 00 09 00 03 00 00 (+MMMM)	UserMMMM	Set		ACE_Authority_Set_Enabled				ACE_Anybody						
C_PIN																
		00 00 00 0B 00 00 00 00	C_PIN	Next		ACE_Anybody				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 0B 00 01 00 01	C_PIN_Admin1	Get		ACE_C_PIN_Admins_Get_All_NOPIN				ACE_Anybody						
		00 00 00 0B 00 01 00 00 (+ XX XX)	C_PIN_AdminXXXX	Get		ACE_C_PIN_Admins_Get_All_NOPIN				ACE_Anybody						
		00 00 00 0B 00 03 00 01	C_PIN_User1	Get		ACE_C_PIN_Admins_Get_All_NOPIN				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 0B 00 03 00 00 (+MM MM)	C_PIN_UserMMMM	Get		ACE_C_PIN_Admins_Get_All_NOPIN				ACE_Anybody						
		00 00 00 0B 00 01 00 01	C_PIN_Admin1	Set		ACE_C_PIN_Admins_Set_PIN				ACE_Anybody						
		00 00 00 0B 00 01 00 00 (+XX XX)	C_PIN_AdminXXXX	Set		ACE_C_PIN_Admins_Set_PIN				ACE_Anybody						
		00 00 00 0B 00 03 00 01	C_PIN_User1	Set		ACE_C_PIN_User1_Set_PIN				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 0B 00 03 00 00 (+MM MM)	C_PIN_UserMMMM	Set		ACE_C_PIN_UserMMMM_Set_PIN				ACE_Anybody						
LockingInfo																
		00 00 08 01 00 00 00 01	LockingInfoObj	Get		ACE_Anybody				ACE_Anybody						
Locking																
		00 00 08 02 00 00 00 00	Locking	Next		ACE_Anybody				ACE_Anybody						
		00 00 08 02 00 00 00 01	Locking_GlobalRange	Get		ACE_Locking_GlobalRange_Get_ RangeStartToActiveKey				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 08 02 00 03 00 01	Locking_Range1	Get		ACE_Locking_Range1_Get_ RangeStartToActiveKey				ACE_Anybody						
		00 00 08 02 00 03 00 00 (+NN NN)	Locking_RangeNNNN	Get		ACE_Locking_RangeNNNN_Get_ RangeStartToActiveKey				ACE_Anybody						
		00 00 08 02 00 00 00 01	Locking_GlobalRange	Set		ACE_Locking_GlblRng_Admins_Set, ACE_Locking_GlobalRange_Set_RdLocked, ACE_Locking_GlobalRange_Set_WrLocked				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 08 02 00 03 00 01	Locking_Range1	Set		ACE_Locking_Admins_RangeStartToLocked, ACE_Locking_Range1_Set_RdLocked, ACE_Locking_Range1_Set_WrLocked				ACE_Anybody						
		00 00 08 02 00 03 00 00 (+NN NN)	Locking_RangeNNNN	Set		ACE_Locking_Admins_RangeStartToLocked, ACE_Locking_RangeNNNN_Set_RdLocked, ACE_Locking_RangeNNNN_Set_WrLocked				ACE_Anybody						
MBRControl																
		00 00 08 03 00 00 00 01	MBRControlObj	Get		ACE_Anybody				ACE_Anybody						
		00 00 08 03 00 00 00 01	MBRControlObj	Set		ACE_MBRControl_Admins_Set, ACE_MBRControl_Set_Done				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
MBR																
		00 00 08 04 00 00 00 00	MBR	Get		ACE_Anybody				ACE_Anybody						
		00 00 08 04 00 00 00 00	MBR	Set		ACE_Admin				ACE_Anybody						
K_AES_128																
		00 00 08 05 00 00 00 01	K_AES_128_GlobalRange_Key	Get		ACE_K_AES_Mode				ACE_Anybody						
		00 00 08 05 00 03 00 01	K_AES_ 128 _Range1_Key	Get		ACE_K_AES_Mode				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 08 05 00 03 00 00 (+NN NN)	K_AES_ 128 _RangeNNNN_Key	Get		ACE_K_AES_Mode				ACE_Anybody						
		00 00 08 05 00 00 00 01	K_AES_128_GlobalRange_Key	GenKey		ACE_K_AES_ 128 _GlobalRange_GenKey				ACE_Anybody						
		00 00 08 05 00 03 00 01	K_AES_ 128 _Range1_Key	GenKey		ACE_K_AES_ 128 _Range1_GenKey				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 08 05 00 03 00 00 (+NN NN)	K_AES_ 128 _RangeNNNN_Key	GenKey		ACE_K_AES_ 128 _RangeNNNN_GenKey				ACE_Anybody						
K_AES_256																
		00 00 08 06 00 00 00 01	K_AES_256_GlobalRange_Key	Get		ACE_K_AES_Mode				ACE_Anybody						
		00 00 08 06 00 03 00 01	K_AES_ 256_Range1_Key	Get		ACE_K_AES_Mode				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 08 06 00 03 00 00 (+NN NN)	K_AES_ 256_RangeNNNN_Key	Get		ACE_K_AES_Mode				ACE_Anybody						
		00 00 08 06 00 00 00 01	K_AES_256_GlobalRange_Key	GenKey		ACE_K_AES_ 256_GlobalRange_GenKey				ACE_Anybody						
		00 00 08 06 00 03 00 01	K_AES_ 256_Range1_Key	GenKey		ACE_K_AES_ 256_Range1_GenKey				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 08 06 00 03 00 00 (+NN NN)	K_AES_ 256_RangeNNNN_Key	GenKey		ACE_K_AES_ 256_RangeNNNN_GenKey				ACE_Anybody						
RestrictedCommands																
(O)		00 00 0C 01 00 00 00 00	RestrictedCommands	Next		ACE_Anybody				ACE_Anybody						
(O)		00 00 0C 01 TT TT TT TT *AC8	RestrictedCommandsObj	Get		ACE_Anybody				ACE_Anybody						
DataStore																
		00 00 10 01 00 00 00 00	DataStore	Get		ACE_DataStore_Get_All				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 10 01 00 00 00 00	DataStore	Set		ACE_DataStore_Set_All				ACE_Anybody						

4.3.1.7 ACE (M)

The following table contains Optional rows designated with (O).

Table 25 Locking SP - ACE Table Preconfiguration

Table Association -Informative Column	UID	Name	CommonName	BooleanExpr	Columns
Base ACEs					
	00 00 00 08 00 00 00 01	"ACE_Anybody"		Anybody	All
	00 00 00 08 00 00 00 02	"ACE_Admin"		Admins	All
ACE					
	00 00 00 08 00 03 80 00	"ACE_ACE_Get_All"		Admins	All
	00 00 00 08 00 03 80 01	"ACE_ACE_Set_BooleanExpression"		Admins	BooleanExpr
Authority					
	00 00 00 08 00 03 90 00	"ACE_Authority_Get_All"		Admins	All
	00 00 00 08 00 03 90 01	"ACE_Authority_Set_Enabled"		Admins	Enabled
C_PIN					
	00 00 00 08 00 03 A0 00	"ACE_C_PIN_Admins_Get_All_NOPIN"		Admins	UID, CharSet, TryLimit, Tries, Persistence
	00 00 00 08 00 03 A0 01	"ACE_C_PIN_Admins_Set_PIN"		Admins	PIN
	00 00 00 08 00 03 A8 01	"ACE_C_PIN_User1_Set_PIN"		Admins OR User1	PIN
(O)	00 00 00 08 00 03 A8 00	"ACE_C_PIN_UserMMMM_Set_PIN"		Admins OR	PIN

Table Association -Informative Column	UID	Name	CommonName	BooleanExpr	Columns
	(+MMMM)			UserMMMM	
K_AES					
	00 00 00 08 00 03 BF FF	"ACE_K_AES_Mode"		Anybody	Mode
K_AES_128					
	00 00 00 08 00 03 B0 00	"ACE_K_AES_128_GlobalRange_ GenKey"		Admins	All
	00 00 00 08 00 03 B0 01	"ACE_K_AES_128_Range1_ GenKey"		Admins	All
(O)	00 00 00 08 00 03 B0 00 (+NNNN)	"ACE_K_AES_128_RangeNNNN_ GenKey"		Admins	All
K_AES_256					
	00 00 00 08 00 03 B8 00	"ACE_K_AES_256_GlobalRange_ GenKey"		Admins	All
	00 00 00 08 00 03 B8 01	"ACE_K_AES_256_Range1_ GenKey"		Admins	All
	00 00 00 08 00 03 B8 00 (+NNNN)	"ACE_K_AES_256_RangeNNNN_ GenKey"		Admins	All
Locking					
	00 00 00 08 00 03 D0 00	"ACE_Locking_GlobalRange_Get_ RangeStartToActiveKey"		Admins	RangeStart, RangeLength, ReadLockEnabled, WriteLockEnabled, ReadLocked, WriteLocked, LockOnReset, ActiveKey
	00 00 00 08 00 03 D0 01	"ACE_Locking_Range1_Get_ RangeStartToActiveKey"		Admins	RangeStart, RangeLength, ReadLockEnabled, WriteLockEnabled, ReadLocked, WriteLocked, LockOnReset, ActiveKey
	00 00 00 08 00 03 D0 00 (+NNNN)	"ACE_Locking_RangeNNNN_Get_ RangeStartToActiveKey"		Admins	RangeStart, RangeLength, ReadLockEnabled, WriteLockEnabled, ReadLocked, WriteLocked, LockOnReset, ActiveKey
	00 00 00 08 00 03 E0 00	"ACE_Locking_GlobalRange_Set_RdLocked"		Admins	ReadLocked
	00 00 00 08 00 03 E0 01	"ACE_Locking_Range1_Set_RdLocked"		Admins	ReadLocked
	00 00 00 08 00 03 E0 00 (+NNNN)	"ACE_Locking_RangeNNNN_Set_RdLocked"		Admins	ReadLocked

Table Association -Informative Column	UID	Name	CommonName	BooleanExpr	Columns
	00 00 00 08 00 03 E8 00	"ACE_Locking_GlobalRange_Set_WrLocked"		Admins	WriteLocked
	00 00 00 08 00 03 E8 01	"ACE_Locking_Range1_Set_WrLocked"		Admins	WriteLocked
	00 00 00 08 00 03 E8 00 (+NNNN)	"ACE_Locking_RangeNNNN_Set_WrLocked"		Admins	WriteLocked
	00 00 00 08 00 03 F0 00	"ACE_Locking_GlblRng_Admins_Set"		Admins	ReadLockEnabled, WriteLockEnabled, ReadLocked, WriteLocked
	00 00 00 08 00 03 F0 01	"ACE_Locking_Admins_RangeStartToLocked"		Admins	RangeStart, RangeLength, ReadLockEnabled, WriteLockEnabled, ReadLocked, WriteLocked
MBRControl					
	00 00 00 08 00 03 F8 00	"ACE_MBRControl_Admins_Set"		Admins	Enable, Done
	00 00 00 08 00 03 F8 01	"ACE_MBRControl_Set_Done"		Admins	Done
DataStore					
	00 00 00 08 00 03 FC 00	"ACE_DataStore_Get_All"		Admins	All
	00 00 00 08 00 03 FC 01	"ACE_DataStore_Set_All"		Admins	All

4.3.1.8 Authority (M)

The following table contains Optional rows designated with (O).
Notes:

1. Admin1 is required; any additional Admin authorities are (O).
2. User1 through User4 SHALL be implemented.

Table 26 Locking SP - Authority Table Preconfiguration

UID	Name	CommonName	IsClass	Class	Enabled	Secure	HashAndSign	PresentCertificate	Operation	Credential	ResponseSign	ResponseExch	ClockStart	ClockEnd	Limit	Uses	Log	LogTo
00 00 00 09 00 00 00 01	"Anybody"		F	Null	T	None	None	F	None	Null	Null	Null						
00 00 00 09 00 00 00 02	"Admins"		T	Null	T	None	None	F	None	Null	Null	Null						
00 00 00 09 00 01 00 01	"Admin1"		F	Admins	T	None	None	F	Password	C_PIN_Admin1	Null	Null						
00 00 00 09 00 01 00 00 (+XX XX)[1] (O)	"AdminXXXX"		F	Admins	F													
00 00 00 09 00 03 00 00	"Users"		T	Null	T	None	None	F	None	Null	Null	Null						
00 00 00 09 00 03 00 01	"User1"		F	Users	F	None	None	F	Password	C_PIN_User1	Null	Null						
00 00 00 09 00 03 00 00 (+MM MM)[2] (O)	"UserMMMM"		F	Users	F	None	None	F	Password	C_PIN_UserMMMM	Null	Null						

4.3.1.9 C_PIN (M)

The following table includes Optional rows designated with (O)

Notes:

1. If the Locking SP's original life cycle state is Manufactured-Inactive, see Section 5.2.1.1 for the initial value of C_PIN_Admin1.PIN. If the Locking SP's original life cycle state is Manufactured, then the initial value of C_PIN_Admin1.PIN is the same as the Admin SP's C_PIN_MSID.PIN value.

Table 27 Locking SP - C_PIN Table Preconfiguration

UID	Name	CommonName	PIN	CharSet	TryLimit	Tries	Persistence
00 00 00 0B 00 01 00 01	"C_PIN_Admin1"		SID or MSID[1]	Null	0	0	FALSE
00 00 00 0B 00 01 00 00 (+XX XX) (O)	"C_PIN_AdminXXXX"		“”	Null	0	0	FALSE
00 00 00 0B 00 03 00 01	"C_PIN_User1"		“”	Null	0	0	FALSE
00 00 00 0B 00 03 00 00 (+MM MM) (O)	"C_PIN_UserMMMM"		“”	Null	0	0	FALSE

4.3.2 Base Template Methods

Refer to section 4.3.1.5 for supported methods.

4.3.3 Locking Template Tables

4.3.3.1 LockingInfo (M)

Note:

1. The MaxRanges column specifies the number of supported ranges and SHALL have a minimum of 4 ranges.

Table 28 Locking SP - LockingInfo Table Preconfiguration

UID	Name	Version	EncryptSupport	MaxRanges	MaxReEncryptions	KeysAvailableCfg
00 00 08 01 00 00 00 01			Media Encryption	4[1]		

4.3.3.2 Locking (M)

The following table contains Optional rows designated with (O).
*LT1 = The ActiveKey can be a K_AES_128 object reference (UID) or a K_AES_256 object reference (UID)

Table 29 Locking SP - Locking Table Preconfiguration

UID	Name	CommonName	RangeStart	RangeLength	ReadLockEnabled	WriteLockEnabled	ReadLocked	WriteLocked	LockOnReset	ActiveKey	NextKey	ReEncryptState	ReEncyptRequest	AdvKeyMode	VerifyMode	ContOnReset	LastReEncryptLBA	LastReEncState	GeneralStatus
00 00 08 02 00 00 00 01	"Locking_GlobalRange"		0	0	F	F	F	F	Power Cycle	K_AES_128[256]_GlobalRange_Key *LT1									

UID	Name	CommonName	RangeStart	RangeLength	ReadLockEnabled	WriteLockEnabled	ReadLocked	WriteLocked	LockOnReset	ActiveKey	NextKey	ReEncryptState	ReEncyptRequest	AdvKeyMode	VerifyMode	ContOnReset	LastReEncryptLBA	LastReEncState	GeneralStatus
00 00 08 02 00 03 00 01	"Locking_Range1"		0	0	F	F	F	F	Power Cycle	K_AES_128[256]_Range1_Key *LT1									
00 00 08 02 00 03 NN NN	"Locking_RangeNNNN"		0	0	F	F	F	F	Power Cycle	K_AES_128[256]_RangeNNNN_Key *LT1									

4.3.3.3 MBRControl (M)

Table 30 Locking SP - MBRControl Table Preconfiguration

UID	Enable	Done	DoneOnReset
00 00 08 03 00 00 00 01	False	False	Power Cycle

4.3.3.4 MBR (M)

The MBR minimum size SHALL be 128 MB (0x08000000).

The initial contents of the MBR table SHALL be vendor unique.

4.3.3.5 K_AES_128 or K_AES_256 (M)

At least one of the following two tables SHALL be supported.

The following table contains Optional rows designated with (O).
*K1 = indirectly writable using the `GenKey` Method.

Table 31 Locking SP - K_AES_128 Table Preconfiguration

UID	Name	CommonName	Key	Mode
00 00 08 05 00 00 00 01	"K_AES_128_GlobalRange_Key"		VU *K1	VU
00 00 08 05 00 03 00 01	"K_AES_128_Range1_Key"		VU *K1	VU
00 00 08 05 00 03 NN NN (O)	"K_AES_128_RangeNNNN_Key"		VU *K1	VU

Table 32 Locking SP - K_AES_256 Table Preconfiguration

UID	Name	CommonName	Key	Mode
00 00 08 06 00 00 00 01	"K_AES_256_GlobalRange_Key"		VU *K1	VU
00 00 08 06 00 03 00 01	"K_AES_256_Range1_Key"		VU *K1	VU
00 00 08 06 00 03 NN NN (O)	"K_AES_256_RangeNNNN_Key"		VU *K1	VU

4.3.4 Locking Template Methods

Refer to Section 4.3.1.5 for supported methods.

4.3.5 SD Read/Write Data Command Locking Behavior

The SD SHALL terminate with a "Data Protection Error" as defined in [6]:

- Read commands that address consecutive LBAs in one or more locked LBA ranges. Locked range is ReadLockEnabled=True and ReadLocked=True.
- Write commands that address consecutive LBAs in one or more LBA ranges for which WriteLockEnabled=True and WriteLocked=True.

If the storage device receives a read or write command that spans multiple LBA ranges and the LBA ranges are not locked, the storage device SHALL either:

- Process the data transfer, if Range Crossing = 0 (in Level 0 Discovery Opal SSC Feature, see 3.1.1)

  OR
- Terminate the command with “Other Invalid Command Parameter” as defined in [6],
  if Range Crossing = 1 (in Level 0 Discovery Opal SSC Feature, see 3.1.1)

The SD SHALL abort the following commands:

- For SCSI [4] commands:
  - READ LONG(10)
  - READ LONG(16)
  - WRITE LONG(10), (WR_UNCOR = 0)
  - WRITE LONG(16), (WR_UNCOR = 0)
- For ATA [5] devices:
  - READ LONG (obsolete)
  - WRITE LONG (obsolete)
  - SCT READ LONG
  - SCT WRITE LONG

4.3.6 Interface Control Template Tables

See Section 5.1 for further details on the Interface Control Template

4.3.6.1 RestrictedCommands (O)

Table 33 RestrictedCommands Table Preconfiguration

UID	Next	CommandMask	ComandFilter	Allowed	AllowedTrueOnReset	AllowedFalseOnReset
VU	VU	VU	VU	VU	VU	VU

4.3.7 Non Template Tables

4.3.7.1 DataStore (M)

The DataStore is a byte table. It can be used by the host for generic secure data storage. The DataStore table SHALL be at least 1 KB in size (the Table table object that represents the DataStore table SHALL have a Rows column value of at least 0x00000400). The access control for modification or retrieval of data in the table initially requires a member of the Admins class authority. These access control settings are personalizable. Initial DataStore content value is VU.

TCG Copyright 2010

5 Appendix – SSC Specific Features

5.1 Interface Control Template

5.1.1 Overview

The Interface Control template enables TPer control over selected interface commands. The benefit is the reduction of undesired side effects. These commands MAY change the runtime or permanent configuration of the Storage Device as a whole. As such, it is in the spirit of being a trusted peripheral that the use of such commands be restricted.

Some examples of interface command operations that MAY be restricted are:

- Downloading new firmware
- Changing the maximum LBA accessible
- Enabling or disabling Storage Device features
- Forcing read errors
- Changing power-on default settings
- Changing Storage Device timing parameters
- Reading and writing raw data
- Formatting the Storage Device

This template provides facilities to restrict unauthorized use of certain commands via the host interface.

The template UID SHALL be 00 00 02 04 00 00 00 07

5.1.2 Data Structures

5.1.2.1 RestrictedCommands (Object Table)

The `RestrictedCommands` table contains rules about host interface command restrictions.

The `RestrictedCommands` table usage model is defined below. The number of actual commands are VU. See Section 5.1.4 for table row examples.

The table SHALL contain at least one required row. The required row has the following attributes:

- The UID of the required row is the UID of the `RestrictedCommands` table, plus one
- SHALL NOT match any command
- SHALL NOT be deletable.

Table 34 RestrictedCommands Table Description

Column	Type	Description
UID	uid	The UID of this row
Next	uid	The UID of the next row to be processed. Exactly one row SHALL have a Next column value of Null, which marks the last row to be processed. See examples in Section 5.1.4
CommandMask	{bytes}	Interface-dependent binary mask of interface command and parameters. Refer to Section 5.1.4 Examples
CommandFilter	{bytes}	Interface-dependent binary filter of interface command and parameters**.** Refer to Section 5.1.4 Examples

Column	Type	Description
Allowed	boolean	If this flag is True, then execution of the described command is not restricted; otherwise, the command is not allowed.
AllowedTrueOnReset	reset_types	Reset types that force the Allowed column to True
AllowedFalseOnReset	reset_types	Reset types that force the Allowed column to False

Table 35 CommandMask and CommandFilter (ATA)

ByteOffset	Length	ATA Command Parameter
0	1	Command
1	1	Device
2	2	Features
4	2	Count
6	6	LBA
12	Vendor specific	Optional data transferred from the host

Table 36 CommandMask and CommandFilter (ATAPI)

ByteOffset	Length	ATA Command Parameter
0	1	Command
1	1	Device
2	1	Features
3	1	Count
4	3	LBA
7	12 or 16	Packet (Command)
19 or 23	VU	Optional data transferred from the host

Table 37 CommandMask and CommandFilter (SCSI)

ByteOffset	Length	SCSI Field
0	VU	CDB
VU	VU	Optional data transferred from the host

5.1.3 Descriptions

A TPer MAY support at most one SP that incorporates the Interface Control Template.

When a TCG reset that is listed in the `AllowedTrueOnReset` column occurs, the TPer SHALL immediately set the value of the `Allowed` column to True. When a TCG reset that is listed in the `AllowedFalseOnReset` column occurs, the TPer SHALL immediately set the value of the `Allowed` column to False. A TCG reset type

SHALL NOT be listed in both the AllowedTrueOnReset and the AllowedFalseOnReset columns. If a TCG reset occurs that is not in either AllowedTrueOnReset or the AllowedFalseOnReset columns, the value of the Allowed column SHALL NOT be changed.

Rows SHALL always be processed starting with the required row, and proceeding in the order specified by the Next column. The command parameters are to be bit-AND'd with the CommandMask column, and the result compared to the CommandFilter column. If the comparison matches, the value of the Allowed column determines if the command is restricted or not. This process is performed for all rows from the beginning of the table until the first match is made. If no match is made, then this facility does not restrict the processing of the command.

If the comparison matches and the value of the Allowed column is False, the SD SHALL terminate the command with a "Data Protection Error" as defined in [7].

See Figure 1 for an example of using the rules in the RestrictedCommands table.

Figure 1 Command Processing Example

```
 // Parse the interface command against the RestrictedCommands table
row=First                       // Always start at the beginning of the table
restrict = false
matched = false
while ( (matched==false) AND (restrict==false) AND (row != NULL) )
{
        If  (CommandFilter[row] ==
                ( (incoming command and parameters) bitwise-AND (CommandMask[row] ))  )
        {
                matched = true
                restrict = Allowed[row]
        }
        else      row = Next
}

if (restrict == true)
  then terminate the command
  else allow the command to proceed to the next level of command processing
```

5.1.3.1 Interface Control Template-Specific Life Cycle State Descriptions/Exceptions

A Manufactured SP instantiated with the Interface Control Template has the following characteristics based on the current life cycle state of that SP:

- **Manufactured Inactive**: restrictions SHALL NOT be applied to the interface commands.
- **Manufactured:** restrictions SHALL be applied to the interface commands.

5.1.4 Examples

These tables show some example commands for which control of execution MAY be desirable.

Table 38 Example RestrictedCommands Table (ATA)

UID	Next	CommandMask	ComandFilter	Allowed	AllowedTrueOnReset	AllowedFalseOnReset
00 00 0C 01 00 00 00 01	00 00 0C 01 00 00 00 02	00	DO NOT MATCH ANY COMMAND FF	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 02	00 00 0C 01 00 00 00 03	FF 00 0000 0000 000000000000	READ BUFFER E4 00 0000 0000 000000000000	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 03	00 00 0C 01 00 00 00 04	FF 00 0000 0000 000000000000	WRITE BUFFER E8 00 0000 0000 000000000000	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 04	00 00 0C 01 00 00 00 05	FF 00 00FF 0000 000000000000	SET FEATURES enable SATA features EF 00 0010 0000 000000000000	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 05	00 00 0C 01 00 00 00 06	FF 00 00FF 0000 000000000000	SET FEATURES disable SATA features EF 00 0090 0000 000000000000	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 06	00 00 0C 01 00 00 00 07	FF 00 0000 0001 000000000000	SET MAX ADDRESS (non-volatile) F9 00 0000 0001 000000000000	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 07	00 00 0C 01 00 00 00 08	FF 00 0000 0001 000000000000	SET MAX ADDRESS EXT (non-volatile) 37 00 0000 0001 000000000000	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 08	00 00 0C 01 00 00 00 09	FF 00 0000 0000 000000000000	WRITE UNCORRECTABLE EXT 45 00 0000 0000 000000000000	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 09	00 00 0C 01 00 00 00 0A	FF 00 0000 0000 000000000000	READ LONG 22 00 0000 0000 000000000000	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 0A	00 00 0C 01 00 00 00 0B	FF 00 0000 0000 000000000000	WRITE LONG 32 00 0000 0000 000000000000	False	(null)	(Power Cycle)

UID	Next	CommandMask	ComandFilter	Allowed	AllowedTrueOnReset	AllowedFalseOnReset
00 00 0C 01 00 00 00 0B	00 00 0C 01 00 00 00 0C	FF 00 00FF 0000 0000000000FF FFFF	SCT READ/WRITE LONG (via SMART WRITE LOG) B0 00 00D6 0000 0000000000E0 0001 <data xfered>	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 0C	00 00 0C 01 00 00 00 0D	FF 00 0000 0000 0000000000FF FFFF	SCT READ/WRITE LONG (via WRITE LOG EXT) 3F 00 0000 0000 0000000000E0 0001 <data xfered>	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 0D	00 00 0C 01 00 00 00 0E	FF 00 0000 0000 0000000000FF FFFF	SCT READ/WRITE LONG (via WRITE LOG DMA EXT) 57 00 0000 0000 0000000000E0 0001 <data xfered>	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 0E	00 00 0C 01 00 00 00 0F	FF 00 00FF 0000 000000000000	SET FEATURES enable PUIS EF 00 0006 0000 000000000000	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 0F	00 00 0C 01 00 00 00 10	FF 00 00FF 0000 000000000000	SET FEATURES disable PUIS EF 00 0086 0000 000000000000	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 10	00 00 0C 01 00 00 00 11	FF 00 00FF 0000 000000000000	SMART DISABLE OPERATIONS B0 00 00D9 0000 000000000000	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 11	00 00 0C 01 00 00 00 12	FF 00 0000 0000 0000000000FF	WRITE LOG DMA EXT (host vendor specific log) 57 00 0000 0000 000000000080 57 00 0000 0000 000000000081 . . . 57 00 0000 0000 00000000009F	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 12	00 00 0C 01 00 00 00 13	FF 00 0000 0000 000000000000	WRITE LOG EXT (host vendor specific log) 3F 00 0000 0000 000000000080 3F 00 0000 0000 000000000081 . . . 3F 00 0000 0000 00000000009F	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 13	00 00 0C 01 00 00 00 14	FF 00 00FF 0000 000000000000	DCO RESTORE B3 00 00C0 0000 000000000000	False	(null)	(Power Cycle)

UID	Next	CommandMask	ComandFilter	Allowed	AllowedTrueOnReset	AllowedFalseOnReset
00 00 0C 01 00 00 00 14	00 00 0C 01 00 00 00 15	FF 00 00FF 0000 000000000000	DCO SET B3 00 00C30000 000000000000	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 15	00 00 0C 01 00 00 00 16	FF 00 0000 0000 000000000000	DOWNLOAD MICROCODE 92 00 0000 0000 000000000000	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 16	00 00 0C 01 00 00 00 17	FF 00 0000 0000 000000000000	READ LONG W/O RETRIES 23 00 0000 0000 000000000000	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 17	00 00 00 00 00 00 00 00	FF 00 0000 0000 000000000000	WRITE LONG W/O RETRIES 33 00 0000 0000 000000000000	False	(null)	(Power Cycle)

Table 39 Example RestrictedCommands Table (ATAPI)

UID	Next	CommandMask	Comand Filter	Allowed	AllowedTrueOnReset	AllowedFalseOnReset
00 00 0C 01 00 00 00 01	00 00 0C 01 00 00 00 02	00	DO NOT MATCH ANY COMMAND FF	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 02	00 00 0C 01 00 00 00 03	FF 00 00FF 0000 000000000000	DCO RESTORE B3 00 00C0 0000 000000000000	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 03	00 00 0C 01 00 00 00 04	FF 00 00FF 0000 000000000000	DCO SET B3 00 00C30000 000000000000	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 04	00 00 0C 01 00 00 00 05	FF 00 00FF 0000 000000000000	SET FEATURES enable PUIS EF 00 0006 0000 000000000000	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 05	00 00 0C 01 00 00 00 06	FF 00 00FF 0000 000000000000	SET FEATURES disable PUIS EF 00 0086 0000 000000000000	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 06	00 00 0C 01 00 00 00 07	FF 00 00 00 000000 FF 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF	PACKET MODE SELECT (6) (allow SP=0 for mode page 1Ah) A0 00 00 00 000000 15 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 <HDR> 1A <PAGE CODE>	True	(Power Cycle)	(null)
00 00 0C 01 00 00 00 07	00 00 0C 01 00 00 00 08	FF 00 00 00 000000 FF 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF	PACKET MODE SELECT (6) (restrict SP=1 for mode page 1Ah) A0 00 00 00 000000 15 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 <HDR> 1A <PAGE CODE>	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 08	00 00 0C 01 00 00 00 09	FF 00 00 00 000000 FF FF 00 00 00 00 00 00 00 00 00 00	PACKET READ BUFFER (10) (allow mode 1Ch) A0 00 00 00 000000 3C 1C 00 00 00 00 00 00 00 00 00 00	True	(Power Cycle)	(null)

UID	Next	CommandMask	Comand Filter	Allowed	AllowedTrueOnReset	AllowedFalseOnReset
00 00 0C 01 00 00 00 09	00 00 0C 01 00 00 00 0A	FF 00 00 00 000000 FF FF 00 00 00 00 00 00 00 00 00 00	PACKET READ BUFFER (10) (restrict all other modes) A0 00 00 00 000000 3C FF 00 00 00 00 00 00 00 00 00 00	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 0A	00 00 0C 01 00 00 00 0B	FF 00 00 00 000000 FF 00 00 00 00 00 00 00 00 00 00 00	PACKET READ LONG(10) A0 00 00 00 000000 3E 00 00 00 00 00 00 00 00 00 00 00	False	(null)	(Power Cycle)
00 00 0C 01 00 00 00 0B	00 00 0C 01 00 00 00 0C	FF 00 00 00 000000 FF FF 00 00 00 00 00 00 00 00 00 00	PACKET WRITE BUFFER (allow mode 04h) A0 00 00 00 000000 3B 04 00 00 00 00 00 00 00 00 00 00	True	(Power Cycle)	(null)
00 00 0C 01 00 00 00 0D	00 00 0C 01 00 00 00 0E	FF 00 00 00 000000 FF FF 00 00 00 00 00 00 00 00 00 00	PACKET WRITE BUFFER (allow mode 05h) A0 00 00 00 000000 3B 05 00 00 00 00 00 00 00 00 00 00	True	(Power Cycle)	(null)
00 00 0C 01 00 00 00 0E	00 00 0C 01 00 00 00 0F	FF 00 00 00 000000 FF FF 00 00 00 00 00 00 00 00 00 00	PACKET WRITE BUFFER (allow mode 06h) A0 00 00 00 000000 3B 06 00 00 00 00 00 00 00 00 00 00	True	(Power Cycle)	(null)
00 00 0C 01 00 00 00 0F	00 00 0C 01 00 00 00 10	FF 00 00 00 000000 FF FF 00 00 00 00 00 00 00 00 00 00	PACKET WRITE BUFFER (allow mode 07h) A0 00 00 00 000000 3B 07 00 00 00 00 00 00 00 00 00 00	True	(Power Cycle)	(null)
00 00 0C 01 00 00 00 10	00 00 0C 01 00 00 00 11	FF 00 00 00 000000 FF FF 00 00 00 00 00 00 00 00 00 00	PACKET WRITE BUFFER (allow mode 0Eh) A0 00 00 00 000000 3B 0E 00 00 00 00 00 00 00 00 00 00	True	(Power Cycle)	(null)
00 00 0C 01 00 00 00 11	00 00 0C 01 00 00 00 12	FF 00 00 00 000000 FF FF 00 00 00 00 00 00 00 00 00 00	PACKET WRITE BUFFER (allow mode 0Fh) A0 00 00 00 000000 3B 0F 00 00 00 00 00 00 00 00 00 00	True	(Power Cycle)	(null)

UID	Next	CommandMask	Comand Filter	Allowed	AllowedTrueOnReset	AllowedFalseOnReset
00 00 0C 01 00 00 00 12	00 00 00 00 00 00 00 00	FF 00 00 00 000000 FF 00 00 00 00 00 00 00 00 00 00 00	PACKET WRITE LONG(10) A0 00 00 00 000000 3F 00 00 00 00 00 00 00 00 00 00 00	False	(null)	(Power Cycle)

Table 40 Example RestrictedCommands Table (SCSI)

UID	Next	CommandMask	CommandFilter	Allowed	AllowedTrueOnReset	AllowedFalseOnReset
00 00 0C 01 00 00 00 01	00 00 0C 01 00 00 00 02	00	FF	False	(null)	(Power Cycle, HW reset)
00 00 0C 01 00 00 00 02	00 00 0C 01 00 00 00 03	FF 00 00 00 00 00 00 00 00 00	READ LONG(10) 3E 00 00 00 00 00 00 00 00 00	False	(null)	(Power Cycle, HW reset)
00 00 0C 01 00 00 00 03	00 00 0C 01 00 00 00 04	FF 00 00 00 00 00 00 00 00 00	WRITE LONG(10) 3F 00 00 00 00 00 00 00 00 00	False	(null)	(Power Cycle, HW reset)
00 00 0C 01 00 00 00 04	00 00 0C 01 00 00 00 05	FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	READ LONG(16) 9E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	False	(null)	(Power Cycle, HW reset)
00 00 0C 01 00 00 00 05	00 00 0C 01 00 00 00 06	FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	WRITE LONG(16) 9F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	False	(null)	(Power Cycle, HW reset)
00 00 0C 01 00 00 00 06	00 00 0C 01 00 00 00 07	FF 1F 00 00 00 00 00 00 00 00	READ BUFFER (allow mode 1Ch) 3C 1C 00 00 00 00 00 00 00 00	True	(Power Cycle, HW reset)	(null)
00 00 0C 01 00 00 00 07	00 00 0C 01 00 00 00 08	FF 1F 00 00 00 00 00 00 00 00	READ BUFFER (restrict all other modes) 3C FF 00 00 00 00 00 00 00 00	False	(null)	(Power Cycle, HW reset)
00 00 0C 01 00 00 00 08	00 00 0C 01 00 00 00 09	FF 1F 00 00 00 00 00 00 00 00	WRITE BUFFER (allow mode 04h) 3B 04 00 00 00 00 00 00 00 00	True	(Power Cycle, HW reset)	(null)
00 00 0C 01 00 00 00 09	00 00 0C 01 00 00 00 0A	FF 1F 00 00 00 00 00 00 00 00	WRITE BUFFER (allow mode 05h) 3B 05 00 00 00 00 00 00 00 00	True	(Power Cycle, HW reset)	(null)
00 00 0C 01 00 00 00 0A	00 00 0C 01 00 00 00 0B	FF 1F 00 00 00 00 00 00 00 00	WRITE BUFFER (allow mode 06h) 3B 06 00 00 00 00 00 00 00 00	True	(Power Cycle, HW reset)	(null)
00 00 0C 01 00 00 00 0B	00 00 0C 01 00 00 00 0C	FF 1F 00 00 00 00 00 00 00 00	WRITE BUFFER (allow mode 07h) 3B 07 00 00 00 00 00 00 00 00	True	(Power Cycle, HW reset)	(null)
00 00 0C 01 00 00 00 0C	00 00 0C 01 00 00 00 0D	FF 1F 00 00 00 00 00 00 00 00	WRITE BUFFER (allow mode 0Eh) 3B 0E 00 00 00 00 00 00 00 00	True	(Power Cycle, HW reset)	(null)

UID	Next	CommandMask	CommandFilter	Allowed	AllowedTrueOnReset	AllowedFalseOnReset
00 00 0C 01 00 00 00 0D	00 00 0C 01 00 00 00 0E	FF 1F 00 00 00 00 00 00 00 00	WRITE BUFFER (allow mode 0Fh) 3B 0F 00 00 00 00 00 00 00 00	True	(Power Cycle, HW reset)	(null)
00 00 0C 01 00 00 00 0E	00 00 00 00 00 00 00 00	FF 1F 00 00 00 00 00 00 00 00	WRITE BUFFER (restrict all other modes) 3B FF 00 00 00 00 00 00 00 00	False	(null)	(Power Cycle, HW reset)

5.2 Opal SSC-Specific Methods

5.2.1 Activate – Admin Template SP Object Method

Activate is an Opal SSC-specific method for managing the life cycle of SPs created in manufacturing, whose initial life cycle state is "Manufactured-Inactive".

```
SPObjectUID.Activate[ ]
=>
[ ]
```

Activate is an object method that operates on objects in the Admin SP's SP table. The TPer SHALL NOT permit Activate to be invoked on the SP objects of issued SPs.

Invocation of Activate on an SP object that is in the "Manufactured-Inactive" state causes the SP to transition to the "Manufactured" state. Invocation of Activate on an SP in any other life cycle state SHALL complete successfully provided access control is satisfied, and have no effect. The Activate method allows the TPer owner to "turn on" an SP that was created in manufacturing.

This method operates within a Read-Write session to the Admin SP. The SP SHALL be activated immediately after the method returns success if its invocation is not contained within a transaction.

Support for Activate within transactions is (N), and the behavior is out of the scope of this document.

If the Locking SP was created in manufacturing, and its Original Factory State is Manufactured-Inactive (see section 5.3.2), support for Activate on the Locking SP's object in the SP Table is mandatory.

If Activate is invoked on the Locking SP while ATA Security is Enabled (i.e., a User Password is set), the method invocation SHALL fail with a status of FAIL.

The MethodID for Activate SHALL be 00 00 00 06 00 00 02 03.

5.2.1.1 Side effects of Activate

Upon successful activation of an SP that was in the "Manufactured-Inactive" state, the following changes SHALL be made:

- The LifeCycleState column of SP's object in the Admin SP's SP table SHALL change to "Manufactured".
- The current SID PIN (C_PIN_SID) in the Admin SP is copied into the PIN column of Admin1's C_PIN credential (C_PIN_Admin1) in the activated SP. This allows for taking ownership of the SP with a known PIN credential.
- Any TPer functionality affected by the life cycle state of the SP based on the templates incorporated into it is modified as defined in the appropriate Template reference section of the Core Spec, and as defined in the "State transitions for Manufactured SPs" section (section 5.3.2.2) and "State behaviors for Manufactured SPs" section (section 5.3.2.3) of this specification.

5.2.2 Revert – Admin Template SP Object Method

`Revert` is an Opal SSC-specific method for managing the life cycle of SPs created in manufacturing.

```
SPObjectUID.Revert[ ]
=>
[ ]
```

`Revert` is an object method that operates on objects in the Admin SP's `SP` table. The TPer SHALL NOT permit `Revert` to be invoked on the SP objects of issued SPs.

Invoking `Revert` on an SP object causes the SP to revert to its Original Factory State. This method allows the TPer owner (or TPer manufacturer, if access control permits and the Maker authorities are enabled) to remove the SP owner's ownership of the SP and revert the SP to its Original Factory State.

This method operates within a Read-Write session to the Admin SP. The TPer SHALL revert the SP immediately after the method is successfully invoked outside of a transaction. If `Revert` is invoked on the Admin SP's object in the SP table, the TPer SHALL abort the session immediately after reporting status of the method invocation if invoked outside of a transaction. The TPer MAY prepare a `CloseSession` method for retrieval by the host to indicate that the session has been aborted.

Support for `Revert` within transactions is (N), and the behavior is out of the scope of this document.

Support for `Revert` on the Admin SP's object in the `SP` table is optional.

Support for `Revert` on the Locking SP's object in the `SP` Table is optional.

Invocation of `Revert` is permitted on Manufactured SPs that are in any life cycle state. Successful invocation of `Revert` on a Manufactured SP that is in the Manufactured-Inactive life cycle state SHALL have no effect on the SP.

The MethodID for `Revert` SHALL be 00 00 00 06 00 00 02 02.

5.2.2.1.1 Side effects of Revert

Upon successful invocation of the `Revert` method, the following changes SHALL be made:

- The row in the Admin SP's `SP` table that represents this SP SHALL revert to its original factory values.
- The SP itself SHALL revert to its Original Factory State. While reverting to its Original Factory State, the TPer SHALL securely erase all personalization of the SP, and revert the personalized values to their original factory values. The mechanism for secure erasure is implementation-specific. Informative note: Unless already in the Manufactured-Inactive life cycle state, reverting the Locking SP will cause the media encryption keys to be eradicated, which has the side effect of securely erasing all data in the User LBA portion of the SD.
- When Revert is successfully invoked on the SP object for the Admin SP (UID = 00 00 02 05 00 00 00 01), the ***entire TPer*** SHALL revert to its Original Factory State, including all personalization of the Admin SP itself. All issued SPs SHALL be deleted, and all Manufactured SPs SHALL revert to Original Factory State. Manufactured SPs that were in the Manufactured-Inactive life cycle state SHALL be unaffected.
- Any TPer functionality affected by the life cycle state of the SP based on the templates incorporated into it is modified as defined in the appropriate Template reference section of the Core Spec, and as defined in the "State transitions for Manufactured SPs" section (section 5.3.2.2) and "State behaviors for Manufactured SPs" section (section 5.3.2.3) of this specification.

5.2.3 RevertSP – Base Template SP Method

RevertSP is an Opal SSC-specific method for managing the life cycle of an SP, if it was created in manufacturing.

```
ThisSP.RevertSP[ KeepGlobalRangeKey = boolean ]
=>
[ ]
```

RevertSP is an SP method in the Base Template.

Invoking RevertSP on an SP SHALL cause it to revert to its Original Factory State. This method allows the SP owner to relinquish control of the SP and revert the SP to its Original Factory State.

This method operates within a Read-Write session to an SP. The TPer SHALL revert the SP immediately after the method is successfully invoked outside of a transaction. Upon completion of reverting the SP, the TPer SHALL report status of the method invocation if invoked outside of a transaction, and then immediately abort the session. The TPer MAY prepare a CloseSession method for retrieval by the host to indicate that the session has been aborted.

Support for RevertSP within transactions is (N), and the behavior is out of the scope of this document.

If the Locking SP was created in manufacturing, support for RevertSP on the Locking SP is mandatory.

The MethodID for RevertSP SHALL be 00 00 00 06 00 00 00 11.

5.2.3.1 KeepGlobalRangeKey parameter (Locking Template-specific)

The optional **KeepGlobalRangeKey** parameter is a Locking Template-specific optional parameter. This parameter provides a mechanism for the Locking SP to be “turned off” without eradicating the media encryption key for the Global locking range. This allows the TCG management of the SD's locking and media encryption features to be disabled without causing a cryptographic erase of the user data associated with the Global locking range.

When this parameter is present and set to True, the TPer SHALL continue to use the media encryption key associated with the Global locking range after the Locking SP transitions to the “Manufactured-Inactive” state.

The following condition SHALL guarantee that the TPer can comply with the request to keep the Global Range’s media encryption key:

- The Global Range is either Read Unlocked or Write Unlocked at the time of invocation of RevertSP

If the TPer cannot comply with the request to keep the Global Range’s media encryption key, then the method invocation SHALL fail with status FAIL, and the SP SHALL NOT change life cycle states.

If the Locking SP was created in manufacturing, support for the **KeepGlobalRangeKey** parameter is mandatory for the Locking SP.

The parameter number for **KeepGlobalRangeKey** SHALL be 0x060000.

5.2.3.2 Side effects of RevertSP

Upon successful invocation of the RevertSP method, the following changes SHALL be made:

- The SP’s object in the Admin SP’s SP table SHALL revert to its original factory values.
- The SP itself SHALL revert to its Original Factory State. While reverting to its Original Factory State, the TPer SHALL securely erase all personalization of the SP, and revert the personalized values to their original factory values. The mechanism for secure erasure is implementation-specific. The exception to the secure erasure is the value of the Global Range’s media encryption key (K_AES_{128,256}_GlobalRange_Key) in the Locking SP, if the **KeepGlobalRangeKey** parameter is present and set to True. Informative note: Reverting the Locking SP will cause the media encryption keys to be eradicated (except for the GlobalRange key if the **KeepGlobalRangeKey** parameter is

present and set to True), which has the side effect of securely erasing all data in the User LBA portion of the SD.

- Any TPer functionality affected by the life cycle state of the SP based on the templates incorporated into it is modified as defined in the appropriate Template reference section of the Core Spec, and as defined in the “State transitions for Manufactured SPs” section (section 5.3.2.2) and “State behaviors for Manufactured SPs” section (section 5.3.2.3) of this specification.

5.3 Life Cycle

5.3.1 Issued vs. Manufactured SPs

5.3.1.1 Issued SPs

The Core Specification describes the life cycle states for SPs that are created through the issuance process. For Opal SSC-compliant TPers that support issuance, refer to the Core Specification for the life cycle states and life cycle management.

5.3.1.2 Manufactured SPs

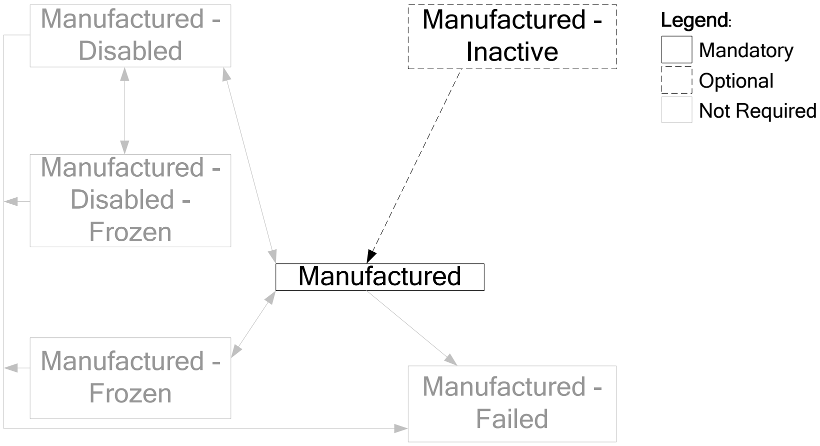
The Core Specification defines the life cycle and life cycle management of Manufactured SPs as implementation-specific.

Opal SSC-compliant SPs that are created in manufacturing (Manufactured SPs) SHALL NOT have implementation-specific life cycle, and SHALL conform to the life cycle defined in section 5.3.2.

5.3.2 Manufactured SP Life Cycle States

The state diagram for Manufactured SPs is shown in Figure 2.

Figure 2 Life Cycle State Diagram for Manufactured SPs



Additional state transitions may exist depending on the states supported by the SD and the SP’s Original Factory State. Invoking `Revert` or `RevertSP` (see sections 5.2.2 and 5.2.3) on the SP will cause the SP to transition back to its Original Factory State.

The Original Factory State of the Admin SP SHALL be Manufactured. The only state that is mandatory for the Admin SP is Manufactured.

If the Locking SP is a Manufactured SP, its Original Factory State SHALL be Manufactured-Inactive or Manufactured.

If the Locking SP is a Manufactured SP, support of the Manufactured state is mandatory and support of the Manufactured-Inactive state is optional for the Locking SP.

The other states in the state diagram are beyond the scope of this document.

5.3.2.1 State definitions for Manufactured SPs

1. **Manufactured-Inactive**: This is the Original Factory State for SPs that are created in manufacturing, where it is not desirable for the functionality of that SP to be active when the TPer is shipped. All templates that exist in an SP that is in the Manufactured-Inactive state SHALL be counted in the

Instances column of the appropriate objects in the Admin SP's Template table. Sessions cannot be opened to SPs in the Manufactured-Inactive state. Only SPs whose Original Factory State was Manufactured-Inactive can return to the Manufactured-Inactive state.

If the Locking SP is a Manufactured SP, support for the Manufactured-Inactive state is optional for the Locking SP.

2. **Manufactured**: This is the standard operational state of a Manufactured SP, and defines the initial required access control settings of an SP based on the Templates incorporated into the SP, prior to personalization.

   The Manufactured state is mandatory for the Admin SP.

   If the Locking SP is a Manufactured SP, support for the Manufactured state is mandatory for the Locking SP.

5.3.2.2 State transitions for Manufactured SPs

The following sections describe the mandatory and optional state transitions for Opal SSC-compliant Manufactured SPs.

For the Admin SP, the only transition for which support is mandatory is "ANY STATE to ORIGINAL FACTORY STATE" (5.3.2.2.2). As the only mandatory state for the Admin SP is Manufactured, the only mandatory transition is from Manufactured to Manufactured with the side effect of reverting the entire TPer to its Original Factory State. See section 5.2.2 for details.

If the Locking SP is a Manufactured SP, support for the "ANY STATE to ORIGINAL FACTORY STATE" transition (5.3.2.2.2) is mandatory. Specifically, support for the transition from Manufactured to either Manufactured-Inactive or Manufactured is mandatory, depending on the Locking SP's Original Factory State. This transition is accomplished via the Revert or RevertSP method (see sections 5.2.2 and 5.2.3).

If the Locking SP's Original Factory State is Manufactured-Inactive, then support for the "Manufactured-Inactive to Manufactured" transition (5.3.2.2.1) is mandatory. This transition is accomplished via the Activate method (see section 5.2).

5.3.2.2.1 Manufactured-Inactive to Manufactured

Triggers:

- The Activate method (see section 5.2) is successfully invoked on the SP's object in the Admin SP's SP table.

Side effects:

- The value in the LifeCycleState column of the SP's object in the Admin SP's SP table changes to Manufactured.
- The current SID PIN (C_PIN_SID) in the Admin SP is copied into the PIN column of Admin1's C_PIN credential (C_PIN_Admin1) in the activated SP. This allows for taking ownership of the SP with a known PIN credential.
- Any functionality enabled by the templates incorporated into the SP becomes active.

When the Locking SP transitions from the Manufactured-Inactive state to the Manufactured state (via invocation of the Activate method), the SD SHALL NOT destroy any user data.

5.3.2.2.2 ANY STATE to ORIGINAL FACTORY STATE

Triggers:

- Revert or RevertSP is successfully invoked on the SP.

Side effects:

- The value in the LifeCycleState column of the SP's object in the Admin SP's SP table changes to the value of the SP's Original Factory State.

- The SP itself reverts to its Original Factory State, as described in the sections 5.2.2 and 5.2.3.
- If the SP's Original Factory State was Manufactured-Inactive, any functionality enabled by the templates incorporated into the SP becomes inactive.

5.3.2.3 State behaviors for Manufactured SPs

5.3.2.3.1 Manufactured-Inactive

Any functionality enabled by the templates incorporated into the SP is inactive in this state. Sessions cannot be opened to SPs in this state.

When the Locking SP is in the Manufactured-Inactive state, the TCG management of the SD's locking and media encryption features SHALL be disabled.

5.3.2.3.2 Manufactured

Behavior of an SP in the Manufactured state is identical to the behavior of an SP in the Issued state, as described by the Core Specification.

When the Locking SP is in the Manufactured state, the TCG management of the SD's locking and media encryption features SHALL be enabled.

5.3.2.4 Locking SP Life Cycle Interactions with the ATA Security Feature Set

The storage device MAY support the ATA Security feature set when the Locking SP is in the Nonexistent state (for TPers that support issuance of the Locking SP) or the Manufactured-Inactive state (for TPers that contain a manufactured Locking SP). In all other life cycle states for the Locking SP, the storage device SHALL report that the ATA Security feature set is "not supported" (IDENTIFY DEVICE, word 82, bit 1 = 0).

When ATA Security is Enabled (i.e., a User Password is set), the TPer SHALL prohibit a Manufactured Locking SP from transitioning out of the Manufactured-Inactive state (see section 5.2)

5.3.3 Type Table Modification

In order to accommodate the additional life cycle states defined in Opal, the `life_cycle_state` type SHALL be defined as follows for Opal:

Table 41 LifeCycle Type Table Modification

UID	Name	Format	Size	Description
00 00 00 05 00 00 04 05	life_cycle_state	Enumeration_Type, 0, 15		Used to represent the current life cycle state. The valid values are: 0 = issued, 1 = issued-disabled, 2 = issued-frozen, 3 = issued-disabled-frozen, 4 = issued-failed, 5-7 = reserved, 8 = manufactured-inactive, 9 = manufactured, 10 = manufactured-disabled, 11 = manufactured-frozen, 12 = manufactured-disabled-frozen, 13 = manufactured-failed, 14-15 = reserved