ULINK_Test Reporter (TR v4.3.6)	[Â©2022 ULINK Technology Inc.] Date : Fri Jul 08 01:02:59 UTC 2022
Application Key	Value 1
Test Suite	ULINK TCG/I1667 - OPAL
Test Version	Script REV 7.5
Device Id	PC601 NVME SK HYNIX 512GB_SI8BQ02611010BQBA
Capacity	512 GB
Tested	805
Passed	805
Failed	0
Warned	
Not tested	323
Passed with Waiver	0
Start Date	8/16/2019
Start Time	3:07:54 PM
End Date	8/16/2019
End Time	3:28:58 PM
Test Time	0:21:03
OS Type	x64
Software Version	7.1.1300
Model Number	PC601 NVME SK HYNIX 512GB
Serial Number	SI8BQ02611010BQBA
Firmware Revision	80000111
Total LBA	1000215216
Device Interface	NVMEDRV+NET
Path	/UTL/Log/Ptc_TCG_PC601NVMeSKhynix512GB_SI8BQ02611010BQBA_01/
CSV Name	Ptc_TCG_PC601NVMeSKhynix512GB_SI8BQ02611010BQBA_01.csv
License By	ULINK
Test Software	
OS Version	
Copyright	
Comment1	null
Comment2	null
Comment3	null
Revert LockingSP	
Start Session with HostChallenge - AdminSP	PASS
Sync Session - AdminSP	PASS
LockingSP.Revert - Request	PASS
LockingSP.Revert - Response	PASS
End Session - Request	PASS
End Session - Response	PASS
** OPAL v1.0 - I1667 Test Cases **	
A0: Identify Device	
(A0-1-1-1-1) Word 48: Identify Device: bit 0 of word 48 shall be set to 1	N/A
(A0-1-1-1-2) Word 119: Identify Device: bit 6 of word 119 = 1 - TPer supports Sense Data Reporting	N/A
A1: Test Trusted Send/Receive cases	
(A1-1-1-1-1) XferLength: Trusted Send with SP=00h; Spcf=ComID; Xfer=00h: Abort	PASS
(A1-1-1-1-1) XferLength: Trusted Send with SP=01h; Spcf=ComID; Xfer=00h: Abort	N/A
(A1-1-1-1-2) XferLength: Trusted Send with SP=01h; Spcf=ComID; Xfer>MaxComPacketSize: Abort	PASS
(A1-1-3-1-3) Spcf: Trusted Send with SP=01h; Spcf=ComID; Xfer=01h NOT in awaiting IF_Send: Abort	PASS
(A1-1-3-1-4) Spcf: Trusted Send with SP=01h; Spcf=Inactive ComID; Xfer=01h in IF_Send: Pass or Abort	PASS
(A1-1-3-1-5) Spcf: Trusted Send with SP=01h; Spcf=Unsupported ComID(0-0FFFh); Xfer=01h: Abort	PASS
(A1-2-1-1-1) XferLength: Trusted Receive with SP=01h; Spcf=ComID; Xfer=00h: Abort	N/A
(A1-2-3-2-2) Spcf: Trusted Receive with SP=01h; Spcf=Inactive ComID; Xfer=01h: Pass or Abort	PASS
(A1-2-3-2-3) Spcf: Trusted Receive with SP=01h; Spcf=Unsupported ComID(0-0FFFh); Xfer=01h: Abort	PASS
A2: Test Protocol ID = 0 related cases	
(A2-1-1-1-2) Spcf=0 DataXfer: TCG-Receive with SP=00h; Spcf=00h; Xfer=00h: Pass	PASS
(A2-1-1-1-1) Spcf=0 DataXfer: TCG-Receive with SP=00h; Spcf=00h; Xfer=01h: Pass	PASS
(A2-1-2-1-2) Spcf=0 DataContent: TCG-Receive with SP=00h; Spcf=00h; Xfer=01h: SP List-Byte6-7 >= 02h	PASS
(A2-1-2-1-3(1)) Spcf=0 DataContent: TCG-Receive with SP=00h; Spcf=00h; Xfer=01h: SP list-Byte8 = 00h	PASS
(A2-1-2-1-3(2)) Spcf=0 DataContent: TCG-Receive with SP=00h; Spcf=00h; Xfer=01h: SP list-Byte9 = 01h	PASS
(A2-1-2-1-3(3)) Spcf=0 DataContent: TCG-Receive with SP=00h; Spcf=00h; Xfer=01h: SP list-Byte10 = 02h(if supported) or 00h	PASS
(A2-2-1-1-2) Spcf=1 DataXfer: TCG-Receive with SP=00h; Spcf=01h; Xfer=00h: Pass	PASS
(A2-2-1-1-1) Spcf=1 DataXfer: TCG-Receive with SP=00h; Spcf=01h; Xfer=01h: Pass	PASS
(A2-2-2-1-2) Spcf=1 DataXfer: TCG-Receive with SP=00h; Spcf=01h; Xfer=01h: Certificate-Byte2-3 = 00h or a value	PASS
A3: Test Level 0 Discovery Protocol	
(A3-1-1-1-1) TCG-Receive with SP=01h; Spcf=01h; Xfer=00h: Abort	N/A
(A3-1-1-1-2) TCG-Receive with SP=01h; Spcf=01h; Xfer=01h: Pass	PASS
A4: Test Synchronous Communication Protocol	
(A4-1-1-1-1) IF_Send: TPer in awaiting IF_Send state after Power-on reset - IF_Send with SP=01h; Spcf=ComID; Xfer=01h: pass	PASS
(A4-1-1-1-3) IF_Send: TPer in awaiting IF_Send state - IF_Send with SP=01h; Spcf=ComID; Xfer=01h: pass	PASS
(A4-1-3-1-1) IF_Send: TPer in awaiting IF_Recv state - IF_Send with SP=01h; Spcf=ComID; Xfer=01h: abort	PASS
(A4-2-1-2-1) IF_Recv: TPer in awaiting IF_Send state - IF_Recv(Level0 discovery) with SP=01h; Spcf=01h; Xfer=01h: pass	PASS
(A4-2-1-2-3) IF_Recv: TPer in awaiting IF_Send state - IF_Recv with SP=01h; Spcf=ComID; Xfer=01h: no data returned	PASS
(A4-2-3-2-1) IF_Recv: TPer in awaiting IF_Recv state - IF_Recv(Level0 discovery) with SP=01h; Spcf=01h; Xfer=01h: pass	PASS

(A4-2-3-2-2) IF_Recv: TPer in awaiting IF_Recv state - IF_Recv with SP=01h; Spcf=ComID; Xfer=01h: All response returned no further data	PASS
(A4-2-3-2-3) IF_Recv: TPer in awaiting IF_Recv state - IF_Recv with Xfer=insufficient; TPer stays in awaiting IF_Recv state	PASS
A5: Check ComPacket/Packet/SubPacket	
(A5-1-1-1-2) IF_Send ComPacket - Reserved field != 0; IF_Send: pass	PASS
(A5-1-2-2-2) IF_Send ComPacket - ComID != current ID; TPer in awaiting IF_Send state	PASS
(A5-1-2-5-1) IF_Send ComPacket - ComID Extension != 0; TPer in awaiting IF_Send state	PASS
(A5-1-3-1-2) IF_Send ComPacket - OutstandingData != 0; IF_Send: pass	PASS
(A5-1-4-1-2) IF_Send ComPacket - MinTransfer!= 0; IF_Send: pass	PASS
(A5-1-5-1-2) IF_Send ComPacket - Length > Xfer-data length; TPer in awaiting IF_Send state	PASS
(A5-1-5-1-2(2)) IF_Send ComPacket - Length < 24; TPer in awaiting IF_Send state	PASS
(A5-1-5-1-3) IF_Send ComPacket - Padding byte != 0; IF_Send: pass	PASS
(A5-2-3-1-2) IF_Send Packet - Reserved field != 0; IF_Send: pass	PASS
(A5-2-6-1-2) IF_Send Packet - Length > Xfer-data length of Compacket; no data returned	PASS
(A5-2-6-1-2(2)) IF_Send Packet - Length < 12 of Subpacket; no data returned	PASS
(A5-3-1-1-1(1)) IF_Send Packet - non-aligned with 4 byte in the start point of Subpacket; no data returned	N/A
(A5-3-1-1-2) IF_Send SubPacket - Reserved field != 0; IF_Send: pass	PASS
(A5-3-3-1-2) IF_Send SubPacket - Length > Packet; no data returned	PASS
(A5-2-1-1-2) IF_Send Packet in regular session - Session ID != open's session number; IF_Recv: no data returned	PASS
(A5-2-6-1-2) IF_Send Packet in regular session - Length > Xfer-data length of Compacket; Session abortion	PASS
(A5-2-6-1-2(2)) IF_Send Packet in regular session - Length < 12 of Subpacket; Session abortion	PASS
(A5-3-3-1-2) IF_Send SubPacket - Length > Packet; Session abortion	PASS
A7: Transaction check	
(A7-1-1-2-1(2)) StartTransaction Request: status != 0; StartTransaction Response: Pass with status = 0	PASS
(A7-1-1-2-1(2)) StartTransaction Request: status = 0 with short atom(81h); StartTransaction Response: Pass	PASS
(A7-1-1-2-1(2)) StartTransaction Request: status = 0 with medium atom(C001h); StartTransaction Response: Pass	PASS
(A7-1-1-2-1(2)) StartTransaction Request: status = 0 with long atom(E0000001h); StartTransaction Response: Pass	PASS
(A7-1-1-2-5) StartTransaction Request: status = 0 with byte atom; StartTransaction Response: Session Abort	PASS
(A7-1-1-2-5) StartTransaction Request: status = 0 with integer atom; StartTransaction Response: Session Abort	PASS
(A7-1-1-2-6) StartTransaction Request: no status encoded; StartTransaction Response: Session Abort	PASS
(A7-1-2-2-1(1)) EndTransaction Request: status = 0 with short atom(81h); EndTransaction Response: Pass	PASS
(A7-1-2-2-1(1)) EndTransaction Request: status = 0 with medium atom(C001h); EndTransaction Response: Pass	PASS
(A7-1-2-2-1(1)) EndTransaction Request: status = 0 with long atom(E0000001h); EndTransaction Response: Pass	PASS
(A7-1-2-2-1(2)) EndTransaction Request: status != 0 with short atom(81h); EndTransaction Response: Pass with status != 0	PASS
(A7-1-2-2-1(2)) EndTransaction Request: status != 0 with medium atom(C001h); EndTransaction Response: Pass with status != 0	PASS
(A7-1-2-2-1(2)) EndTransaction Request: status != 0 with long atom(E0000001h); EndTransaction Response: Pass with status != 0	PASS
(A7-1-2-2-5) EndTransaction Request: status = 0 with byte atom; EndTransaction Response: Session Abort	PASS
(A7-1-2-2-5) EndTransaction Request: status = 0 with integer atom; EndTransaction Response: Session Abort	PASS
(A7-1-2-2-6) EndTransaction Request: no status encoded; EndTransaction Response: Session Abort	PASS
(A7-1-3-1-1) Trans-Start attempt: StartTransaction <= MaxTransLimit; Response: pass	PASS
(A7-1-3-1-2) Trans-Start attempt: StartTransaction > MaxTransLimit; Response: Session Abort	PASS
(A7-1-3-2-1) Trans-End attempt: EndTransaction Request: outside of a transaction with status = 0; Response: Session Abort	PASS
(A7-1-3-2-2) Trans-End attempt: EndTransaction Request: within a transaction with status = 0; Response (commit): pass	PASS
(A7-1-3-3-1) Trans-Abort attempt: EndTransaction Request: outside of a transaction with status = 1; Response: Session Abort	PASS
(A7-1-3-3-2) Trans-Abort attempt: EndTransaction Request: within a transaction with status = 0; Response (abort): pass	PASS
(A7-1-3-4-1) Stand-Alone: StartTransaction Request: only with Start_Trans token and status token; Response: Pass	PASS
(A7-1-3-5-1) Stand-Alone: EndTransaction Request: only with End_Trans token and status token; Response: Pass	PASS
(A7-1-3-6-1) Multiple Trans: Trans-Start request after one or more; Response: Pass	PASS
(A7-1-3-8-1) Trans-attempt in a CtrlSession: Trans-Start request outside of methodInvoke: Token shall be discarded	PASS
(A7-1-3-8-1) Trans-attempt in a CtrlSession: Trans-End request outside of methodInvoke: Token shall be discarded	PASS
(A7-1-6-1-1) Trans+Session Abort: Transaction is aborted after session gets aborted	PASS
(A7-1-7-1-1) Trans+Session Close: Transaction is aborted after session gets closed	PASS
(A7-1-8-1-1) Trans+Session Close: Transaction is aborted after power cycle	PASS
A8: Test Ending Session	
(A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned	PASS
(A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans and EndTrans; Session shall be closed	PASS
(A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed	PASS
(A8-1-1-1-1(3)) EndSession Grammar: End Session is encoded outside of a method invocation in a control session; End token shall be discarded	PASS
(A8-1-2-1-1) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed	PASS
(A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass	PASS
(A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed	PASS

(A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession	PASS
(A8-2-3-1-1) Session after CloseSession: Start a new session shall pass after the Session is aborted	PASS
(A8-3-2-1-1) Session Timeout: If session# = MaxSessions and a session is timeout; Start/Sync Session - pass	PASS
(A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass	PASS
A9: Check Empty Atom	
(A9-1-1-1-1) StartSession - '0xFF' before a call token(0xF8); SyncSession: pass	PASS
(A9-1-1-2-1) StartSession - '0xFF' between a call token and an 'InvokingID'; SyncSession: pass	PASS
(A9-1-1-3-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass	PASS
(A9-1-1-4-1) StartSession - '0xFF' between a 'MethodID' and 'F0'; SyncSession: pass	PASS
(A9-1-1-5-1) StartSession - '0xFF' among HostSID and SPUID parameters; SyncSession: pass	PASS
(A9-1-1-5-1) StartSession - '0xFF' among SPUID and Write paramters; SyncSession: pass	PASS
(A9-1-1-6-1) StartSession - '0xFF' between endList('F1') and endData('F9'); SyncSession: pass	PASS
(A9-1-1-7-1) StartSession - '0xFF' between endData('F9') and statusCode('F0'); SyncSession: pass	PASS
(A9-1-1-8-1) StartSession - '0xFF' among tokens and statusCode list; SyncSession: pass	PASS
(A9-1-1-9-1) StartSession - '0xFF' after statusCode list's ending; SyncSession: pass	PASS
(A9-1-1-10-1) StartTransaction - '0xFF' before a TransactionStart token; Response: pass	PASS
(A9-1-1-11-1) EndTransaction - '0xFF' before a TransactionEnd token; Response: pass	PASS
(A9-1-1-12-1) StartTransaction - '0xFF' between a TransactionStart token and the status code; Response: pass	PASS
(A9-1-1-13-1) EndTransaction - '0xFF' between a TransactionEnd token and the status code; Response: pass	PASS
(A9-1-1-14-1) StartTransaction - '0xFF' after a TransactionStart token; Response: pass	PASS
(A9-1-1-15-1) EndTransaction - '0xFF' after a TransactionEnd token; Response: pass	PASS
(A9-1-2-1-1) StartSession - Empty atoms in plural places; SyncSession: pass	PASS
(A9-1-2-1-1) Get Request - Empty atoms in plural places; Get Response: pass	PASS
(A9-1-2-1-1) StartTransaction - Empty atoms in plural places; SyncSession: pass	PASS
A10: Set Properties test	
(A10-1-6-2-6) Set Host Properties - name in name-value not supported by TPer: Response - pass and the pair is ignored	PASS
(A10-1-6-3-1) Optional Params: Check Host Properties - MaxComPacketSize < 800h: Response value = 800h	PASS
(A10-1-6-3-1) Optional Params: Check Host Properties - MaxComPacketSize = 800h: Response value = 800h	PASS
(A10-1-6-5-1) Optional Params: Check Host Properties - MaxPacketSize < 7ECh: Response value = 7ECh	PASS
(A10-1-6-5-1) Optional Params: Check Host Properties - MaxPacketSize = 7ECh: Response value = 7ECh	PASS
(A10-1-6-6-1) Optional Params: Check Host Properties - MaxIndTokenSize < 7C8h: Response value = 7C8h	PASS
(A10-1-6-6-1) Optional Params: Check Host Properties - MaxIndTokenSize = 7C8h: Response value = 7C8h	PASS
(A10-1-6-7-1) Optional Params: Check Host Properties - MaxPackets = a number: Response value <=	PASS
(A10-1-6-8-1) Optional Params: Check Host Properties - MaxSubPackets = a number: Response value <= a number	PASS
(A10-1-6-9-1) Optional Params: Check Host Properties - MaxMethods = a number: Response value	PASS
(A10-1-6-15-1) Optional Params: Check Host Properties - Omission of HostParams: no HostParams	PASS
A10: Properties response and effect test	
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h	PASS
(A10-3-1-3-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h	PASS
(A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort	PASS
Properties Effect - TPerProp in regular session: = TPer's MaxComPacketSize; Response: Pass	PASS
(A10-3-2-1-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: ST = 51h at ATA interface level	PASS
(A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxPacketSize; Response:	N/A
(A10-3-2-4-1) Properties Effect - TPerProp in regular session: > TPer's MaxIndTokenSize; Response: Session abort	N/A
Properties Effect - TPerProp in control session: = TPer's MaxComPackets; Response - Pass	PASS
(A10-3-2-1-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: ST = 51h at ATA interface level	PASS
(A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxPackets; Response: Discarded by TPer	N/A
(A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer	N/A
(A10-3-2-6-1) Properties Effect - TPerProp in control session: > TPer's MaxSubPackets; Response: Discarded by TPer	PASS
(A10-3-2-15-1) Properties Effect - TPerProp in regular session: MaxAuthentications shall not be 1	PASS
A11: Test Start/SyncSession()	
(A11-1-1-1-1) StartSession - SessionID: not all 0; SyncSession - Status Code: 01h (Not_Authorized)	PASS
(A11-3-2-1-1) StartSession - HostSessionID: 4-byte uinteger(<0FFFFFFFFh); SyncSession - Pass and Tries = 0 in C_PIN table	PASS
(A11-3-2-1-1) StartSession - HostSessionID: 4-byte uinteger(=0FFFFFFFFh); SyncSession - Pass and Tries = 0 in C_PIN table	PASS
(A11-3-2-1-3) StartSession - HostSessionID: > 4-byte; SyncSession - Status Code: no data returned	PASS
(A11-3-2-2-2) StartSession - SPUID: nonexistent in the SP table; SyncSession - Status Code: 0Ch (Invalid_Param)	PASS
(A11-3-2-2-3) StartSession - SPUID: LockingSP in manufactured-inactive; SyncSession - Status Code: 0Ch (Invalid_Param)	PASS
(A11-3-2-3-3(2)) StartSession - Write: 1; SyncSession - Pass and Tries = 0 in C_PIN table	PASS
(A11-3-2-3-4) StartSession - Write: 2; SyncSession - Status Code: 0Ch (Invalid_Param)	PASS
A11: Test Start/SyncSession with Optional Parameters	
(A11-3-4-1-5) StartSession - HostChallenge: correct credential; SyncSession - Pass	PASS
(A11-3-4-1-6) StartSession - HostChallenge: correct credential(if Tries=TryLimit); SyncSession - Status Code: 01h or 12h (Not_Authorized/Authority_locked_out)	PASS
(A11-3-4-1-7) StartSession - HostChallenge: incorrect credential; SyncSession - Status Code: 01h (Not_Authorized)	PASS
(A11-3-4-1-10) StartSession - HostChallenge: anybody (explicitly in HostSignAuth); SyncSession -	PASS
(A11-3-4-1-11) StartSession - HostChallenge: omitted (any authority); SyncSession - Status Code: 0Ch (Invalid_Param)	PASS

(A11-3-4-2-6) StartSession - HostSignAuth: nonexistent UID; SyncSession - Status Code: 0Ch	PASS
(A11-3-4-2-6(2)) StartSession - HostSignAuth: disabled authority's UID; SyncSession - Status Code: 01h (Not_Authorized)	PASS
(A11-3-4-2-6(3)) StartSession - HostSignAuth: a class authority UID; SyncSession - Status Code: 0Ch (Invalid_Param)	PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential;	PASS
(A11-3-5-6-1-1) StartSession - exceed MaxSessions property; SyncSession - Status Code: 03h or 07h (SP_Busy/No_Sessions_Available)	PASS
Activating the Locking SP	
Start Session with HostChallenge - AdminSP	PASS
Sync Session - AdminSP	PASS
Activate_LockingSP	PASS
Activate_LockingSP - Response	PASS
Get - LifeCycle(Locking SP) - Request	PASS
Get - LifeCycle(Locking SP) - Response	PASS
Check the state of LockingSP	PASS
End Session - Request	PASS
End Session - Response	PASS
A6: Grammar Check on Method/InvokeUID in regular session	
(A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) Get Request - with medium atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) Get Request - with long atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass	PASS
(A6-0-1-1-1) Get Request - with long atom for MethodID; Response - Pass	PASS
(A6-1-1-1-1(1)) Get Request - with invalid token for 'Call'(F8h); Response - Session Abort	PASS
(A6-1-1-2-1) Get Request - with nonexistent InvokingID; Response - Status Code:	PASS
(A6-1-1-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-1-3-1(2)) Get Request - with non-8-long token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-2-2-1) Get Request - with nonexistent MethodID; Response - Status Code:	PASS
(A6-1-2-3-1(2)) Get Request - with non-byte token for MethodID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-2-3-1(2)) Get Request - with non-8-long token for MethodID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-3-1-1) Get Request - no ACE in the ACL; Response - empty data returned with SUCCESS status	PASS
(A6-1-3-1-1(2)) Get Request - nonexistent InvokingID/MethodID in ACL; Response - Status Code: 01h(Not_Authorized) and an empty results list	PASS
(A6-1-4-2-1) Get Request - with invalid token type of StartList: 0e0h; Response - Session Abort	PASS
(A6-1-5-2-1) Get Request - with invalid token type of EndList: 0e0h; Response - Session Abort	PASS
(A6-1-6-2-1) Get Request - with invalid token type of EndData: 0e0h; Response - Session Abort	PASS
(A6-1-7-2-1) Get Request - with invalid token type of StatusCode Start: 0e0h; Response - Session	PASS
(A6-1-8-1-2) Get Request - with first Status token = 81h(short); Response - Pass	PASS
(A6-1-8-2-1) Get Request - with first Status Code != 0h(found in status code); Response - fail	PASS
(A6-1-8-2-1) Get Request - with first Status Code != 0h(not in the status code); Response - fail	PASS
(A6-1-8-3-2) Get Request - with second Status Code != 0h; Response - Normal	PASS
(A6-1-8-3-2) Get Request - with third Status Code != 0h; Response - Normal	PASS
(A6-1-8-6-1) Get Request - with 1st Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) Get Request - with 1st Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-8-6-1) Get Request - with 2nd Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) Get Request - with 2nd Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-8-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-9-2-1) Get Request - with invalid token type of StatusCode End: 0e0h; Response - Session	PASS
(A6-1-4-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param)	PASS
(A6-0-1-1-1) Set Request - with short atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) Set Request - with medium atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) Set Request - with long atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) Set Request - with medium atom for MethodID; Response - Pass	PASS
(A6-0-1-1-1) Set Request - with long atom for MethodID; Response - Pass	PASS
(A6-1-1-1-1(1)) Set Request - with invalid token for 'Call'(F8h); Response - Session Abort	PASS
(A6-1-1-2-1) Set Request - with nonexistent InvokingID; Response - Status Code:	PASS
(A6-1-1-3-1(2)) Set Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-1-3-1(2)) Set Request - with non-8-long token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-2-2-1) Set Request - with nonexistent MethodID; Response - Status Code:	PASS
(A6-1-2-3-1(2)) Set Request - with non-byte token for MethodID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-2-3-1(2)) Set Request - with non-8-long token for MethodID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-3-1-1) Set Request - no ACE in the ACL; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-4-2-1) Set Request - with invalid token type of StartList: 0e0h; Response - Session Abort	PASS
(A6-1-5-2-1) Set Request - with invalid token type of EndList: 0e0h; Response - Session Abort	PASS
(A6-1-6-2-1) Set Request - with invalid token type of EndData: 0e0h; Response - Session Abort	PASS
(A6-1-7-2-1) Set Request - with invalid token type of StatusCode Start: 0e0h; Response - Session	PASS
(A6-1-8-1-2) Set Request - with first Status token = 81h(short); Response - Pass	PASS
(A6-1-8-2-1) Set Request - with first Status Code != 0h(found in status code); Response - fail	PASS
(A6-1-8-2-1) Set Request - with first Status Code != 0h(not in the status code); Response - fail	PASS
(A6-1-8-3-2) Set Request - with second Status Code != 0h; Response - Normal	PASS
(A6-1-8-3-2) Set Request - with third Status Code != 0h; Response - Normal	PASS
(A6-1-8-6-1) Set Request - with 1st Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) Set Request - with 1st Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-8-6-1) Set Request - with 2nd Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) Set Request - with 2nd Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-8-6-1) Set Request - with 3rd Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) Set Request - with 3rd Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-9-2-1) Set Request - with invalid token type of StatusCode End: 0e0h; Response - Session	PASS
(A6-1-4-2-1(1)) Set Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param)	PASS
(A6-1-4-2-1(2)) Set Request - with the same optional parameter encoded twice; Response - Status Code: 0Ch(Invalid_Param)	PASS
(A6-1-4-2-1(3)) Set Request - with the descending order of optional parameter; Response - Status Code: 0Ch(Invalid_Param)	PASS
(A6-0-1-1-1) Next Request - with short atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) Next Request - with medium atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) Next Request - with long atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) Next Request - with medium atom for MethodID; Response - Pass	PASS

(A6-0-1-1-1) Next Request - with long atom for MethodID; Response - Pass	PASS
(A6-1-1-1-1(1)) Next Request - with invalid token for 'Call'(F8h); Response - Session Abort	PASS
(A6-1-1-2-1) Next Request - with nonexistent InvokingID; Response - Status Code:	PASS
(A6-1-1-3-1(2)) Next Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-1-3-1(2)) Next Request - with non-8-long token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-2-2-1) Next Request - with nonexistent MethodID; Response - Status Code:	PASS
(A6-1-2-3-1(2)) Next Request - with non-byte token for MethodID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-2-3-1(2)) Next Request - with non-8-long token for MethodID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-3-1-1(2)) Next Request - nonexistent InvokingID/MethodID in ACL; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-4-2-1) Next Request - with invalid token type of StartList: 0e0h; Response - Session Abort	PASS
(A6-1-5-2-1) Next Request - with invalid token type of EndList: 0e0h; Response - Session Abort	PASS
(A6-1-6-2-1) Next Request - with invalid token type of EndData: 0e0h; Response - Session Abort	PASS
(A6-1-7-2-1) Next Request - with invalid token type of StatusCode Start: 0e0h; Response - Session	PASS
(A6-1-8-1-2) Next Request - with first Status token = 81h(short); Response - Pass	PASS
(A6-1-8-2-1) Next Request - with first Status Code != 0h(found in status code); Response - fail	PASS
(A6-1-8-2-1) Next Request - with first Status Code != 0h(not in the status code); Response - fail	PASS
(A6-1-8-3-2) Next Request - with second Status Code != 0h; Response - Normal	PASS
(A6-1-8-3-2) Next Request - with third Status Code != 0h; Response - Normal	PASS
(A6-1-8-6-1) Next Request - with 1st Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) Next Request - with 1st Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-8-6-1) Next Request - with 2nd Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) Next Request - with 2nd Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-8-6-1) Next Request - with 3rd Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) Next Request - with 3rd Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-9-2-1) Next Request - with invalid token type of StatusCode End: 0e0h; Response - Session	PASS
(A6-1-4-2-1(1)) Next Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param)	PASS
(A6-1-4-2-1(2)) Next Request - with the same optional parameter encoded twice; Response - Status Code: 0Ch(Invalid_Param)	PASS
(A6-1-4-2-1(3)) Next Request - with the descending order of optional parameter; Response - Status Code: 0Ch(Invalid_Param)	PASS
(A6-0-1-1-1) GetACL Request - with short atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) GetACL Request - with medium atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) GetACL Request - with long atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) GetACL Request - with medium atom for MethodID; Response - Pass	PASS
(A6-0-1-1-1) GetACL Request - with long atom for MethodID; Response - Pass	PASS
(A6-1-1-1-1(1)) GetACL Request - with invalid token for 'Call'(F8h); Response - Session Abort	PASS
(A6-1-1-2-1) GetACL Request - with nonexistent InvokingID; Response - Status Code:	PASS
(A6-1-1-3-1(2)) GetACL Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-1-3-1(2)) GetACL Request - with non-8-long token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-2-2-1) GetACL Request - with nonexistent MethodID; Response - Status Code:	PASS
(A6-1-2-3-1(2)) GetACL Request - with non-byte token for MethodID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-2-3-1(2)) GetACL Request - with non-8-long token for MethodID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-4-2-1) GetACL Request - with invalid token type of StartList: 0e0h; Response - Session Abort	PASS
(A6-1-5-2-1) GetACL Request - with invalid token type of EndList: 0e0h; Response - Session Abort	PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort	PASS
(A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Start: 0e0h; Response - Session	PASS
(A6-1-8-1-2) GetACL Request - with first Status token = 81h(short); Response - Pass	PASS
(A6-1-8-2-1) GetACL Request - with first Status Code != 0h(found in status code); Response - fail	PASS
(A6-1-8-2-1) GetACL Request - with first Status Code != 0h(not in the status code); Response - fail	PASS
(A6-1-8-3-2) GetACL Request - with second Status Code != 0h; Response - Normal	PASS
(A6-1-8-3-2) GetACL Request - with third Status Code != 0h; Response - Normal	PASS
(A6-1-8-6-1) GetACL Request - with 1st Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) GetACL Request - with 1st Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-8-6-1) GetACL Request - with 2nd Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-9-2-1) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Session	PASS
(A6-1-4-2-1(1)) GetACL Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param)	PASS
(A6-0-1-1-1) GenKey Request - with short atom for InvokingID; Response - Pass	N/A
(A6-0-1-1-1) GenKey Request - with medium atom for InvokingID; Response - Pass	N/A
(A6-0-1-1-1) GenKey Request - with long atom for InvokingID; Response - Pass	N/A
(A6-0-1-1-1) GenKey Request - with medium atom for MethodID; Response - Pass	N/A
(A6-0-1-1-1) GenKey Request - with long atom for MethodID; Response - Pass	N/A
(A6-1-1-1-1(1)) GenKey Request - with invalid token for 'Call'(F8h); Response - Session Abort	N/A
(A6-1-1-2-1) GenKey Request - with nonexistent InvokingID; Response - Status Code:	N/A
(A6-1-1-3-1(2)) GenKey Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized)	N/A
(A6-1-1-3-1(2)) GenKey Request - with non-8-long token for InvokingID; Response - Status Code: 01h(Not_Authorized)	N/A
(A6-1-2-2-1) GenKey Request - with nonexistent MethodID; Response - Status Code:	N/A
(A6-1-2-3-1(2)) GenKey Request - with non-byte token for MethodID; Response - Status Code: 01h(Not_Authorized)	N/A
(A6-1-2-3-1(2)) GenKey Request - with non-8-long token for MethodID; Response - Status Code: 01h(Not_Authorized)	N/A
(A6-1-3-1-1) GenKey Request - no ACE in the ACL; Response - Status Code: 01h(Not_Authorized)	N/A
(A6-1-3-1-1(2)) GenKey Request - nonexistent InvokingID/MethodID in ACL; Response - Status Code: 01h(Not_Authorized)	N/A
(A6-1-4-2-1) GenKey Request - with invalid token type of StartList: 0e0h; Response - Session Abort	N/A
(A6-1-5-2-1) GenKey Request - with invalid token type of EndList: 0e0h; Response - Session Abort	N/A
(A6-1-6-2-1) GenKey Request - with invalid token type of EndData: 0e0h; Response - Session Abort	N/A
(A6-1-7-2-1) GenKey Request - with invalid token type of StatusCode Start: 0e0h; Response - Session	N/A
(A6-1-8-1-2) GenKey Request - with first Status token = 81h(short); Response - Pass	N/A
(A6-1-8-2-1) GenKey Request - with first Status Code != 0h(found in status code); Response - fail	N/A
(A6-1-8-2-1) GenKey Request - with first Status Code != 0h(not in the status code); Response - fail	N/A
(A6-1-8-3-2) GenKey Request - with second Status Code != 0h; Response - Normal	N/A
(A6-1-8-3-2) GenKey Request - with third Status Code != 0h; Response - Normal	N/A
(A6-1-8-6-1) GenKey Request - with 1st Status token = A1h(byte); Response - Session Abort	N/A

(A6-1-8-6-1) GenKey Request - with 1st Status token = 91h(integer); Response - Session Abort	N/A
(A6-1-8-6-1) GenKey Request - with 2nd Status token = A1h(byte); Response - Session Abort	N/A
(A6-1-8-6-1) GenKey Request - with 2nd Status token = 91h(integer); Response - Session Abort	N/A
(A6-1-8-6-1) GenKey Request - with 3rd Status token = A1h(byte); Response - Session Abort	N/A
(A6-1-8-6-1) GenKey Request - with 3rd Status token = 91h(integer); Response - Session Abort	N/A
(A6-1-9-2-1) GenKey Request - with invalid token type of StatusCode End: 0e0h; Response - Session	N/A
A6: Grammar check on Method/InvokeUID in control session	
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared	PASS
(A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response	PASS
(A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response	PASS
(A6-3-2-2-1) Request - with nonexistent MethodID; Response - no response prepared	PASS
(A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared	PASS
(A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared	PASS
(A6-3-2-3-1(2)) Request - unexpected token(98: integer) in MethodID; Response - no response	PASS
(A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response	PASS
(A6-3-4-2-1) Request - with invalid token type of StartList: 0e0h; Response - no response prepared	PASS
(A6-3-4-2-1(1)) Request - without 'F2' for the beginning of Name-Value; Response - no response prepared or Status Code: 0Ch(invalid_param)	PASS
(A6-3-4-2-1(1)) Request - with byte atom for value in Name-Value; Response - no response prepared or Status Code: 0Ch(invalid_param)	PASS
(A6-3-4-2-1(1)) Request - without 'F3' for the ending of Name-Value; Response - no response prepared or Status Code: 0Ch(invalid_param)	PASS
(A6-3-4-2-1(2)) Request - Host properties encoded twice; Response - Status Code:	PASS
(A6-3-5-2-1) Request - with invalid token type of EndList: 0e0h; Response - no response prepared	PASS
(A6-3-6-2-1) Request - with invalid token type of EndData: 0e0h; Response - no response prepared	PASS
(A6-3-7-2-1) Request - with invalid token type of StatusCode Start: 0e0h; Response - no response	PASS
(A6-3-8-1-2) Request - with first Status token = 81h(short); Response - pass	PASS
(A6-3-8-2-1) Request - with first Status Code != 0h(found in status code); Response - fail	PASS
(A6-3-8-3-2) Request - with second Status Code != 0h; Response - Normal	PASS
(A6-3-8-3-2) Request - with third Status Code != 0h; Response - Normal	PASS
(A6-3-8-6-1) Request - with non-uinteger(byte) atom for 1st statusCode; Response - no response	PASS
(A6-3-8-6-1) Request - with non-uinteger(integer) atom for 2nd statusCode; Response - no response	PASS
(A6-3-8-6-1) Request - with non-uinteger(integer) atom for 3rd statusCode; Response - no response	PASS
(A6-3-9-2-1) Request - with invalid token type of StatusCode End: 0e0h; Response - no response	PASS
(A6-3-4-2-1(3)) StartSession Request - with non-ascending order of optional parameter; Response - Status Code: 0Ch(invalid_param)	PASS
A12: Get() - Byte Table Grammar check	
(A12-0-1-1-1) DataStore RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) DataStore RequiredParams: Get with 'EndRow' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-1-1-4-5(2)) DataStore RequiredParams: Get with 'StartRow' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-1-1-4-10) DataStore RequiredParams: Get without 'StartRow' component; Get response - Pass	PASS
(A12-1-1-5-6) DataStore RequiredParams: Get with 'EndRow' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-1-1-5-10) DataStore RequiredParams: Get without 'EndRow' component; Get response - Pass	PASS
(A12-1-1-5-11) DataStore RequiredParams: Get with 'EndRow' encoded prior to 'StartRow'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-1-1-5-12) DataStore RequiredParams: Get with the number of 'StartRow' > 'EndRow'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-1-1-6-1) DataStore RequiredParams: Get with 'StartColumn'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-1-1-7-1) DataStore RequiredParams: Get with 'EndColumn'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-1) MBR RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	N/A
(A12-0-1-1-2) MBR RequiredParams: Get with 'EndRow' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	N/A
(A12-1-1-4-5(2)) MBR RequiredParams: Get with 'StartRow' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	N/A
(A12-1-1-4-10) MBR RequiredParams: Get without 'StartRow' component; Get response - Pass	N/A
(A12-1-1-5-6) MBR RequiredParams: Get with 'EndRow' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	N/A
(A12-1-1-5-10) MBR RequiredParams: Get without 'EndRow' component; Get response - Pass	N/A
(A12-1-1-5-11) MBR RequiredParams: Get with 'EndRow' encoded prior to 'StartRow'; Get response - Status Code: 0Ch (Invalid_Param)	N/A
(A12-1-1-5-12) MBR RequiredParams: Get with the number of 'StartRow' > 'EndRow'; Get response - Status Code: 0Ch (Invalid_Param)	N/A
(A12-1-1-6-1) MBR RequiredParams: Get with 'StartColumn'; Get response - Status Code: 0Ch	N/A
(A12-1-1-7-1) MBR RequiredParams: Get with 'EndColumn'; Get response - Status Code: 0Ch	N/A
A12: Get() - Object Table to AdminSP Grammar check	
(A12-0-1-1-1) Table RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) Table RequiredParams: Get with 'StartColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) Table RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-4-1) Table RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch	PASS
(A12-3-1-5-1) Table RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch	PASS
(A12-3-1-6-6) Table RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-6-10) Table RequiredParams: Get without 'StartCol' component; Get response - Pass	PASS
(A12-3-1-7-6) Table RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-9) Table RequiredParams: Get without 'EndCol' component; Get response - Pass	PASS
(A12-3-1-7-10) Table RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-10(2)) Table RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-1) SPInfo RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) SPInfo RequiredParams: Get with 'StartColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) SPInfo RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-4-1) SPInfo RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch	PASS

(A12-3-1-5-1) SPInfo RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch	PASS
(A12-3-1-6-6) SPInfo RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-6-10) SPInfo RequiredParams: Get without 'StartCol' component; Get response - Pass	PASS
(A12-3-1-7-6) SPInfo RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-9) SPInfo RequiredParams: Get without 'EndCol' component; Get response - Pass	PASS
(A12-3-1-7-10) SPInfo RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-10(2)) SPInfo RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-1) SPTemplates RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) SPTemplates RequiredParams: Get with 'StartColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) SPTemplates RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-4-1) SPTemplates RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-5-1) SPTemplates RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-6-6) SPTemplates RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-6-10) SPTemplates RequiredParams: Get without 'StartCol' component; Get response -	PASS
(A12-3-1-7-6) SPTemplates RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-9) SPTemplates RequiredParams: Get without 'EndCol' component; Get response - Pass	PASS
(A12-3-1-7-10) SPTemplates RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-10(2)) SPTemplates RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-1) MethodID RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) MethodID RequiredParams: Get with 'StartColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) MethodID RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-4-1) MethodID RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-5-1) MethodID RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch	PASS
(A12-3-1-6-6) MethodID RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-6-10) MethodID RequiredParams: Get without 'StartCol' component; Get response - Pass	PASS
(A12-3-1-7-6) MethodID RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-9) MethodID RequiredParams: Get without 'EndCol' component; Get response - Pass	PASS
(A12-3-1-7-10) MethodID RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-10(2)) MethodID RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-1) ACE RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) ACE RequiredParams: Get with 'StartColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) ACE RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-4-1) ACE RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch	PASS
(A12-3-1-5-1) ACE RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch	PASS
(A12-3-1-6-6) ACE RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-6-10) ACE RequiredParams: Get without 'StartCol' component; Get response - Pass	PASS
(A12-3-1-7-6) ACE RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-9) ACE RequiredParams: Get without 'EndCol' component; Get response - Pass	PASS
(A12-3-1-7-10) ACE RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-10(2)) ACE RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-1) Authority RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) Authority RequiredParams: Get with 'StartColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) Authority RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-4-1) Authority RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-5-1) Authority RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch	PASS
(A12-3-1-6-6) Authority RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-6-10) Authority RequiredParams: Get without 'StartCol' component; Get response - Pass	PASS
(A12-3-1-7-6) Authority RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-9) Authority RequiredParams: Get without 'EndCol' component; Get response - Pass	PASS
(A12-3-1-7-10) Authority RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-10(2)) Authority RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-1) C_PIN RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) C_PIN RequiredParams: Get with 'StartColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) C_PIN RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-4-1) C_PIN RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch	PASS
(A12-3-1-5-1) C_PIN RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch	PASS
(A12-3-1-6-6) C_PIN RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-6-10) C_PIN RequiredParams: Get without 'StartCol' component; Get response - Pass	PASS

(A12-3-1-7-6) C_PIN RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-9) C_PIN RequiredParams: Get without 'EndCol' component; Get response - Pass	PASS
(A12-3-1-7-10) C_PIN RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-10(2)) C_PIN RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-1) TPerInfo RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) TPerInfo RequiredParams: Get with 'StartColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) TPerInfo RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-4-1) TPerInfo RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch	PASS
(A12-3-1-5-1) TPerInfo RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch	PASS
(A12-3-1-6-6) TPerInfo RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-6-10) TPerInfo RequiredParams: Get without 'StartCol' component; Get response - Pass	PASS
(A12-3-1-7-6) TPerInfo RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-9) TPerInfo RequiredParams: Get without 'EndCol' component; Get response - Pass	PASS
(A12-3-1-7-10) TPerInfo RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-10(2)) TPerInfo RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-1) Template RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) Template RequiredParams: Get with 'StartColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) Template RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-4-1) Template RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-5-1) Template RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch	PASS
(A12-3-1-6-6) Template RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-6-10) Template RequiredParams: Get without 'StartCol' component; Get response - Pass	PASS
(A12-3-1-7-6) Template RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-9) Template RequiredParams: Get without 'EndCol' component; Get response - Pass	PASS
(A12-3-1-7-10) Template RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-10(2)) Template RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-1) SP RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) SP RequiredParams: Get with 'StartColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) SP RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-4-1) SP RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch	PASS
(A12-3-1-5-1) SP RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch	PASS
(A12-3-1-6-6) SP RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-6-10) SP RequiredParams: Get without 'StartCol' component; Get response - Pass	PASS
(A12-3-1-7-6) SP RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-9) SP RequiredParams: Get without 'EndCol' component; Get response - Pass	PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-10(2)) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
A13: Set() - Byte Table Grammar check	
(A13-2-1-2-5) DataStore OptParams-where: Set with the limit of the byte table; Set response - pass	PASS
(A13-2-1-2-6) DataStore OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param)	PASS
(A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass	PASS
(A13-2-1-3-5) DataStore OptParams-value: Set with data whthin limit of the table; Set response -	PASS
(A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param)	PASS
(A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response -	PASS
(A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - pass	N/A
(A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param)	N/A
(A13-2-1-2-9) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass	N/A
(A13-2-1-3-5) MBR OptParams-value: Set with data whthin limit of the table; Set response - Pass	N/A
(A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param)	N/A
(A13-2-1-3-9) MBR OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass	N/A
A13: Set() - Object Table (LockingSP) Grammar check	
(A13-4-1-2-1) Authority OptParams-where: Set with 'Where' parameter; Set response - Status Code: 0Ch (Invalid_Param)	PASS
(A13-4-1-4-15) Authority OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param)	PASS
(A13-4-1-2-1) Locking OptParams-where: Set with 'Where' parameter; Set response - Status Code: 0Ch (Invalid_Param)	PASS
(A13-4-1-4-14) Locking OptParams-where: Set with ColumnName-Value not encoded in ascending order; Set response - Pass	PASS
(A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param)	PASS
(A13-4-1-2-1) MBRControl OptParams-where: Set with 'Where' parameter; Set response - Status Code: 0Ch (Invalid_Param)	N/A
(A13-4-1-4-14) MBRControl OptParams-where: Set with ColumnName-Value not encoded in ascending order; Set response - Pass	N/A
(A13-4-1-4-15) MBRControl OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param)	N/A
A14: Next()-AdminSP Basic Grammar check	
(A14-1-3-2-5(2)) Table OptParams-where: Next with an exiting UID in the table; Next response - Pass	PASS

(A14-1-3-2-8) Table OptParams-where: Next with nonexistent UID; Next response - Status Code: 0Ch (Invalid_Param)	PASS
(A14-1-3-2-11) Table OptParams-where: Next with omitted 'Where' parameter; Next response - first UID in the table	PASS
(A14-1-3-3-6) Table OptParams-count: Next with a larger the number of UIDs; Next response - all	PASS
(A14-1-3-3-6(2)) Table OptParams-count: Next with count = 0; Next response - no UID returned	PASS
(A14-1-3-3-10) Table OptParams-count: Next with omitted count; Next response - Pass	PASS
(A14-1-3-2-5(2)) SPTemplates OptParams-where: Next with an exiting UID in the table; Next	PASS
(A14-1-3-2-8) SPTemplates OptParams-where: Next with nonexistent UID; Next response - Status Code: 0Ch (Invalid_Param)	PASS
(A14-1-3-2-11) SPTemplates OptParams-where: Next with omitted 'Where' parameter; Next response - first UID in the table	PASS
(A14-1-3-3-6) SPTemplates OptParams-count: Next with a larger the number of UIDs; Next response -	PASS
(A14-1-3-3-6(2)) SPTemplates OptParams-count: Next with count = 0; Next response - no UID	PASS
(A14-1-3-3-10) SPTemplates OptParams-count: Next with omitted count; Next response - Pass	PASS
(A14-1-3-2-5(2)) MethodID OptParams-where: Next with an exiting UID in the table; Next response -	PASS
(A14-1-3-2-8) MethodID OptParams-where: Next with nonexistent UID; Next response - Status Code: 0Ch (Invalid_Param)	PASS
(A14-1-3-2-11) MethodID OptParams-where: Next with omitted 'Where' parameter; Next response - first UID in the table	PASS
(A14-1-3-3-6) MethodID OptParams-count: Next with a larger the number of UIDs; Next response -	PASS
(A14-1-3-3-6(2)) MethodID OptParams-count: Next with count = 0; Next response - no UID returned	PASS
(A14-1-3-3-10) MethodID OptParams-count: Next with omitted count; Next response - Pass	PASS
(A14-1-3-2-5(2)) ACE OptParams-where: Next with an exiting UID in the table; Next response - Pass	PASS
(A14-1-3-2-8) ACE OptParams-where: Next with nonexistent UID; Next response - Status Code: 0Ch (Invalid_Param)	PASS
(A14-1-3-2-11) ACE OptParams-where: Next with omitted 'Where' parameter; Next response - first UID in the table	PASS
(A14-1-3-3-6) ACE OptParams-count: Next with a larger the number of UIDs; Next response - all UIDs	PASS
(A14-1-3-3-6(2)) ACE OptParams-count: Next with count = 0; Next response - no UID returned	PASS
(A14-1-3-3-10) ACE OptParams-count: Next with omitted count; Next response - Pass	PASS
(A14-1-3-2-5(2)) Authority OptParams-where: Next with an exiting UID in the table; Next response -	PASS
(A14-1-3-2-8) Authority OptParams-where: Next with nonexistent UID; Next response - Status Code: 0Ch (Invalid_Param)	PASS
(A14-1-3-2-11) Authority OptParams-where: Next with omitted 'Where' parameter; Next response - first UID in the table	PASS
(A14-1-3-3-6) Authority OptParams-count: Next with a larger the number of UIDs; Next response -	PASS
(A14-1-3-3-6(2)) Authority OptParams-count: Next with count = 0; Next response - no UID returned	PASS
(A14-1-3-3-10) Authority OptParams-count: Next with omitted count; Next response - Pass	PASS
(A14-1-3-2-5(2)) C_PIN OptParams-where: Next with an exiting UID in the table; Next response -	PASS
(A14-1-3-2-8) C_PIN OptParams-where: Next with nonexistent UID; Next response - Status Code: 0Ch (Invalid_Param)	PASS
(A14-1-3-2-11) C_PIN OptParams-where: Next with omitted 'Where' parameter; Next response - first UID in the table	PASS
(A14-1-3-3-6) C_PIN OptParams-count: Next with a larger the number of UIDs; Next response - all	PASS
(A14-1-3-3-6(2)) C_PIN OptParams-count: Next with count = 0; Next response - no UID returned	PASS
(A14-1-3-3-10) C_PIN OptParams-count: Next with omitted count; Next response - Pass	PASS
(A14-1-3-2-5(2)) Template OptParams-where: Next with an exiting UID in the table; Next response -	PASS
(A14-1-3-2-8) Template OptParams-where: Next with nonexistent UID; Next response - Status Code: 0Ch (Invalid_Param)	PASS
(A14-1-3-2-11) Template OptParams-where: Next with omitted 'Where' parameter; Next response - first UID in the table	PASS
(A14-1-3-3-6) Template OptParams-count: Next with a larger the number of UIDs; Next response -	PASS
(A14-1-3-3-6(2)) Template OptParams-count: Next with count = 0; Next response - no UID returned	PASS
(A14-1-3-3-10) Template OptParams-count: Next with omitted count; Next response - Pass	PASS
(A14-1-3-2-5(2)) SP OptParams-where: Next with an exiting UID in the table; Next response - Pass	PASS
(A14-1-3-2-8) SP OptParams-where: Next with nonexistent UID; Next response - Status Code: 0Ch (Invalid_Param)	PASS
(A14-1-3-2-11) SP OptParams-where: Next with omitted 'Where' parameter; Next response - first UID in the table	PASS
(A14-1-3-3-6) SP OptParams-count: Next with a larger the number of UIDs; Next response - all UIDs	PASS
(A14-1-3-3-6(2)) SP OptParams-count: Next with count = 0; Next response - no UID returned	PASS
(A14-1-3-3-10) SP OptParams-count: Next with omitted count; Next response - Pass	PASS
A15: GetACL()-AdminSP Basic Grammar check	
(A15-1-1-0-1) Table Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-2-1-1(2)) Table ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-1-1(2)) Table ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL	PASS
(A15-1-2-2-1(2)) Table ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) Table ReqParams-methodID: GetACL with long atom for MethodID; GetACL	PASS
(A15-1-2-3-1) Table ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-1-0-1) SPInfo Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-2-1-1(2)) SPInfo ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-1-1(2)) SPInfo ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL	PASS
(A15-1-2-2-1(2)) SPInfo ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) SPInfo ReqParams-methodID: GetACL with long atom for MethodID; GetACL	PASS
(A15-1-2-3-1) SPInfo ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-1-0-1) SPTemplates Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-2-1-1(2)) SPTemplates ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-1-1(2)) SPTemplates ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) SPTemplates ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) SPTemplates ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-3-1) SPTemplates ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-1-0-1) MethodID Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority)	PASS

(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-3-1) MethodID ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL	PASS
(A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with medium atom for MethodID; GetACL	PASS
(A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response -	PASS
(A15-1-2-3-1) ACE ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-1-0-1) Authority Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) Authority ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) Authority ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-3-1) Authority ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-1-0-1) C_PIN Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-2-1-1(2)) C_PIN ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-1-1(2)) C_PIN ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL	PASS
(A15-1-2-2-1(2)) C_PIN ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) C_PIN ReqParams-methodID: GetACL with long atom for MethodID; GetACL	PASS
(A15-1-2-3-1) C_PIN ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-1-0-1) TPerInfo Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-2-1-1(2)) TPerInfo ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-1-1(2)) TPerInfo ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) TPerInfo ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) TPerInfo ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-3-1) TPerInfo ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-1-0-1) Template Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-2-1-1(2)) Template ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-1-1(2)) Template ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) Template ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) Template ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-3-1) Template ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-1-0-1) SP Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-2-1-1(2)) SP ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL	PASS
(A15-1-2-1-1(2)) SP ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response -	PASS
(A15-1-2-2-1(2)) SP ReqParams-methodID: GetACL with medium atom for MethodID; GetACL	PASS
(A15-1-2-2-1(2)) SP ReqParams-methodID: GetACL with long atom for MethodID; GetACL response -	PASS
(A15-1-2-3-1) SP ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority)	PASS
A19: RevertSP() Grammar check	
(A19-1-3-1-10) KeepGlbRange: RevertSP to LockingSP with the omitted KeepGlobalRangeKey; RevertSP Response - Pass	NM
Revert LockingSP	
Start Session with HostChallenge - AdminSP	PASS
Sync Session - AdminSP	PASS
LockingSP.Revert - Request	PASS
LockingSP.Revert - Response	PASS
End Session - Request	PASS
End Session - Response	PASS
Activating the Locking SP	
Start Session with HostChallenge - AdminSP	PASS
Sync Session - AdminSP	PASS
Activate_LockingSP	PASS
Activate_LockingSP - Response	PASS
Get - LifeCycle(Locking SP) - Request	PASS
Get - LifeCycle(Locking SP) - Response	PASS
Check the state of LockingSP	PASS
End Session - Request	PASS
End Session - Response	PASS
C1: Level 0 Discovery Contents	
(C1) Display the contents from Level0_Discovery	PASS
(C1) Check TPer Feature	PASS
(C1) Check Locking Feature	PASS
(C1) Check SSC Feature	PASS
C2: Properties Contents	

(C2) Properties Parameter and Host Properties Parameter	PASS
(C2) Check TPer properties	PASS
(C2(1)) Check Host properties	PASS
C3: Get() Byte Table: DataStore and MBR Contents check	
(C3-26) Set data to DataStore table	PASS
(C3-26) Get data from DataStore table	PASS
(C3-26) Data Comparison from DataStore table	PASS
(C3-24) Set data to MBR table	N/A
(C3-24) Get data from MBR table	N/A
(C3-24) Data Comparison from MBR table	N/A
C4: Next() Table Contents (AdminSP)	
(C4-1) Next() - Table Table	PASS
(C4-1) Verify UIDs for Table Table	PASS
(C4-3) Next() - SPTemplates Table	PASS
(C4-3) Verify UIDs for SPTemplates Table	PASS
(C4-4) Next() - MethodID Table	PASS
(C4-4) Verify UIDs for MethodID Table	PASS
(C4-6) Next() - ACE Table	PASS
(C4-6) Verify UIDs for ACE Table	PASS
(C4-7) Next() - Authority Table	PASS
(C4-7) Verify UIDs for Authority Table	PASS
(C4-8) Next() - C_PIN Table	PASS
(C4-8) Verify UIDs for C_PIN Table	PASS
(C4-10) Next() - Template Table	PASS
(C4-10) Verify UIDs for Template Table	PASS
(C4-11) Next() - SP Table	PASS
(C4-11) Verify UIDs for SP Table	PASS
C4: Next() Table Contents (LockingSP)	
(C4-12) Next() - Table Table	PASS
(C4-12) Verify UIDs for Table Table	PASS
(C4-14) Next() - SPTemplates Table	PASS
(C4-14) Verify UIDs for SPTemplates Table	PASS
(C4-16) Next() - MethodID Table	PASS
(C4-16) Verify UIDs for MethodID Table	PASS
(C4-18) Next() - ACE Table	PASS
(C4-18) Verify UIDs for ACE Table	PASS
(C4-19) Next() - Authority Table	PASS
(C4-19) Verify UIDs for Authority Table	PASS
(C4-20) Next() - C_PIN Table	PASS
(C4-20) Verify UIDs for C_PIN Table	PASS
(C4-22) Next() - Locking Table	PASS
(C4-22) Verify UIDs for Locking Table	PASS
(C4-23) Next() - RestrictedCmds Table	N/A
D1: ACE.Set() Grammar and Effect	
(D1-1-1-1-9) ACE.Set Grammar: Request with right params; Set response - pass	PASS
(D1-1-1-1-10) ACE.Set Grammar: Request with non-parsed boolean expression form; Set response - Session abort	PASS
(D1-1-1-1-11) ACE.Set Grammar: Request with at most the maximum size of AC_Element; Set	PASS
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (Invalid_Param)	N/A
(D1-1-2-1-1) ACE.Get is issued to verify; Data comparison - Matching	PASS
(D1-1-2-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching	PASS
(D1-1-3-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set	PASS
(D1-1-3-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value	PASS
D2: Authority.Set() testing	
(D2-1-2-1-1) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column	PASS
(D2-1-2-2-1) Authenticate - User1(Enabled=1); Authenticate Response - Success (AuthStatus = 01h)	PASS
(D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail	PASS
(D2-1-2-3-2) The previous successful authentication result with this authority in this session shall not be affected	N/A
(D2-1-2-2-1) Start Session - as User1(Enabled=1); Sync Session - Pass	PASS
(D2-1-2-3-1) Start Session - as User1(Enabled=0); Sync Session - Fail	PASS
(D2-1-3-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set	PASS
(D2-1-3-1-2) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value	PASS
D3: C_PIN.Set()	
(D3-1-2-1-2) Set Request: PIN = Null; Response: Pass	PASS
(D3-1-2-1-2) Set Request: PIN with 32 byte; Response: Pass	PASS
(D3-1-3-1-1) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set	PASS
(D3-1-3-1-2) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value	PASS
D4: Locking.Set() for 'RangeStart' and 'RangeLength'	
(D4-1-2-1-2) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: 0Ch (Invalid_Param)	N/A
(D4-1-3-1-1) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass	N/A
(D4-1-3-1-1) RangeStart/Len Effect: Get the values of 'RangeStart' and 'RangeLength'; Get() retrieves the values indicated by Set()	N/A
(D4-1-3-2-1) RangeStart/Len Effect: with 'RangeStart'=changed and 'RangeLength'!=0; Response with Get - the values as intended by Set()	N/A
(D4-1-3-3-1) RangeStart/Len Effect: with 'RangeStart'=changed and 'RangeLength'=0; Response with Get - no LBA covered by that range	N/A
(D4-1-3-4-1) RangeStart/Len Effect: with 'RangeLength'!=0; Response with Get - the values as intended by Set()	N/A
(D4-1-3-5-1) RangeStart/Len Effect: with 'RangeLength'=0; Response with Get - no LBA covered by	N/A
(D4-1-4-1-1) RangeStart/Len Effect in Trans: Set RangeStart in a transaction and endTran's status=0; The value retains the set value	N/A
(D4-1-4-1-2) RangeStart/Len Effect in Trans: Set RangeStart in a transaction and endTran's status=1; The value changes back to the original value	N/A
D4: Locking.Set() for 'ReadLockEnabled' and 'ReadLocked'	
(D4-2-2-1-1) RdLockEnabled/Locked: Set and Get the contents of 'ReadLockEnabled' and 'ReadLocked'; Get() retrieves the values indicated by Set()	PASS
(D4-2-2-2-1) RdLockEnabled/Locked=1 w/ inactive MBR shadowing: Read with this locked range; Response - Command abortion	N/A
(D4-2-2-2-1(2)) RdLockEnabled/Locked=1 w/ inactive MBR shadowing: Read with other range; Response - Command abortion	N/A
(D4-2-2-2-2) RdLockEnabled/Locked=1 w/ active MBR shadowing: Read with LBA covered by this range and not by MBR; Response - all-0 data returned	N/A
(D4-2-2-2-3) RdLockEnabled/Locked=1: Locked bit = 1 in Level 0 Discovery	N/A

(D4-2-2-2-5) RdLockEnabled/Locked=1: Power-on reset; Response - 'ReadLocked' = 1	N/A
(D4-2-2-3-1) RdLockEnabled/Locked=1/0 w/ inactive MBR shadowing: Read with this locked range; Response - Pass	N/A
(D4-2-2-3-1(2)) RdLockEnabled/Locked=1/0 w/ inactive MBR shadowing: Read with multiple ranges (range2); Response - abortion/pass(if rangeCrossing=1/0)	N/A
(D4-2-2-3-1(2)) RdLockEnabled/Locked=1/0 w/ inactive MBR shadowing: Read with multiple ranges (globalRange); Response - abortion/pass(if rangeCrossing=1/0)	N/A
(D4-2-2-3-2) RdLockEnabled/Locked=1/0 w/ active MBR shadowing: Read with LBA covered by this range and not by MBR; Response - pass	N/A
(D4-2-2-3-3) RdLockEnabled/Locked=1/0: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges)	N/A
(D4-2-2-3-5) RdLockEnabled/Locked=1/0: Power-on reset; Response - 'ReadLocked' = 1	N/A
(D4-2-2-4-1) RdLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Read with this range;	N/A
(D4-2-2-4-1(2)) RdLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Read with multiple ranges (globalRange); Response - abortion/pass(if rangeCrossing=1/0)	N/A
(D4-2-2-4-1(2)) RdLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Read with multiple ranges (range2); Response - abortion/pass(if rangeCrossing=1/0)	N/A
(D4-2-2-4-2) RdLockEnabled/Locked=0/0 w/ active MBR shadowing: Read with LBA covered by this range and not by MBR; Response - Pass	N/A
(D4-2-2-4-3) RdLockEnabled/Locked=0/0: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges)	N/A
(D4-2-2-4-1) RdLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Read with this range;	N/A
(D4-2-2-4-1(2)) RdLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Read with multiple ranges (globalRange); Response - abortion/pass(if rangeCrossing=1/0)	N/A
(D4-2-2-4-1(2)) RdLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Read with multiple ranges (range2); Response - abortion/pass(if rangeCrossing=1/0)	N/A
(D4-2-2-4-2) RdLockEnabled/Locked=0/1 w/ active MBR shadowing: Read with LBA covered by this range and not by MBR; Response - Pass	N/A
(D4-2-2-4-3) RdLockEnabled/Locked=0/1: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges)	N/A
(D4-2-3-1-1) ReadLock Effect in Trans: Set ReadLockEnabled in a transaction and endTran's status=0; The value retains the set value	PASS
(D4-2-3-1-2) ReadLock Effect in Trans: Set ReadLockEnabled in a transaction and endTran's status=1; The value changes back to the original value	PASS
D4: Locking.Set() for 'WriteLockEnabled' and 'WriteLocked'	
(D4-3-2-1-1) WrLockEnabled/Locked: Set WriteLockEnabled with tiny atom; Response - Pass	PASS
(D4-3-2-1-1) WrLockEnabled/Locked: Get the contents of 'WriteLockEnabled' and 'WriteLocked'; Get() retrieves the values indicated by Set()	PASS
(D4-3-2-2-1) WrLockEnabled/Locked=1/1 w/ inactive MBR shadowing: Write with this locked range; Response - Command abortion	N/A
(D4-3-2-2-1(2)) WrLockEnabled/Locked=1/1 w/ inactive MBR shadowing: Write with other range; Response - Command abortion	N/A
(D4-3-2-2-2) WrLockEnabled/Locked=1/1 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Command abortion	N/A
(D4-3-2-2-3) WrLockEnabled/Locked=1/1: Locked bit = 0 in Level 0 Discovery	N/A
(D4-3-2-2-5) WrLockEnabled/Locked=1/1: Power-on reset; Response - 'WriteLocked' = 1	N/A
(D4-3-2-3-1) WrLockEnabled/Locked=1/0 w/ inactive MBR shadowing: Write with this locked range; Response - Pass	N/A
(D4-3-2-3-1(2)) WrLockEnabled/Locked=1/0 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0)	N/A
(D4-3-2-3-1(2)) WrLockEnabled/Locked=1/0 w/ inactive MBR shadowing: Write with multiple ranges (globalrange); Response - Abortion/Pass(if rangeCrossing=1/0)	N/A
(D4-3-2-3-2) WrLockEnabled/Locked=1/0 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass	N/A
(D4-3-2-3-3) WrLockEnabled/Locked=1/0: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges)	N/A
(D4-3-2-3-5) WrLockEnabled/Locked=1/0: Power-on reset; Response - 'WriteLocked' = 1	N/A
(D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range;	N/A
(D4-3-2-4-1(2)) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0)	N/A
(D4-3-2-4-1(2)) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0)	N/A
(D4-3-2-4-2) WrLockEnabled/Locked=0/0 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass	N/A
(D4-3-2-4-3) WrLockEnabled/Locked=0/0: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges)	N/A
(D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range;	N/A
(D4-3-2-4-1(2)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0)	N/A
(D4-3-2-4-1(2)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0)	N/A
(D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass	N/A
(D4-3-2-4-3) WrLockEnabled/Locked=0/1: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges)	N/A
(D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value	PASS
(D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value	PASS
D5: MBRControl.Set() Grammar and Effect	
(D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass	N/A
(D5-1-2-1-1) Get Enable/Done value; Get() retrieves the values indicated by Set()	N/A
(D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0)	N/A
(D5-1-2-2-3) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0)	N/A
(D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	N/A
(D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	N/A
(D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	N/A
(D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	N/A
(D5-1-2-2-2) Enable/Done=1: Read command: abort (Read/WriteLockEnabled = 1)	N/A
(D5-1-2-2-3) Enable/Done=1: Write command: abort (Read/WriteLockEnabled = 1)	N/A
(D5-1-2-2-4) Enable/Done=1: 'MBRDone' bit = 1 from Level0_Discovery	N/A
(D5-1-2-2-4(2)) Enable/Done=1: 'MBREnable' bit = 1 from Level0_Discovery	N/A
(D5-1-2-2-1) Enable/Done=1: 'MBRDone' bit = 0 after power cycle	N/A
(D5-1-2-3-1) Enable/Done=1/0: 'MBRDone' bit = 0 after power cycle	N/A
(D5-1-2-3-2) Enable/Done=1/0: Read addressing ONLY LBA covered by MBR table; MBR data	N/A

(D5-1-2-3-2(2)) Enable/Done=1/0: Read addressing LBA covered by MBR table and not by MBR; Command aborted	N/A
(D5-1-2-3-3) Enable/Done=1/0: Write addressing ONLY LBA covered by MBR table; Write Command	N/A
(D5-1-2-3-3(2)) Enable/Done=1/0: Write addressing LBA covered by MBR table and not by MBR; Write Command aborted	N/A
(D5-1-2-3-6) Enable/Done=1/0: 'MBRDone' bit = 0 from Level0_Discovery	N/A
(D5-1-2-3-6(2)) Enable/Done=1/0: 'MBREnable' bit = 1 from Level0_Discovery	N/A
(D5-1-2-4-1) Enable/Done=0/0: Read command: pass (Read/WriteLockEnabled = 0)	N/A
(D5-1-2-4-2) Enable/Done=0/0: Write command: pass (Read/WriteLockEnabled = 0)	N/A
(D5-1-2-4-1) Enable/Done=0/0: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	N/A
(D5-1-2-4-2) Enable/Done=0/0: Write with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	N/A
(D5-1-2-4-1) Enable/Done=0/0: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	N/A
(D5-1-2-4-2) Enable/Done=0/0: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	N/A
(D5-1-2-4-1) Enable/Done=0/0: Read command: fail (Read/WriteLockEnabled = 1)	N/A
(D5-1-2-4-2) Enable/Done=0/0: Write command: fail (Read/WriteLockEnabled = 1)	N/A
(D5-1-2-4-3) Enable/Done=0/0: 'MBRDone' bit = 0 from Level0_Discovery	N/A
(D5-1-2-4-3(2)) Enable/Done=0/0: 'MBREnable' bit = 0 from Level0_Discovery	N/A
(D5-1-2-4-1) Enable/Done=0/1: Read command: pass (Read/WriteLockEnabled = 0)	N/A
(D5-1-2-4-2) Enable/Done=0/1: Write command: pass (Read/WriteLockEnabled = 0)	N/A
(D5-1-2-4-1) Enable/Done=0/1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	N/A
(D5-1-2-4-2) Enable/Done=0/1: Write with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	N/A
(D5-1-2-4-1) Enable/Done=0/1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	N/A
(D5-1-2-4-2) Enable/Done=0/1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	N/A
(D5-1-2-4-1) Enable/Done=0/1: Read command: fail (Read/WriteLockEnabled = 1)	N/A
(D5-1-2-4-2) Enable/Done=0/1: Write command: fail (Read/WriteLockEnabled = 1)	N/A
(D5-1-2-4-3) Enable/Done=0/1: 'MBRDone' bit = 0 from Level0_Discovery	N/A
(D5-1-2-4-3(2)) Enable/Done=0/1: 'MBREnable' bit = 0 from Level0_Discovery	N/A
(D5-1-3-1-1) Set 'Enabled' = 1 in a transaction and endTransaction status = 0; The value retains the	N/A
(D5-1-3-1-2) Set 'Enabled' = 0 in a transaction and endTransaction status = 1; The value changes back to the original value	N/A
D6: MBR.Set() Grammar and Effect	
(D6-1-1-1-1) Set data into MBR table; Response - Pass	N/A
(D6-1-1-1-1) Get data from MBR table; Compare data - Matching	N/A
(D6-1-1-1-1(2)) Read commands will retrieve MBR data - Pass	N/A
(D6-1-2-1-1) Set data to MBR table in a transaction with endTransaction status = 0; The data retains the set value	N/A
(D6-1-2-1-2) Set data to MBR table in a transaction with endTransaction status = 1; The data changes back to the original value	N/A
D7: DataStore.Set() -Basic Grammar and Effect	
(D7-1-1-1-1) Set Datastore; Response - Pass	PASS
(D7-1-1-1-1) Get Datastore and Compare data; Data - matching	PASS
(D7-1-2-1-1) Datastore.Set in a transaction with endTransaction status = 0; The data retains the set	PASS
(D7-1-2-1-2) Datastore.Set in a transaction with endTransaction status = 1; The data changes back to the original value	PASS
D8: GenKey() Effect check	
(D8-1-1-1-1) GenKey Grammar: Request with rigth parameter; Response - pass	N/A
(D8-1-2-1-1) GenKey Effect: The media encryption key used to encrypt/decrypt user data changes	N/A
(D8-1-3-1-1) GenKey Effect in a transaction with endTransaction status = 0; The range's media encryption key changes	N/A
(D8-1-3-1-2) GenKey Effect in a transaction with endTransaction status = 1; The range's media encryption key backs to the value before	N/A
D9: Activate() Effect check	
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail)	N/A
(D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled;	N/A
(D9-1-1-1-1) LockignSP.Activate() Conditon: Activate to LockingSP; Response - Pass	PASS
(D9-1-3-1-1) LockignSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 85 and all bits of word 89; 90; 92; 128 = 0	N/A
(D9-1-3-1-2) LockignSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery	PASS
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table	PASS
(D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass	PASS
(D9-1-3-1-5) LockignSP.Activate() Effect: Read data in sector 1; Compare the data - matching	PASS
(D9-1-3-2-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the	PASS
(D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change	PASS
D10: AdminSP.Revert() Effect check	
(D10-1-1-1-1) AdminSP.Revert Grammar: Revert Session to AdminSP; Revert response - pass	PASS
(D10-1-2-1-1) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be	PASS
(D10-1-2-1-2) AdminSP.Revert Effect: for ATA devices: check bit1 of word 82; bit1 of word 85; word	N/A
(D10-1-2-1-3) AdminSP.Revert Effect: LockingEnabled bit = 0 from Level0_Discovery	PASS
(D10-1-2-1-4) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured-Inactivate)	PASS
(D10-1-2-1-5) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned	PASS
(D10-1-2-1-6) AdminSP.Revert Effect: StartSession on AdminSP with MSID's PIN; SyncSession - pass	PASS
(D10-1-2-3-1) AdminSP.Revert Effect: LockingSP in inactive: Read data in sector 1; Compare the data - matching	PASS
(D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Read data in sector 1; Compare the data -	NM
(D10-1-2-2-2) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value	PASS
(D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in	N/A
D10: LockingSP.Revert() Effect check	
(D10-2-1-1-1) LockingSP.Revert Grammar: Revert Session to LockingSP; Revert response - pass	PASS
(D10-2-2-1-1) LockingSP.Revert Effect: The session remains open after issuing Locking.Revert()	PASS
(D10-2-2-1-2) LockingSP.Revert Effect: for ATA devices: check bit1 of word 82; bit1 of word 85; word	N/A
(D10-2-2-1-3) LockingSP.Revert Effect: LockingEnabled bit = 0 from Level0_Discovery	PASS
(D10-2-2-1-4) LockingSP.Revert Effect: LifeCycleState = 08h (Manufactured-Inactivate)	PASS
(D10-2-2-1-5) LockingSP.Revert Effect: StartSession on LockingSP; SyncSession - failed (Status Code: != 0 or no data returned)	PASS
(D10-2-2-3-1) LockingSP.Revert Effect: LockingSP in inactive: Read data in sector 1; Compare the data - matching	PASS
(D10-2-2-3-2) LockingSP.Revert Effect: LockingSP in inactive: Data in DataStore table shall be the	PASS
(D10-2-2-3-3) LockingSP.Revert Effect: LockingSP in inactive: Data in MBR table shall be the value in	N/A
(D10-2-2-2-1) LockingSP.Revert Effect: LockingSP in active: Read data in sector 1; Compare the data -	NM

(D10-2-2-2-2) LockingSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value	PASS
(D10-2-2-2-3) LockingSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in	N/A
D10: RevertSP() Effect check	
(D10-3-1-1-1) LockingSP.RevertSP Grammar: RevertSP wothout parameters; RevertSP Response -	PASS
(D10-3-2-1-1) LockingSP.RevertSP Condition: RevertSP if 'KeepGlobalRangeKey' = 1 and read-unlocked/write-unlocked for the Locking GlobalRange; RevertSP Response - Pass	PASS
(D10-3-2-1-1) LockingSP.RevertSP Condition: RevertSP if 'KeepGlobalRangeKey' = 1 and read-unlocked for the Locking GlobalRange; RevertSP Response - Pass	PASS
(D10-3-2-1-1) LockingSP.RevertSP Condition: RevertSP if 'KeepGlobalRangeKey' = 1 and write-unlocked for the Locking GlobalRange; RevertSP Response - Pass	PASS
(D10-3-2-1-2) LockingSP.RevertSP Condition: RevertSP if 'KeepGlobalRangeKey' = 1 and read and write-locked for the Locking GlobalRange; RevertSP Response - Fail(3Fh)	PASS
(D10-3-3-1-1) LockingSP.RevertSP Effect: The session shall be aborted: Get_Rqs for LifeCycleState after RevertSP() is successful; Get_Rsp - no data returned	PASS
(D10-3-3-1-2) LockingSP.RevertSP Effect: for ATA devices: check bit1 of word 82; bit1 of word 85; word 89; 90; 128	N/A
(D10-3-3-1-3) LockingSP.RevertSP Effect: LockingEnabled bit = 0 from Level0_Discovery	PASS
(D10-3-3-1-4) LockingSP.RevertSP Effect: LifeCycleState = 08h (Manufactured-Inactivate)	PASS
(D10-3-3-1-5) LockingSP.RevertSP Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned	PASS
(D10-3-3-2-1) LockingSP.RevertSP Effect: LockingSP in active: KeepGKey=1 and data covered by GlobalRange; Data shall not change	PASS
(D10-3-3-2-3) LockingSP.RevertSP Effect: LockingSP in active; KeepGKey=omitted; Data shall not	NM
(D10-3-3-2-4) LockingSP.RevertSP Effect: LockingSP in active: Data in DataStore table shall be the	PASS
(D10-3-3-2-5) LockingSP.RevertSP Effect: LockingSP in active: Data in MBR table shall be the value in	N/A
D9-D10 Activate and Revert: ATA command check in RestrictedCommands table	
(D9-1-3-1-6) RestrictedCmds: ATA command check after LockingSP.Activate	N/A
(D10-3-3-1-6) RestrictedCmds: ATA command check after LockingSP.RevertSP	N/A
(D10-2-2-1-6) RestrictedCmds: ATA command check after LockingSP.Revert	N/A
(D10-1-2-1-7) RestrictedCmds: ATA command check after AdminSP.Revert	N/A
D11: Power Cycle	
(D11-1-1-1-1) C_PIN: after power cycle 1.if Persistence=1 Tries=no change ; 2.if Persistence=0	PASS
(D11-2-1-1-1) RestrictedCmds: Next()/Get() to get the cell contents	N/A
(D11-2-1-1-1) RestrictedCmds: Allowed column check after power cycle	N/A
Revert LockingSP	
Start Session with HostChallenge - AdminSP	PASS
Sync Session - AdminSP	PASS
LockingSP.Revert - Request	PASS
LockingSP.Revert - Response	PASS
End Session - Request	PASS
End Session - Response	PASS
** OPAL v2.0 - Generic **	
Revert LockingSP	
Start Session with HostChallenge - AdminSP	PASS
Sync Session - AdminSP	PASS
LockingSP.Revert - Request	PASS
LockingSP.Revert - Response	PASS
End Session - Request	PASS
End Session - Response	PASS
Protocol 2 Command Test	
Check Get_ComID command	N/A
Check Verify_ComID_Valid command	N/A
Check Get_ComID_Rsp command	N/A
Check SSC information	
Identify the device type from the TPerInfo table	PASS
Check the support of OPAL SSC v2.00	N/A
Verify Geometry infornmation	
Geometry Reporting Feature returned from Level0_Discovery	N/A
Contents of column 07-0Ah returned from the LockingInfo table	PASS
Verify Geometry Info between LockingInfo table and Level0_Discovery	N/A
TPer Reset Command Test	
Check the support of TPer_Reset command	PASS
If TPer_Reset is disabled; Issue TPer_Reset - aborted	PASS
Enable TPer_Reset command: set ProgrammaticResetEnable=1 in the TPerInfo table	PASS
All open session SHALL be aborted on all ComID	PASS
All uncommitted transactions SHAll be aborted on all ComID	PASS
The synchronous protocol stack for all ComID SHAll be reset to its initial state	PASS
All related method processing occurring on all ComIDs SHALL be aborted	PASS
Host's communications capabilities SHAll be reset to the initial minimum assumptions	PASS
Read/WriteLocked = True for all Locking objects if the LockOnReset = Programmatic enumeration	PASS
Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration value	N/A
Check Read/WriteLocked for all Locking objects before and after TPer_Reset is disabled	PASS
Check Done in the MBRControl table before and after TPer_Reset is disabled	N/A
Stack Reset Test	
Check the support of Stack_Reset command	PASS
The data returned from Stack_Reset rsponse - Get_ComID_Rsp	PASS
All open session for that ComID SHALL be aborted	PASS
All uncommitted transactions SHALL be aborted	PASS
All related method on that ComID SHALL be aborted	PASS
The protocol stack for all ComIDs SHALL be reset to its initial state	PASS
All communications properties SHALL be reset to their default values	PASS
No Response Available if no handle_ComID_Request command preceded the Get_ComID_Response	PASS
Check 'ReadLocked' and 'WriteLocked' values in Locking table	PASS
Check 'Done' value in MBRControl table	N/A
Stack_Reset with non-zero reserved byte; It shall be ignored by both host and device	PASS
Revert LockingSP	
Start Session with HostChallenge - AdminSP	PASS
Sync Session - AdminSP	PASS
LockingSP.Revert - Request	PASS
LockingSP.Revert - Response	PASS
End Session - Request	PASS
End Session - Response	PASS
Activating the Locking SP	
Start Session with HostChallenge - AdminSP	PASS
Sync Session - AdminSP	PASS
Activate_LockingSP	PASS
Activate_LockingSP - Response	PASS
Get - LifeCycle(Locking SP) - Request	PASS
Get - LifeCycle(Locking SP) - Response	PASS

Check the state of LockingSP	PASS
End Session - Request	PASS
End Session - Response	PASS
Check Authenticate method	
Check the support of Authenticate method - AdminSP	PASS
Authenticate - SID; Authenticate Response - Success(AuthStatus = 01h)	PASS
Authenticate - Admin1(non-authorized UID); Authenticate Response - Fail(AuthStatus = 0h)	N/A
Check the support of Authenticate method - LockingSP	PASS
Authenticate - Admin1; Authenticate Response - Success(AuthStatus = 01h)	PASS
Authenticate - User1 (authority UID); Authenticate Response - Success(AuthStatus = 01h)	PASS
Authenticate - User2 (non-authority UID); Authenticate Response - Fail(AuthStatus = 0h)	PASS
Number of authenticate attempts > MaxAuthentications; Authenticate Response - Fail(AuthStatus =	PASS
Authenticate - UserX (invalid-authority UID); Authenticate Response - StatusCode =	PASS
Authenticate - User1 with incorrect optional param; Authenticate Response - StatusCode = 0Ch(Invalid_Param)	PASS
Check Random method	
Check the support of Random method - AdminSP	PASS
Random Request with count < 20h in AdminSP; Random Response - Success	PASS
Random Request with count = 20h in AdminSP; Random Response - Success	PASS
Random Request with count > 20h in AdminSP; Random Response - Success or StatusCode=0Ch(Invalid_Param)	PASS
Check the support of Random method - LockingSP	PASS
Random Request with count < 20h in LockingSP; Random Response - Success	PASS
Random Request with count = 20h in LockingSP; Random Response - Success	PASS
Random Request with count > 20h in LockingSP; Random Response - Success or StatusCode=0Ch(Invalid_Param)	PASS
Alignment LBA Test	
RangeStart/Length: Aligned; Response - Pass	N/A
RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams)	N/A
RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams)	N/A
RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams)	N/A
Data Alignment Restriction on Byte Table - DataStore	
Get MandatoryWriteGranularity and RecommendedAccessGranularity of DataStore from Table table	PASS
MandatoryWriteGranularity of DataStore SHALL be less than or equal to 8192	PASS
Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param)	N/A
Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param)	N/A
Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into DataStore table; Response - Pass	PASS
Get and Compare data from DataStore - Matching	PASS
Data Alignment Restriction on Byte Table - MBR	
Get MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table	N/A
MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192	N/A
Set data(lengthMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param)	N/A
Set data(offsetMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param)	N/A
Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into MBR table; Response - Pass	N/A
Get and Compare data from MBR table - Matching	N/A
AdminSP.Revert() Effect check	
AdminSP.Revert with 'Behavior of C_PIN_SID PIN on TPer Revert'=0 or 1: Revert response - pass	PASS
'Behavior of C_PIN_SID PIN'=0: PIN = C_PIN_MSID and 'Initial C_PIN_SID'=0	PASS
RevertSP() Grammar check for Pyrite 1.00	NC
KeepGlbRange: RevertSP to LockingSP with 'KeepGlobalRangeKey'=0; RevertSP response - Status Code: 0Ch(Invalid Param)	NC
KeepGlbRange: RevertSP to LockingSP with 'KeepGlobalRangeKey'=1; RevertSP response - Status Code: 0Ch(Invalid Param)	NC
Revert LockingSP	
Start Session with HostChallenge - AdminSP	PASS
Sync Session - AdminSP	PASS
LockingSP.Revert - Request	PASS
LockingSP.Revert - Response	PASS
End Session - Request	PASS
End Session - Response	PASS
** OPAL v2.0 - Table Contents **	
C3: Get() Object Table Contents to AdminSP	
(C3-1) Table: Next() method for table (AdminSP)	PASS
(C3-1) Table: Get the entries from table (AdminSP)	PASS
(C3-1) Table: Verify the table contents (AdminSP)	PASS
(C3-2) SPInfo: Get the entries from table (AdminSP)	PASS
(C3-2) SPInfo: Verify the table contents (AdminSP)	PASS
(C3-3) SPTemplates: Next() method for table (AdminSP)	PASS
(C3-3) SPTemplates: Get the entries from table (AdminSP)	PASS
(C3-3) SPTemplates: Verify the table contents (AdminSP)	PASS
(C3-4) MethodID: Next() method for table (AdminSP)	PASS
(C3-4) MethodID: Get the entries from table (AdminSP)	PASS
(C3-4) MethodID: Verify the table contents (AdminSP)	PASS
(C3-6) ACE: Next() method for table (AdminSP)	PASS
(C3-6) ACE: Get the entries from table (AdminSP)	PASS
(C3-6) ACE: Verify the table contents (AdminSP)	PASS
(C3-7) Authority: Next() method for table (AdminSP)	PASS
(C3-7) Authority: Get the entries from table (AdminSP)	PASS
(C3-7) Authority: Verify the table contents (AdminSP)	PASS
(C3-8) C_PIN: Next() method for table (AdminSP)	PASS
(C3-8) C_PIN: Get the entries from table (AdminSP)	PASS
(C3-8) C_PIN: Verify the table contents (AdminSP)	PASS
(C3-9) TPerInfo: Get the entries from table (AdminSP)	PASS
(C3-9) TPerInfo: Verify the table contents (AdminSP)	PASS
(C3-10) Template: Next() method for table (AdminSP)	PASS
(C3-10) Template: Get the entries from table (AdminSP)	PASS
(C3-10) Template: Verify the table contents (AdminSP)	PASS
(C3-11) SP: Next() method for table (AdminSP)	PASS
(C3-11) SP: Get the entries from table (AdminSP)	PASS
(C3-11) SP: Verify the table contents (AdminSP)	PASS
Activating the Locking SP	
Start Session with HostChallenge - AdminSP	PASS
Sync Session - AdminSP	PASS
Activate_LockingSP	PASS
Activate_LockingSP - Response	PASS
Get - LifeCycle(Locking SP) - Request	PASS
Get - LifeCycle(Locking SP) - Response	PASS

Check the state of LockingSP	PASS
End Session - Request	PASS
End Session - Response	PASS
C3: Get() Object Table Contents to LockingSP	
(C3-12) Table: Next() method for table (LockingSP)	PASS
(C3-12) Table: Get the entries from table (LockingSP)	PASS
(C3-12) Table: Verify the table contents (LockingSP)	PASS
(C3-13) SPInfo: Get the entries from table (LockingSP)	PASS
(C3-13) SPInfo: Verify the table contents (LockingSP)	PASS
(C3-14) SPTemplates: Next() method for table (LockingSP)	PASS
(C3-14) SPTemplates: Get the entries from table (LockingSP)	PASS
(C3-14) SPTemplates: Verify the table contents (LockingSP)	PASS
(C3-16) MethodID: Next() method for table (LockingSP)	PASS
(C3-16) MethodID: Get the entries from table (LockingSP)	PASS
(C3-16) MethodID: Verify the table contents (LockingSP)	PASS
(C3-18) ACE: Next() method for table (LockingSP)	PASS
(C3-18) ACE: Get the entries from table (LockingSP)	PASS
(C3-18) ACE: Verify the table contents (LockingSP)	PASS
(C3-19) Authority: Next() method for table (LockingSP)	PASS
(C3-19) Authority: Get the entries from table (LockingSP)	PASS
(C3-19) Authority: Verify the table contents (LockingSP)	PASS
(C3-20) C_PIN: Next() method for table (LockingSP)	PASS
(C3-20) C_PIN: Get the entries from table (LockingSP)	PASS
(C3-20) C_PIN: Verify the table contents (LockingSP)	PASS
(C3-21) LockingInfo: Get the entries from table (LockingSP)	PASS
(C3-21) LockingInfo: Verify the table contents (LockingSP)	PASS
(C3-22) Locking: Next() method for table (LockingSP)	PASS
(C3-22) Locking: Get the entries from table (LockingSP)	PASS
(C3-22) Locking: Verify the table contents (LockingSP)	PASS
(C3-23) MBRControl: Get the entries from table (LockingSP)	N/A
(C3-23) MBRControl: Verify the table contents (LockingSP)	N/A
(C3-) SecretProtect: Next() method for table (LockingSP)	N/A
(C3-) SecretProtect: Get the entries from table (LockingSP)	N/A
(C3-) SecretProtect: Verify the table contents (LockingSP)	N/A
(C3-26) K_AES: Next() method for table (LockingSP)	N/A
(C3-26) K_AES: Get the entries from table (LockingSP)	N/A
(C3-26) K_AES: Verify the table contents (LockingSP)	N/A
(C3-27) RestrictedCmds: Next() method for table (LockingSP)	N/A
(C3-27) RestrictedCmds: Get the entries from table (LockingSP)	N/A
(C3-27) RestrictedCmds: Verify the table contents (LockingSP)	N/A
C5: GetACL() Table Contents (AdminSP)	
(C5-1) Next() - Table Table	PASS
(C5-1) GetACL() - Table Table	PASS
(C5-1) Verify ACL values for Table Table	PASS
(C5-2) GetACL() - SPInfo Table	PASS
(C5-2) Verify ACL values for SPInfo Table	PASS
(C5-3) Next() - SPTemplates Table	PASS
(C5-3) GetACL() - SPTemplates Table	PASS
(C5-3) Verify ACL values for SPTemplates Table	PASS
(C5-4) Next() - MethodID Table	PASS
(C5-4) GetACL() - MethodID Table	PASS
(C5-4) Verify ACL values for MethodID Table	PASS
(C5-5) Next() - ACE Table	PASS
(C5-5) GetACL() - ACE Table	PASS
(C5-5) Verify ACL values for ACE Table	PASS
(C5-6) Next() - Authority Table	PASS
(C5-6) GetACL() - Authority Table	PASS
(C5-6) Verify ACL values for Authority Table	PASS
(C5-7) Next() - C_PIN Table	PASS
(C5-7) GetACL() - C_PIN Table	PASS
(C5-7) Verify ACL values for C_PIN Table	PASS
(C5-8) GetACL() - TPerInfo Table	PASS
(C5-8) Verify ACL values for TPerInfo Table	PASS
(C5-9) Next() - Template Table	PASS
(C5-9) GetACL() - Template Table	PASS
(C5-9) Verify ACL values for Template Table	PASS
(C5-10) Next() - SP Table	PASS
(C5-10) GetACL() - SP Table	PASS
(C5-10) Verify ACL values for SP Table	PASS
C5: GetACL() Table Contents (LockingSP)	
(C5-11) Next() - Table Table	PASS
(C5-11) GetACL() - Table Table	PASS
(C5-11) Verify ACL values for Table Table	PASS
(C5-12) GetACL() - SPInfo Table	PASS
(C5-12) Verify ACL values for SPinfo Table	PASS
(C5-13) Next() - SPTemplates Table	PASS
(C5-13) GetACL() - SPTemplates Table	PASS
(C5-13) Verify ACL values for SPTemplates Table	PASS
(C5-15) Next() - MethodID Table	PASS
(C5-15) GetACL() - MethodID Table	PASS
(C5-15) Verify ACL values for MethodID Table	PASS
(C5-16) Next() - ACE Table	PASS
(C5-16) GetACL() - ACE Table	PASS
(C5-16) Verify ACL values for ACE Table	PASS
(C5-17) Next() - Authority Table	PASS
(C5-17) GetACL() - Authority Table	PASS
(C5-17) Verify ACL values for Authority Table	PASS
(C5-18) Next() - C_PIN Table	PASS
(C5-18) GetACL() - C_PIN Table	PASS
(C5-18) Verify ACL values for C_PIN Table	PASS
(C5-19) GetACL() - LockingInfo Table	PASS
(C5-19) Verify ACL values for LockingInfo Table	PASS
(C5-20) Next() - Locking Table	PASS
(C5-20) GetACL() - Locking Table	PASS
(C5-20) Verify ACL values for Locking Table	PASS
(C5-21) GetACL() - MBRControl Table	N/A
(C5-21) Verify ACL values for MBRControl Table	N/A
(C5-22) GetACL() - MBR Table	N/A
(C5-22) Verify ACL values for MBR Table	N/A

(C5-23) GetACL() - K_AES_128/256 Table	N/A
(C5-23) Verify ACL values for K_AES_128/256 Table	N/A
(C5-24) GetACL() - DataStore Table	PASS
(C5-24) Verify ACL values for DataStore Table	PASS
(C5-25) GetACL() - SP Table	PASS
(C5-25) Verify ACL values for SP Table	PASS
(C5-) Next() - SecretProtect Table	N/A
(C5-) GetACL() - SecretProtect Table	N/A
(C5-) Verify ACL values for SecretProtect Table	N/A
(C5-26) Next() - RestrictedCmds Table	N/A
(C5-26) GetACL() - RestrictedCmds Table	N/A
Revert LockingSP	
Start Session with HostChallenge - AdminSP	PASS
Sync Session - AdminSP	PASS
LockingSP.Revert - Request	PASS
LockingSP.Revert - Response	PASS
End Session - Request	PASS
End Session - Response	PASS
Activating the Locking SP	
Start Session with HostChallenge - AdminSP	PASS
Sync Session - AdminSP	PASS
Activate_LockingSP	PASS
Activate_LockingSP - Response	PASS
Get - LifeCycle(Locking SP) - Request	PASS
Get - LifeCycle(Locking SP) - Response	PASS
Check the state of LockingSP	PASS
End Session - Request	PASS
End Session - Response	PASS
** OPAL v2.0 - Feature Set **	
Opal SSC new Feature Set: Additional DataStore Tables	
Check the feature support of Additional DataStore from Level0_Discovery	N/A
Compare the number of Additional DataStore in Table table and maximum number from	N/A
Check the new entries added to the AccessControl table	N/A
Activate() method with all DataStore table; Response - Pass	N/A
Activate() method with DataStore size (<= maxDSSize); Response - Pass	N/A
Activate() method with DataStore size (> maxDSSize); Response -	N/A
Activate() method with non-align DataStore; Response - StatusCode=0Ch(Invalid_Param)	N/A
Activate() method without dataStoreList; Response - Pass	N/A
Activate():The size of dataStore is equal to the 'Maximum total size of DataStore' from	N/A
ReActivate() method with all DataStore table; Response - Pass	N/A
Reactivate() method with DataStore size (<= maxDSSize); Response - Pass	N/A
Reactivate() method with DataStore size (> maxDsSize); Response -	N/A
Reactivate() method with non-align DataStore; Response - StatusCode=0Ch(Invalid_Param)	N/A
ReActivate() method without dataStoreList; Response - Pass	N/A
ReActivate():The size of dataStore is equal to the 'Maximum total size of DataStore' from	N/A
Opal SSC new Feature Set: Single User Mode	
Check the feature support of Single User Mode from Level0_Discovery	N/A
Check the support of ReActivate and Erase methods in the MethodID table	N/A
Get the values of 'SingleUserModeRange' and 'RangeStartLenPolicy' from the LockingInfo table	N/A
Activate() method with SP not included in Locking Template; Response -	N/A
Activate() method with LockingObject not included in Locking table; Response - StatusCode=0Ch(Invalid_Param)	N/A
Activate() method with RangeN(N=LockingInfo.MaxRanges/2); Response - Pass	N/A
Verify: StartSession to Locking SP as UserN(N=MaxRanges/2); SyncSession -	N/A
Verify: StartSession to Locking SP as User(N+1)(N=MaxRanges/2); SyncSession - Pass	N/A
Activate() method with RangeN(N=LockingInfo.MaxRanges); Response - Pass	N/A
Verify: StartSession to Locking SP as UserN(N=MaxRanges); SyncSession -	N/A
Verify: StartSession to Locking SP as User(N+1)(N=MaxRanges); SyncSession - Pass	N/A
Activate() method with empty ObjList and 'RangeStartLenPolicy'=0; Response - pass	N/A
Verify 'SingleUserModeRange'=empty and 'RangeStartLenPolicy'=1 from the LockingInfo table	N/A
Verify 'Policy'=1; 'All'=0; 'Any'=0 from Level0_Discovery	N/A
Activate() method with empty ObjList and 'RangeStartLenPolicy'=1; Response - pass	N/A
Verify 'SingleUserModeRange'=empty and 'RangeStartLenPolicy'=1 from the LockingInfo table	N/A
Verify 'Policy'=1; 'All'=0; 'Any'=0 from Level0_Discovery	N/A
Activate() method with SingleUserMode for Range1 and Range2 if LockingSP = mfg state; Response -	N/A
The method shall have no effect: 'SingleUserModeRange' and 'RangeStartLenPolicy' keep the	N/A
Activate() method with SingleUserMode for Range1 and Range2 after LockingSP.Revert; Response -	N/A
Verify 'SingleUserModeRange'=Range1/Range2 and 'RangeStartLenPolicy'=0 from the LockingInfo	N/A
Verify 'Policy'=0; 'All'=0; 'Any'=1 from Level0_Discovery	N/A
Locking_Range1.Set Request in LockingSP as User2; Response - Pass	N/A
Locking_Range1.Set Request in LockingSP as Admin1; Response - StatusCode = 01h(Not_Authorized)	N/A
Activate() method with entire Locking table and 'RangeStartLenPolicy'=0; Response - Pass	N/A
Activate w/ entireLocking: Verify 'SingleUserModeRange'=EntireLocking and 'RangeStartLenPolicy'=0 from the LockingInfo table	N/A
Activate w/ entireLocking: Verify 'Policy'=0; 'All'=1; 'Any'=1 from Level0_Discovery	N/A
Activate w/ entireLocking: Range1-GlobalRange.Set Request in LockingSP as User1-(N+1); Response - StatusCode = 01h(Not_Authorized)	N/A
Activate w/ entireLocking: GlobalRange-RangeN.Set Request in LockingSP as User1-(N+1); Response -	N/A
Activate() method with all Locking Objects and 'RangeStartLenPolicy'=0; Response - Pass	N/A
Activate w/ allLockingObj: Verify 'SingleUserModeRange'=all objects and 'RangeStartLenPolicy'=0 from the LockingInfo table	N/A
Activate w/ allLockingObj: Verify 'Policy'=0; 'All'=1; 'Any'=1 from Level0_Discovery	N/A
Activate w/ allLockingObj: Range1-GlobalRange.Set Request in LockingSP as User1-(N+1); Response - StatusCode = 01h(Not_Authorized)	N/A
Activate w/ allLockingObj: GlobalRange-RangeN.Set Request in LockingSP as User1-(N+1); Response -	N/A
ReActivate() method with Read/WriteLockEnabled=True; Response - StatusCode=3Fh(Fail)	N/A
ReActivate() method with ReadLockEnabled=True; Response - StatusCode=3Fh(Fail)	N/A
ReActivate() method with WriteLockEnabled=True; Response - StatusCode=3Fh(Fail)	N/A
ReActivate() method with LockingObject not included in Locking table; Response - StatusCode=0Ch(Invalid_Param)	N/A
ReActivate() method with RangeN(N=LockingInfo.MaxRanges/2); Response - Pass	N/A
Verify: StartSession to Locking SP as UserN(N=MaxRanges/2); SyncSession -	N/A
Verify: StartSession to Locking SP as User(N+1)(N=MaxRanges/2); SyncSession - Pass	N/A
ReActivate() method with RangeN(N=LockingInfo.MaxRanges); Response - Pass	N/A
Verify: StartSession to Locking SP as UserN(N=MaxRanges); SyncSession -	N/A
Verify: StartSession to Locking SP as User(N+1)(N=MaxRanges); SyncSession - Pass	N/A
ReActivate() with Admin1PIN=omitted - Succeed	N/A
ReActivate() effect: The session - Abort (no data returned)	N/A
ReActivate() effect: The LifeCycleState of the LockingSP remains the same	N/A

ReActivate() effect: The value of 'C_PIN_Admin1.PIN' remains at their current values	N/A
ReActivate() effect: RangeStart and RangeLength remain at their current values	N/A
ReActivate() effect: The media encryption keys remain at their current values	N/A
ReActivate() with Admin1PIN; Response - Pass	N/A
Start a session to LockingSP as Admin1 with new Admin1PIN; Response - Pass	N/A
ReActivate() method with empty ObjList and 'RangeStartLenPolicy'=0; Response - pass	N/A
Verify 'SingleUserModeRange'=empty and 'RangeStartLenPolicy'=1 from the LockingInfo table	N/A
Verify 'Policy'=1; 'All'=0; 'Any'=0 from Level0_Discovery	N/A
ReActivate() method with empty ObjList and 'RangeStartLenPolicy'=1; Response - pass	N/A
Verify 'SingleUserModeRange'=empty and 'RangeStartLenPolicy'=1 from the LockingInfo table	N/A
Verify 'Policy'=1; 'All'=0; 'Any'=0 from Level0_Discovery	N/A
ReActivate() method with SingleUserMode for Range1 and Ragne2; Response - Pass	N/A
Verify 'SingleUserModeRange' and 'RangeStartLenPolicy' from the LockingInfo table	N/A
Verify 'Policy'=0; 'All'=0; 'Any'=1 from Level0_Discovery	N/A
ReActivate() method with entire Locking table and 'RangeStartLenPolicy'=0; Response - Pass	N/A
ReActivate w/ entireLocking: Verify 'SingleUserModeRange'=EntireLocking and 'RangeStartLenPolicy'=0 from the LockingInfo table	N/A
ReActivate w/ entireLocking: Verify 'Policy'=0; 'All'=1; 'Any'=1 from Level0_Discovery	N/A
ReActivate w/ entireLocking: Range1-GlobalRange.Set Request in LockingSP as User1-(N+1); Response - StatusCode = 01h(Not_Authorized)	N/A
ReActivate w/ entireLocking: GlobalRange-RangeN.Set Request in LockingSP as User1-(N+1);	N/A
ReActivate() method with all Locking Objects and 'RangeStartLenPolicy'=0; Response - Pass	N/A
ReActivate w/ allLockingObj: Verify 'SingleUserModeRange'=all objects and 'RangeStartLenPolicy'=0 from the LockingInfo table	N/A
ReActivate w/ allLockingObj: Verify 'Policy'=0; 'All'=1; 'Any'=1 from Level0_Discovery	N/A
ReActivate w/ allLockingObj: Range1-GlobalRange.Set Request in LockingSP as User1-(N+1); Response - StatusCode = 01h(Not_Authorized)	N/A
ReActivate w/ allLockingObj: GlobalRange-RangeN.Set Request in LockingSP as User1-(N+1);	N/A
Set a new PIN to userX Request; Response - Pass	N/A
Erase() effect: Locking_Range(X-1).Erase Request; Response - Pass	N/A
Erase() effect: Read/WriteLockEnabled and Read/WriteLocked = 0	N/A
Erase() effect: RangeStart and RangeLength are not changed	N/A
Erase() effect: Generate a new media encrypion key for LBA range	N/A
Erase(): C_PIN.UserX = empty	N/A
Erase(): Tries = 0 from the C_PIN table	N/A
New Feature Set: Block SID Authentication	
Check the support of BlockSID Authentication from Level0_Discovery	PASS
Block SID Authentication command: pass/abort(the command is supported/not supported)	PASS
Check SID Blocked State after Block SID Authentication command: SID Blocked State = 1	PASS
Start Session as SID after successful execution of Block SID Authentication command:	PASS
Authenticate - SID (authority UID); Authenticate Response -	PASS
The Tries column of the SID C_PIN shall not be incremented after Block SID Authentication	N/A
Clear Events: Revert AdminSP	N/A
Check SID Blocked State(=0) after Revert	N/A
Clear Events: Power Cycle	PASS
Check SID Blocked State(=0) after power cycle	PASS
Block SID Authentication command with Hardware Reset bit=1: Pass	PASS
Check SID Blocked State(=0) after Hardware Reset	PASS
Subsequent invocation of Block SID Authentication command: Fail with 'Other Invalid Command	PASS
Revert LockingSP	
Start Session with HostChallenge - AdminSP	PASS
Sync Session - AdminSP	PASS
LockingSP.Revert - Request	PASS
LockingSP.Revert - Response	PASS
End Session - Request	PASS
End Session - Response	PASS