ULINK_Test Reporter (TR v4.3.6)

Application Key Test Suite Test Version Device Id Capacity Tested Passed Failed Warned Not tested Passed with Waiver Start Date Start Time End Date

End Time Test Time OS Type Software Version Model Number Serial Number Firmware Revision Total LBA Device Interface

Path

CSV Name License By 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) 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) XferLength: Trusted Send with SP=01h; Spcf=ComID; Xfer=00h: Abort N/A (A1-1-1-2) 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: Ahort 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 (A2-1-1-1-1) Spcf=0 DataXfer: TCG-Receive with SP=00h; Spcf=00h; Xfer=01h: Pass 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 (A2-2-1-1-1) Spcf=1 DataXfer: TCG-Receive with SP=00h; Spcf=01h; Xfer=01h: Pass 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) 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) IF_Send: TPer in awaiting IF_Send state after Power-on reset - IF_Send with SP=01h; Spcf=ComID; Xfer=01h: pass PASS (A4-1-1-3) 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; PASS Spcf=01h; Xfer=01h: pass

[©2022 ULINK Technology Inc.] Date : Fri Jul 08 01:02:59 UTC 2022 Value 1 ULINK TCG/I1667 - OPAL Script REV 7.5 PC601 NVME SK HYNIX 512GB_SI8BQ02611010BQBA 512 GB 805 805 323 8/16/2019 3:07:54 PM 8/16/2019 3-28-58 PM 0:21:03 x64 7.1.1300 PC601 NVME SK HYNIX 512GB SI8BQ02611010BQBA 80000111 1000215216 NVMEDRV+NET

0

0

/UTL/Log/Ptc TCG PC601NVMeSKhynix512GB SI8BQ02611010BQBA 01/

Ptc_TCG_PC601NVMeSKhynix512GB_SI8BQ02611010BQBA_01.csv ULINK

(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 (A4-2-3-2-3) IF_Recv: TPer in awaiting IF_Recv state - IF_Recv with Xfer=insufficient; TPer stays in	PASS
awaiting JF_Recv state A5: Check ComPacket/Packet/SubPacket	PASS
(A5-1-1-2) IF_Send ComPacket - Reserved field != 0; IF_Send: pass (A5-1-2-2-2) IF_Send ComPacket - ComID != current ID; TPer in awaiting IF_Send state	PASS 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 (A5-1-4-1-2) IF_Send ComPacket - MinTransfer!= 0; IF_Send: pass	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 (A5-1-5-1-3) IF_Send ComPacket - Padding byte != 0; IF_Send: pass	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 (A5-2-6-1-2(2)) IF_Send Packet - Length < 12 of Subpacket; no data returned	PASS PASS
(A5-3-1-1-1(1)) IF_Send Packet - non-aligned with 4 byte in the start point of Subpacket; no data	1 435
returned (A5-3-1-1-2) IF_Send SubPacket - Reserved field != 0; IF_Send: pass	N/A 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	PASS
abortion	
(A5-2-6-1-2(2)) IF_Send Packet in regular session - Length < 12 of Subpacket; Session abortion (A5-3-3-1-2) IF_Send SubPacket - Length > Packet; Session abortion	PASS PASS
A7: Transaction check	
(A7-1-1-2-1(2)) StartTransaction Request: status != 0; StartTransaction Response: Pass with status = 0	PASS
(A7-1-12-1(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 (A7-1-2-2-1(1)) EndTransaction Request: status = 0 with short atom(81h); EndTransaction Response:	
Pass (A7-1-2-2-1(1)) EndTransaction Request: status = 0 with medium atom(C001h); EndTransaction	PASS
Response: Pass (A7-1-2-2-1(1)) EndTransaction Request: status = 0 with long atom(E0000001h); EndTransaction	PASS
Response: Pass (A7-1-2-2-1(2)) EndTransaction Request: status != 0 with short atom(81h); EndTransaction	PASS
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 (A7-1-3-1-1) Trans-Start attempt: StartTransaction <= MaxTransLimit; Response: pass	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
(A7-3-2-2) 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 (A7-1-3-3-2) Trans-Abort attempt: EndTransaction Request: within a transaction with status = 0;	PASS
Response (abort): pass (A7-1-3-4-1) Stand-Alone: StartTransaction Request: only with Start_Trans token and status token;	PASS
Response: Pass (A7-1-3-5-1) Stand-Alone: EndTransaction Request: only with End_Trans token and status token;	PASS
Response: Pass	PASS
(A7-1-3-6-1) Multiple Trans: Trans-Start request after one or more; Response: Pass (A7-1-3-8-1) Trans-attempt in a CtrlSession: Trans-Start request outside of methodInvoke: Token	PASS
shall be discarded (A7-1-3-8-1) Trans-attempt in a CtrlSession: Trans-End request outside of methodInvoke: Token	PASS
shall be discarded	PASS
(A7-1-6-1-1) Trans+Session Abort: Transaction is aborted after session gets aborted (A7-1-7-1-1) Trans+Session Close: Transaction is aborted after session gets closed	PASS PASS
(A7-1-8-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session	PASS
(A8-1-1-1) 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
pro 2 - 2 2, session arter Endsession, start a new session shan pass after the session closed	

(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 (A8-3-2-1-1) Session Timeout: If session# = MaxSessions and a session is timeout; Start/Sync Session - pass	PASS PASS
·	PASS
(A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (A9-1-1-1) StartSession - Y0xFF' 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 (A9-1-1-5-1) StartSession - '0xFF' among HostSID and SPUID parameters; SyncSession: pass	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;	PASS
Response: pass (A9-1-1-13-1) EndTransaction - '0xFF' between a TransactionEnd token and the status code;	PASS
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 (A9-1-2-1-1) StartTransaction - Empty atoms in plural places; SyncSession: pass	PASS PASS
A10: Set Properties test	PASS
(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 (A10-1-6-3-1) Optional Params: Check Host Properties - MaxComPacketSize = 800h: Response value	PASS
= 800h (A10-1-6-5-1) Optional Params: Check Host Properties - MaxPacketSize < 7ECh: Response value =	PASS
7ECh (A10-1-6-5-1) Optional Params: Check Host Properties - MaxPacketSize = 7ECh: Response value =	PASS
7ECh (A10-1-6-6-1) Optional Params: Check Host Properties - MaxIndTokenSize < 7C8h: Response value =	PASS
7C8h (A10-1-6-6-1) Optional Params: Check Host Properties - MaxIndTokenSize = 7C8h: Response value =	PASS
7C8h (A10-1-6-7-1) Optional Params: Check Host Properties - MaxPackets = a number: Response value <=	PASS 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 (A10-1-6-15-1) Optional Params: Check Host Properties - Omission of HostParams: no HostParams	PASS PASS
	1 7.55
A10: Properties response and effect test (A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data >	
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize;	PASS
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h	
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort	PASS PASS PASS
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: = TPer's MaxComPacketSize; Response: Pass	PASS PASS
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: = TPer's MaxComPacketSize; Response: Pass (A10-3-2-1-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response:	PASS PASS PASS PASS
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: = TPer's MaxComPacketSize; Response: Pass	PASS PASS PASS
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: = TPer's MaxComPacketSize; Response: Pass (A10-3-2-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: ST = 51h at ATA interface level	PASS PASS PASS PASS PASS N/A
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: = TPer's MaxComPacketSize; Response: Pass (A10-3-2-1-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: S = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxPacketSize; Response: (A10-3-2-4-1) Properties Effect - TPerProp in regular session: > TPer's MaxIndTokenSize; Response: (A10-3-2-4-1) Properties Effect - TPerProp in regular session: > TPer's MaxIndTokenSize; Response: (A10-3-2-4-1) Properties Effect - TPerProp in regular session: > TPer's MaxIndTokenSize; Response:	PASS PASS PASS PASS PASS N/A
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxPacketSize; Response: Session abort Properties Effect - TPerProp in control session: = TPer's MaxComPacket; Response - Pass	PASS PASS PASS PASS PASS N/A
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: = TPer's MaxComPacketSize; Response: Pass (A10-3-2-1-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: S = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxPacketSize; Response: (A10-3-2-4-1) Properties Effect - TPerProp in regular session: > TPer's MaxIndTokenSize; Response: (A10-3-2-4-1) Properties Effect - TPerProp in regular session: > TPer's MaxIndTokenSize; Response: (A10-3-2-4-1) Properties Effect - TPerProp in regular session: > TPer's MaxIndTokenSize; Response:	PASS PASS PASS PASS PASS N/A
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: = TPer's MaxComPacketSize; Response: Pass (A10-3-2-1-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: ST = 51h at ATA interface level (A10-3-2-4-1) Properties Effect - TPerProp in regular session: > TPer's MaxPacketSize; Response: (A10-3-2-4-1) Properties Effect - TPerProp in regular session: > TPer's MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in control session: = TPer's MaxComPackets; Response - Pass (A10-3-2-1-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: ST	PASS PASS PASS PASS N/A N/A PASS
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: = TPer's MaxComPacketSize; Response: Pass (A10-3-2-1-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: ST = 51h at ATA interface level (A10-3-2-4-1) Properties Effect - TPerProp in regular session: > TPer's MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in control session: = TPer's MaxComPacketS; Response: ST = 51h at ATA interface level (A10-3-2-1-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: ST = 51h at ATA interface level (A10-3-2-1-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: ST = 51h at ATA interface level (A10-3-2-1-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxPacketS; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxPacketS; Response: ST	PASS PASS PASS PASS N/A N/A PASS
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: Pass (A10-3-2-1-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: S = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxCamPacketSize; Response: Session abort Properties Effect - TPerProp in cegular session: > TPer's MaxIndTokenSize; Response: ST = 51h at ATA interface level (A10-3-2-1-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: ST = 51h at ATA Interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: ST = 51h at ATA Interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxPacketS; Response: Discarded by TPer	PASS PASS PASS PASS N/A N/A PASS PASS N/A
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: S1 = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: S1 = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxPacketSize; Response: Session abort Properties Effect - TPerProp in control session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: S1 = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: S1 = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: S1 = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: S1 = 51h at ATA interface level	PASS PASS PASS PASS N/A N/A PASS PASS
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: Pass (A10-3-2-1-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: S = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxCamPacketSize; Response: Session abort Properties Effect - TPerProp in cegular session: > TPer's MaxIndTokenSize; Response: ST = 51h at ATA interface level (A10-3-2-1-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: ST = 51h at ATA Interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: ST = 51h at ATA Interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxPacketS; Response: Discarded by TPer	PASS PASS PASS PASS N/A N/A PASS PASS N/A
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: S1 = 51h at ATA interface level (A10-3-2-1) Properties Effect - TPerProp in regular session: > TPer's MaxCamPacketSize; Response: S1 = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxPacketSize; Response: Session abort Properties Effect - TPerProp in control session: > TPer's MaxCamPacketSize; Response: S1h at ATA interface level (A10-3-2-1-1) Properties Effect - TPerProp in control session: > TPer's MaxCamPacketSize; Response: S1h at ATA interface level (A10-3-2-1-1) Properties Effect - TPerProp in control session: > TPer's MaxCamPacketS; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxCamPacketS; Response: S1 = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-5-1) Properties Effect - TPerProp in control session: > TPer's MaxSubPackets; Response: Discarded by TPer (A10-3-2-5-1) Properties Effect - TPerProp in control session: > TPer's MaxSubPackets; Response: Discarded by TPer	PASS PASS PASS N/A N/A PASS N/A N/A N/A
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-32) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: Pass (A10-3-2-1-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: S = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in cegular session: > TPer's MaxComPacketSize; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxSubPacketS; Response: Discarded by TPer	PASS PASS PASS N/A N/A PASS N/A N/A N/A PASS
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: Pass (A10-3-2-1-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: S1 = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in cegular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in control session: > TPer's MaxComPacketSize; Response: S1h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: S1 = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: S1 = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-5-1) Properties Effect - TPerProp in control session: > TPer's MaxSubPackets; Response: Discarded by TPer (A10-3-2-1) Properties Effect - TPerProp in control session: > TPer's MaxSubPackets; Response: Discarded by TPer (A10-3-2-1) Properties Effect - TPerProp in control session: MaxAuthentications shall not be 1 A11: Test Start/SyncSession] (A11-3-1-1) StartSession - SessionID: not all 0; SyncSession - Status Code: 01h (Not_Authorized) (A11-3-1-1) StartSession - HostSessionID: 4-byte uinteger(PASS PASS PASS N/A PASS N/A N/A PASS N/A PASS PASS PASS
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Sesion abort Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: Pass (A10-3-2-1-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: S = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in cegular session: > TPer's MaxComPacketSize; Response: S = 51h at ATA interface level (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketSize; Response: S = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxSubPackets; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxSubPackets; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxSubPackets; Response: Discarded by TPer (A10-3-2-4-5) Properties Effect - TPerProp in regular session: MaxAuthentications shall not be 1 A11: Test Start/SyncSession - SessionID: not all 0; SyncSession - Status Code: 01h (Not_Authorized) (A11-3-2-1-1) StartSession - HostSessionID: 4-byte uinteger(<0FFFFFFFFh); SyncSession - Pass and Tries = 0 in C_PIN table	PASS PASS PASS N/A N/A PASS N/A N/A N/A PASS PASS PASS
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: = TPer's MaxComPacketSize; Response: StatusCode = 11 properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxPacketSize; Response: Session abort Properties Effect - TPerProp in control session: = TPer's MaxComPacketSize; Response: (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in control session: = TPer's MaxComPackets; Response: ST = 51h at ATA interface level (A10-3-2-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-6-1) Properties Effect - TPerProp in control session: > TPer's MaxUndTokenSize; Response: Discarded by TPer (A10-3-2-6-1) Properties Effect - TPerProp in control session: > TPer's MaxUndTokenSize; Response: Discarded by TPer (A10-3-2-6-1) Properties Effect - TPerProp in control session: MaxAuthentications shall not be 1 A11: Test Start/SyncSession() (A11-1-1-1) StartSession - HostSessionID: not all 0; SyncSession - Status Code: 01h (Not_Authorized) (A11-3-2-1-1) StartSession - HostSessionID: 4-byte uinteger(=OFFFFFFFFh); SyncSession - Pass and Tries = 0 in C_PIN table (A11-3-2-1-3) StartSession - HostSessionID: 4-byte uinteger(=OFFFFFFFFh); SyncSession - Pass and Tries = 0 in C_PIN table	PASS PASS PASS N/A PASS N/A N/A PASS N/A PASS PASS PASS
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session : TPer's MaxComPacketSize; Response: Pass (A10-3-2-1-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in cegular session: > TPer's MaxCamPacketSize; Response: (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: ST = 51h at ATA Interface level (A10-3-2-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: ST = 51h at ATA Interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-4) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-6-1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-1) Properties Effect - TPerProp in control session: MaxAuthentications shall not be 1 A11: Test Start/SyncSession] (A11-1-1-1) StartSession - HostSessionD: not all 0; SyncSession - Status Code: Olh (Not_Authorized) (A11-3-1-1) StartSession - HostSessionD: A-byte uinteger(=OFFFFFFFFh); SyncSession - Pass and Tries = 0 in C_PIN table (A11-3-2-1-3) StartSession - HostSessionD: > 4-byte; SyncSession - Status Code: no data	PASS PASS PASS N/A N/A PASS N/A N/A PASS PASS PASS PASS PASS
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: = TPer's MaxComPacketSize; Response: Pass (A10-3-2-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxAndTokenSize; Response: Session abort Properties Effect - TPerProp in control session: = TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in control session: > TPer's MaxComPacketSize; Response: (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: ST = 51h at ATA interface level (A10-3-2-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-5-1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-5-1) Properties Effect - TPerProp in control session: MaxAuthentications shall not be 1 A11: Test Start/SymCsession() (A11-1-1-1) StartSession - HostSessionID: not all 0; SymCsession - Status Code: 10h (Not_Authorized) (A11-3-2-1) StartSession - HostSessionID: A-byte uinteger(=OFFFFFFFh); SynCsession - Pass and Tries = 0 in C_PIN table (A11-3-2-1) StartSession - HostSessionID: A-byte uinteger(=OFFFFFFFh); SynCsession - Pass and Tries = 0 in C_PIN table (A11-3-2-2) StartSession - HostSessionID: > 4-byte uinteger(=OFFFFFFFh); SynCs	PASS PASS PASS N/A PASS PASS N/A N/A PASS PASS PASS PASS
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session : TPer's MaxComPacketSize; Response: Pass (A10-3-2-1-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in cegular session: > TPer's MaxCamPacketSize; Response: (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: ST = 51h at ATA Interface level (A10-3-2-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketS; Response: ST = 51h at ATA Interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-4) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-6-1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-1) Properties Effect - TPerProp in control session: MaxAuthentications shall not be 1 A11: Test Start/SyncSession] (A11-1-1-1) StartSession - HostSessionD: not all 0; SyncSession - Status Code: Olh (Not_Authorized) (A11-3-1-1) StartSession - HostSessionD: A-byte uinteger(=OFFFFFFFFh); SyncSession - Pass and Tries = 0 in C_PIN table (A11-3-2-1-3) StartSession - HostSessionD: > 4-byte; SyncSession - Status Code: no data	PASS PASS PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Sesion abort Properties Effect - TPerProp in regular session: = TPer's MaxComPacketSize; Response: Pass (A10-3-2-1-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: T = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in cegular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in control session: > TPer's MaxComPacketSiz; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxJudTokenSize; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxJudTokenSize; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxJudPackets; Response: Discarded by TPer (A10-3-2-1-5) Properties Effect - TPerProp in control session: > TPer's MaxJudPackets; Response: Discarded by TPer (A11-3-2-1-1) StartSession - SessionID: not all 0; SyncSession - Status Code: 01h (Not_Authorized) (A11-3-2-1-1) StartSession - HostSessionID: 4-byte uinteger(<0FFFFFFFF); SyncSession - Pass and Tries = 0 in C_PIN table (A11-3-2-1-3) StartSession - HostSessionID: 4-byte uinteger(<0FFFFFFFFF); SyncSession - Pass and Tries = 0 in C_PIN table (A11-3-2-2) StartSession - SPUID: nockistent in the SP table; SyncSes	PASS PASS PASS N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: = TPer's MaxComPacketSize; Response: Pass (A10-3-2-1.1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: ST = 51h at ATA interface level (A10-3-2-3.1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in cegular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in control session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: ST = 51h at ATA interface level (A10-3-2-3.1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: ST = 51h at ATA Interface level (A10-3-2-3.1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-4.1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-5.1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer (A10-3-2-1.1) Properties Effect - TPerProp in control session: MaxAuthentications shall not be 1 A11: Test Start/SyncSession] - SessionID: not all 0; SyncSession - Status Code: Olh (Not_Authorized) (A11: 3-2-1.1) StartSession - HostSessionID: A-byte uinteger(=OFFFFFFFFh;); SyncSession - Pass and Tries = 0 in C_PIN table (A11: 3-2-1.3) StartSession - HostSessionID: > 4-byte; SyncSession - Status Code: no data returned (A11: 3-2-2.3) StartSession - SPUID: nonexistent in the SP table; SyncSession - Status Code: Och (invalid_Param) (A	PASS PASS PASS N/A PASS N/A N/A PASS N/A N/A PASS PASS PASS PASS PASS
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Sesion abort Properties Effect - TPerProp in regular session: = TPer's MaxComPacketSize; Response: Pass (A10-3-2-1-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: T = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in cegular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxJudTokenSize; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxJudPackets; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxJudPackets; Response: Discarded by TPer (A10-3-2-1-5) Properties Effect - TPerProp in control session: > TPer's MaxJudPackets; Response: Discarded by TPer (A11-3-2-1-1) StartSession - SessionID: not all 0; SyncSession - Status Code: 01h (Not_Authorized) (A11-3-2-1-1) StartSession - HostSessionID: 4-byte uinteger(<0FFFFFFFF); SyncSession - Pass and Tries = 0 in C_PIN table (A11-3-2-1-3) StartSession - HostSessionID: 4-byte uinteger(=0FFFFFFFFF); SyncSession - Status Code: 0Ch (Invalid_Param) (A11-3-2-2-3) StartSession - Write: 1; SyncSession - Status Code: 0Ch (Inv	PASS PASS PASS N/A PASS N/A PASS N/A N/A PASS PASS PASS PASS PASS PASS
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3:2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4:2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Sesion abort Properties Effect - TPerProp in regular session : TPer's MaxComPacketSize; Response: Pass (A10-3-2-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: S = 51h at ATA interface level (A10-3-2-4-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in cegular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxSubPackets; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in regular session: MaxAuthentications shall not be 1 A11: Test Start/SyncSession) (A11-1-1-1) StartSession - HostSessionID: A-byte uinteger(=0FFFFFFFFh); SyncSession - Pass and Tries = 0 in C_PIN table (A11-3-2-1) StartSession - HostSessionID: 4-byte uinteger(=0FFFFFFFFh); SyncSession - Pass and Tries = 0 in C_PIN table (A11-3-2-2) StartSession - SPUID: nonexistent in the SP table; SyncSession - Status Code: OCh (Invalid_Param) (A11-3-2-3) StartSession - SPUID: nonexistent in the SP table; SyncSession - Status Code: OCh (Invalid_Param) (A11-3-2-3) StartSession - Write: 1; SyncSession - Status Code	PASS PASS PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort Properties Effect - TPerProp in regular session: = TPer's MaxComPacketSize; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in cegular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in cegular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: Discarded by TPer (A10-3-2-1) Properties Effect - TPerProp in control session: > TPer's MaxSubPackets; Response: Discarded by TPer (A10-3-2-1) Properties Effect - TPerProp in control session: > TPer's MaxSubPackets; Response: Discarded by TPer (A10-3-2-1) Properties Effect - TPerProp in control session: > TPer's MaxSubPackets; Response: Discarded by TPer (A11-3-2-1) StartSession - HostSessionID: A-byte uinteger(<0FFFFFFFh); SyncSession - Pass and Tries = 0 in C_PIN table (A11-3-2-1) StartSession - HostSessionID: 4-byte uinteger(<0FFFFFFFFh); SyncSession - Status Code: OCh (Invaild_Param) (A11-3-2-3)(2) StartSession - Write: 1; SyncSession - Status Code: OCh (Invaild_Param) (A11-3-2-3)(2) StartSession - Write: 1; SyncSession - Status Code: OCh (Invaild_Param) (A11-3-2-3-3)(2) StartSession - Write: 1; SyncSession - Status Code: OCh (Invaild_Param) (A11-3-2-3-3)(2) StartSession - Write: 1	PASS PASS PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h (A10-3-1-3:2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h (A10-3-1-4:2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Sesion abort Properties Effect - TPerProp in regular session : TPer's MaxComPacketSize; Response: Pass (A10-3-2-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: S = 51h at ATA interface level (A10-3-2-4-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in cegular session: > TPer's MaxComPacketSize; Response: Session abort Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: ST = 51h at ATA interface level (A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxComPackets; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxSubPackets; Response: Discarded by TPer (A10-3-2-4-1) Properties Effect - TPerProp in regular session: MaxAuthentications shall not be 1 A11: Test Start/SyncSession) (A11-1-1-1) StartSession - HostSessionID: not all 0; SyncSession - Status Code: 01h (Not_Authorized) (A11-3-2-1) StartSession - HostSessionID: 4-byte uinteger(=0FFFFFFFFh); SyncSession - Pass and Tries = 0 in C_PN table (A11-3-2-2) StartSession - HostSessionID: 4-byte uinteger(=0FFFFFFFFh); SyncSession - Status Code: 0Ch (Invalid_Param) (A11-3-2-3) StartSession - SPUID: nonexistent in the SP table; SyncSession - Status Code: 0Ch (Invalid_Param) (A11-3-2-3) StartSession - Write: 1; SyncSession - Status Code: 0Ch (Inva	PASS PASS PASS PASS PASS PASS N/A PASS PASS PASS PASS PASS PASS PASS PA

(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) (A11-3-4-2-6(3)) StartSession - HostSignAuth: a class authority UID; SyncSession - Status Code: 0Ch	PASS
(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) Activating the Locking SP	PASS
Start Session with HostChallenge - AdminSP	PASS
Sync Session - AdminSP	PASS
Activate_LockingSP	PASS
Activate_LockingSP - Response	PASS
Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response	PASS PASS
Check the state of Locking SP	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 (A6-0-1-1-1) Get Request - with long atom for InvokingID; Response - Pass	PASS PASS
(A6-0-1-1-1) Get Request - with holig 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:	DACC
01h(Not_Authorized) (A6-1-1-3-1(2)) Get Request - with non-8-long token for InvokingID; Response - Status Code:	PASS
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 PASS
(A6-1-3-1-1) Get Request - no ACE in the ACL; Response - empty data returned with SUCCESS status (A6-1-3-1-1(2)) Get Request - nonexistent InvokingID/MethodID in ACL; Response - Status Code:	PASS
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 (A6-1-8-2-1) Get Request - with first Status Code != 0h(found in status code); Response - fail	PASS PASS
(A6-1-8-2-1) Get Request - with first Status Code != Oh(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 (A6-1-8-6-1) Get Request - with 2nd Status token = 91h(integer); Response - Session Abort	PASS PASS
(A6-1-8-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) Get Request - with 3rd 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 = A1h(byte); Response - Session Abort (A6-1-8-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-9-2-1) Get Request - with invalid token type of StatusCode End: Oeb); Response - Session (A6-1-4-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status	PASS PASS PASS
(A6-1-8-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-9-2-1) Get Request - with invalid token type of StatusCode End: 0e0h; Response - Session (A6-1-4-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param)	PASS PASS PASS PASS
(A6-1-8-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-9-2-1) Get Request - with invalid token type of StatusCode End: 0e0h; Response - Session (A6-1-4-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param) (A6-0-1-1-1) Set Request - with short atom for InvokingID; Response - Pass	PASS PASS PASS PASS PASS
(A6-1-8-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-9-2-1) Get Request - with invalid token type of Status Code End: Oeb); Response - Session (A6-1-4-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: OCh(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass	PASS PASS PASS PASS
(A6-1-8-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-9-2-1) Get Request - with invalid token type of StatusCode End: 0e0h; Response - Session (A6-1-4-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param) (A6-0-1-1-1) Set Request - with short atom for InvokingID; Response - Pass	PASS PASS PASS PASS PASS PASS
(A6-18-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-18-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-19-2-1) Get Request - with invalid token type of Status Code End: OeDh; Response - Session (A6-14-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: OCh(Invalid_Param) (A6-01-12) Set Request - with short atom for InvokingID; Response - Pass (A6-01-12) Set Request - with medium atom for InvokingID; Response - Pass (A6-01-12) Set Request - with medium atom for InvokingID; Response - Pass (A6-01-12) Set Request - with medium atom for InvokingID; Response - Pass (A6-01-12) Set Request - with medium atom for InvokingID; Response - Pass (A6-01-12) Set Request - with medium atom for InvokingID; Response - Pass (A6-01-12) Set Request - with long atom for InvokingID; Response - Pass (A6-01-12) Set Request - with long atom for MethodID; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-8-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-9-2-1) Get Request - with invalid token type of StatusCode End: Oeb); Response - Session (A6-1-4-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: OCh(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with long atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with invalid token for /Call'(F8h); Response - Session Abort	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-8-6-1) Get Request - with 3rd Status token = 41h(byte); Response - Session Abort (A6-1-8-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-9-2-1) Get Request - with invalid token type of StatusCode End. 0e0h; Response - Session (A6-1-4-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1.3) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1.3) Set Request - with medium atom for MethodID; Response - Pass (A6-1-1-1.3) Set Request - with nonexistent InvokingID; Response - Session Abort (A6-1-1-2) Set Request - with nonexistent InvokingID; Response - Status Code:	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-18-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-18-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-19-2-1) Get Request - with invalid token type of Status Code End: OeDi; Response - Session (A6-14-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: OCh(Invalid_Param) (A6-01-12) Set Request - with short atom for InvokingID; Response - Pass (A6-01-11) Set Request - with medium atom for InvokingID; Response - Pass (A6-01-12) Set Request - with long atom for InvokingID; Response - Pass (A6-01-12) Set Request - with long atom for InvokingID; Response - Pass (A6-01-12) Set Request - with long atom for InvokingID; Response - Pass (A6-01-12) Set Request - with long atom for MethodID; Response - Pass (A6-01-12) Set Request - with long atom for Call'(F8h); Response - Pass (A6-01-12) Set Request - with long atom for Call'(F8h); Response - Status Code: (A6-12-12) Set Request - with nonexistent InvokingID; Response - Status Code:	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-8-6-1) Get Request - with 3rd Status token = 41h(byte); Response - Session Abort (A6-1-8-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-9-2-1) Get Request - with invalid token type of StatusCode End. 0e0h; Response - Session (A6-1-4-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1.3) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1.3) Set Request - with medium atom for MethodID; Response - Pass (A6-1-1-1.3) Set Request - with nonexistent InvokingID; Response - Session Abort (A6-1-1-2) Set Request - with nonexistent InvokingID; Response - Status Code:	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-18-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-19-8-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-19-2-1) Get Request - with invalid token type of Status Code End. OeD; Response - Session (A6-19-42-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: OCh(Invalid_Param) (A6-01-12) Set Request - with short atom for InvokingID; Response - Pass (A6-01-11) Set Request - with short atom for InvokingID; Response - Pass (A6-01-11) Set Request - with ing atom for InvokingID; Response - Pass (A6-01-11) Set Request - with long atom for InvokingID; Response - Pass (A6-01-11) Set Request - with ing atom for InvokingID; Response - Pass (A6-01-11) Set Request - with ing atom for MethodID; Response - Pass (A6-01-11) Set Request - with ing atom for MethodID; Response - Pass (A6-01-12) Set Request - with ing atom for Call'(F8h); Response - Pass (A6-11-11) Set Request - with invalid token for 'Call'(F8h); Response - Status Code: (A6-11-3-1(2)) Set Request - with non-byte token for InvokingID; Response - Status Code: Oth(Not_Authorized) (A6-1-1-3-1(2)) Set Request - with non-8-long token for InvokingID; Response - Status Code: Oth(Not_Authorized)	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-1-8-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) Get Request - with invalid token type of StatusCode End: Oe0h; Response - Session (A6-1-4-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: Och(invalid_Param) (A6-0-1-1.1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1.1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1.1) Set Request - with ing atom for MethodID; Response - Pass (A6-0-1-1.1) Set Request - with ing atom for MethodID; Response - Pass (A6-0-1-1.1) Set Request - with ing atom for MethodID; Response - Pass (A6-0-1-1.1) Set Request - with ing atom for MethodID; Response - Pass (A6-1-1-1.1) Set Request - with invalid token for Call'(F8h); Response - Session Abort (A6-1-1-2.1) Set Request - with non-byte token for InvokingID; Response - Status Code: (D1(Not_Authorized)) (A6-1-2-3.1(2)) Set Request - with non-8-long token for InvokingID; Response - Status Code: (D1(Not_Authorized) (A6-1-2-2.1) Set Request - with non-8-long token for InvokingID; Response - Status Code: (D1(Not_Authorized) (A6-1-2-2.1) Set Request - with non-8-long token for InvokingID; Response - Status Code: (D1(Not_Authorized)) (A6-1-2-2.1) Set Request - with non-8-long token for InvokingID; Response - Status Code: 	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-8-6-1) Get Request - with 3rd Status token = 41h(byte); Response - Session Abort (A6-1-8-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-9-2.1) Get Request - with invalid token type of StatusCode End. 0e0h; Response - Session (A6-1-4-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param) (A6-0-1-1.1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1.1) Set Request - with molium atom for InvokingID; Response - Pass (A6-0-1-1.1) Set Request - with molium atom for InvokingID; Response - Pass (A6-0-1-1.1) Set Request - with molium atom for MethodID; Response - Pass (A6-0-1-1.1) Set Request - with molium atom for MethodID; Response - Pass (A6-0-1-1.1) Set Request - with nong atom for MethodID; Response - Pass (A6-0-1-1.2) Set Request - with none atom for MethodID; Response - Pass (A6-1-1-1.2) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-3.1) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-3.1) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-3.1(2)) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-3.1(2)) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-2.1(2)) Set Request - with nonexistent InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-1.2) Set Request - with non-8-long token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-2.1) Set Request - with nonexistent MethodID; Response - Status Code: (A6-1-2-3.1(2)) Set Request - with nonexistent MethodID; Response - Status Code:	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-1-8-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) Get Request - with invalid token type of StatusCode End: Oe0h; Response - Session (A6-1-4-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: Och(invalid_Param) (A6-0-1-1.1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1.1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1.1) Set Request - with ing atom for MethodID; Response - Pass (A6-0-1-1.1) Set Request - with ing atom for MethodID; Response - Pass (A6-0-1-1.1) Set Request - with ing atom for MethodID; Response - Pass (A6-0-1-1.1) Set Request - with ing atom for MethodID; Response - Pass (A6-1-1-1.1) Set Request - with invalid token for Call'(F8h); Response - Session Abort (A6-1-1-2.1) Set Request - with non-byte token for InvokingID; Response - Status Code: (D1(Not_Authorized)) (A6-1-2-3.1(2)) Set Request - with non-8-long token for InvokingID; Response - Status Code: (D1(Not_Authorized) (A6-1-2-2.1) Set Request - with non-8-long token for InvokingID; Response - Status Code: (D1(Not_Authorized) (A6-1-2-2.1) Set Request - with non-8-long token for InvokingID; Response - Status Code: (D1(Not_Authorized)) (A6-1-2-2.1) Set Request - with non-8-long token for InvokingID; Response - Status Code: 	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-18-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-18-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-19-2-1) Get Request - with invalid token type of StatusCode End. OeD; Response - Session (A6-14-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: OCh(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with invalid token for InvokingID; Response - Pass (A6-0-1-1) Set Request - with long atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with long atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with invalid token for 'Call'(F8h); Response - Pass (A6-0-1-1) Set Request - with none/stent InvokingID; Response - Pass (A6-1-1-1) Set Request - with none/stent InvokingID; Response - Session Abort (A6-1-1-3) Set Request - with none/stent InvokingID; Response - Session Abort (A6-1-1-3) Set Request - with none/stent InvokingID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-1(2)) Set Request - with none/stent MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-3-1) Set Request - with none/stent MethodID; Response - Status Code: (A6-1-2-3-1) Set Request - with none/stent MethodID; Response - Status Code: (A6-1-2-3-1) Set Request - with none/stent MethodID; Response - Status Code: (A6-1-2-3-1) Set Request - with none/stent MethodID; Response - Status Code: (A6-1-2-3-1) Set Request - with none/stent MethodID; Response - Status Code: (A6-1-2-3-1) Set Request - with none/stent MethodID; Response - Status Code: (A6-1-2-3-1) Set Request - with none/stent MethodID; Response - Status Code: (A6-1-2-3-1) Set Reque	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-1-8-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) Get Request - with invalid token type of StatusCode End: Oe0h; Response - Session (A6-1-4-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: OCh(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with long atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with invalid token for 'Call'(F8h); Response - Pass (A6-1-1-1-1) Set Request - with nonexistent InvokingID; Response - Pass (A6-1-1-1) Set Request - with none-byte token for InvokingID; Response - Status Code: (A6-1-1-3-1(2)) Set Request - with non-8-long token for InvokingID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-31(2)) Set Request - with non-8-long token for InvokingID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-31(2)) Set Request - with non-byte token for InvokingID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-31(2)) Set Request - with non-8-long token for InvokingID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-31(2)) Set Request - with non-8-long token for MethodID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-31(2)) Set Request - with non-8-long token for MethodID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-31(2)) Set Request - with non-8-long token for MethodID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-31(2)) Set Request - with non-8-long token for MethodID; Response - Status Code:	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-1-8-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) Get Request - with invalid token type of StatusCode End: Ocolh; Response - Session (A6-1-4-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: OCh(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with long atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with invalid token for 'Call'(F8h); Response - Pass (A6-1-1-1) Set Request - with nonexistent InvokingID; Response - Pass (A6-1-1-1) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-3-1(2)) Set Request - with non-Byte token for InvokingID; Response - Status Code: (A6-1-1-3-1(2)) Set Request - with non-8-long token for InvokingID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-1(2)) Set Request - with non-8-long token for InvokingID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-1(2)) Set Request - with non-byte token for MethodID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-1(2)) Set Request - with non-byte token for MethodID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-31(2)) Set Request - with non-8-long token for MethodID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-31(2)) Set Request - with non-8-long token for MethodID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-31(2)) Set Request - with non-8-long token for MethodID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-1(2	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-18-6-1) Get Request - with 3rd Status token - 41h(byte); Response - Session Abort (A6-18-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-19-2-1) Get Request - with invalid token type of StatusCode End. OeD; Response - Session (A6-14-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: Och(Invalid_Param) (A6-0-1-1.1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1.1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1.1) Set Request - with modium atom for InvokingID; Response - Pass (A6-0-1-1.1) Set Request - with modium atom for MethodID; Response - Pass (A6-0-1-1.1) Set Request - with modium atom for MethodID; Response - Pass (A6-0-1-1.3) Set Request - with nong atom for MethodID; Response - Pass (A6-0-1-1.3) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-3.1) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-3.1) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-3.1(2)) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-3.1(2)) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-2.1(2)) Set Request - with non-8-long token for InvokingID; Response - Status Code: (A6-1-2.3.1(2)) Set Request - with non-byte token for InvokingID; Response - Status Code: (A6-1-2.3.1(2)) Set Request - with nonexistent MethodID; Response - Status Code: (A6-1-2.3.1(2)) Set Request - with non-byte token for MethodID; Response - Status Code: (A6-1-3.3.1(2)) Set Request - with non-8-long token for MethodID; Response - Status Code: (A6-1-3.3.1(2)) Set Request - with non-8-long token for MethodID; Response - Status Code: (A6-1-3.3.1(2)) Set Request - with non-8-long token for MethodID; Respon	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-18-6-1) Get Request - with and Status token = A1h(byte); Response - Session Abort (A6-18-6-1) Get Request - with and Status token = 91h(integer); Response - Session Abort (A6-19-2-1) Get Request - with and Status token = 91h(integer); Response - Session Abort (A6-19-2-1) Get Request - with unexpected token encoded inside the Params; Response - Status Code: OCh(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with invalid token for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with invalid token for Call'(F8h); Response - Pass (A6-0-1-1) Set Request - with invalid token for Call'(F8h); Response - Pass (A6-0-1-1) Set Request - with non-Byte token for InvokingID; Response - Session Abort (A6-1-1-3(1)) Set Request - with non-Byte token for InvokingID; Response - Status Code: (A6-1-1-3(1)) Set Request - with non-Byte token for InvokingID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-2.1) Set Request - with non-Byte token for InvokingID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-2.1) Set Request - with non-Byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-3.1(2)) Set Request - with non-Byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-3.1(2)) Set Request - with non-Byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-3.1(2)) Set Request - with non-Byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-3.1(2)) Set Request - with non-Byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-3.1(2)) Set Request - with no	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-1-8-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) Get Request - with invalid token type of StatusCode End: Ocol; Response - Session (A6-1-4-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: OCh(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with long atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with long atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with invalid token for Call'(F8h); Response - Pass (A6-0-1-1) Set Request - with nonexistent InvokingID; Response - Pass (A6-1-1-1) Set Request - with nonexistent InvokingID; Response - Pass (A6-1-1-1) Set Request - with nonexistent InvokingID; Response - Session Abort (A6-1-1-3.1(2)) Set Request - with none-byte token for InvokingID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-3.1(2)) Set Request - with none-8-long token for InvokingID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-3.1(2)) Set Request - with none-8-long token for InvokingID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-3.1(2)) Set Request - with none-8-long token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-3.1(2)) Set Request - with none-8-long token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-3.1(2)) Set Request - with none-8-long token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-3.1(2)) Set Request - with non-8-long token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-3.1(2)) Set Request - with n	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-18-6-1) Get Request - with and Status token = A1h(byte); Response - Session Abort (A6-18-6-1) Get Request - with and Status token = 91h(integer); Response - Session Abort (A6-19-2-1) Get Request - with and Status token = 91h(integer); Response - Session Abort (A6-19-2-1) Get Request - with unexpected token encoded inside the Params; Response - Status Code: OCh(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with invalid token for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with invalid token for Call'(F8h); Response - Pass (A6-0-1-1) Set Request - with invalid token for Call'(F8h); Response - Pass (A6-0-1-1) Set Request - with non-Byte token for InvokingID; Response - Session Abort (A6-1-1-3(1)) Set Request - with non-Byte token for InvokingID; Response - Status Code: (A6-1-1-3(1)) Set Request - with non-Byte token for InvokingID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-2.1) Set Request - with non-Byte token for InvokingID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-2.1) Set Request - with non-Byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-3.1(2)) Set Request - with non-Byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-3.1(2)) Set Request - with non-Byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-3.1(2)) Set Request - with non-Byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-3.1(2)) Set Request - with non-Byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-3.1(2)) Set Request - with no	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-1-8-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) Get Request - with invalid token type of StatusCode End: Ocol; Response - Session (A6-1-4-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: OCh(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with invalid token for 'Call'(F8h); Response - Pass (A6-1-1-1)(1) Set Request - with nonexistent InvokingID; Response - Pass (A6-1-1-21) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-31(2)) Set Request - with none-byte token for InvokingID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-31) Set Request - with none-8-long token for InvokingID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-3-1(2)) Set Request - with non-byte token for InvokingID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-3-1(2)) Set Request - with non-byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-3-1(2)) Set Request - with non-byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-3-1(2)) Set Request - with non-8-long token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-3-1(2)) Set Request - with non-8-long token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-3-1(2)) Set Request - with non-8-long token for MethodID; Response - Status Code: O1h(Not_Authorized)<td>PASS PASS PASS PASS PASS PASS PASS PASS</td>	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-18-6-1) Get Request - with 3rd Status token - A1h(byte); Response - Session Abort (A6-18-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-4-2-1) Get Request - with invalid token type of StatusCode End. OeD; Response - Session (A6-1-4-2-1) Get Request - with unexpected token encoded inside the Params; Response - Status Code: Och(Invalid_Param) (A6-0-1-1.1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1.1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1.1) Set Request - with modium atom for InvokingID; Response - Pass (A6-0-1-1.1) Set Request - with modium atom for MethodID; Response - Pass (A6-0-1-1.1) Set Request - with mol atom for InvokingID; Response - Pass (A6-0-1-1.1) Set Request - with invalid token for 'Call'(FBh); Response - Pass (A6-0-1-1.2) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-3.1) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-3.12)) Set Request - with non-B-long token for InvokingID; Response - Status Code: (A6-1-2-1) Set Request - with non-B-long token for InvokingID; Response - Status Code: (A6-1-2-1) Set Request - with non-B-long token for MethodID; Response - Status Code: (A6-1-2-1) Set Request - with non-B-long token for MethodID; Response - Status Code: (A6-1-3-1(2)) Set Request - with non-B-long token for MethodID; Response - Status Code: (A6-1-3-1(2)) Set Request - with non-B-long token for MethodID; Response - Status Code: (A6-1-3-12)) Set Request - with non-B-long token for MethodID; Response - Status Code: (A1(Not, Authorized)) (A6-1-3-1) Set Request - with non-B-long token for MethodID; Response - Status Code: (A1(Not, Authorized)) (A6-1-3-1) Set Request - with non-B-long token for MethodID; Response - Session Abort<td>PASS PASS PASS PASS PASS PASS PASS PASS</td>	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-18-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-18-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-19-2-1) Get Request - with invalid token type of StatusCode End. Oe0h; Response - Session (A6-14-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: OCh(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with long atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with invalid token for 'Call'(F8h); Response - Pass (A6-0-1-1) Set Request - with non-Byte token for InvokingID; Response - Session Abort (A6-1-1-3(1)) Set Request - with non-byte token for InvokingID; Response - Status Code: (A6-1-1-3(1)) Set Request - with non-byte token for InvokingID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-2.1) Set Request - with non-B-long token for InvokingID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-3.1(2)) Set Request - with non-B-long token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-3.1(2)) Set Request - with non-B-long token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-1.1) Set Request - with non-B-long token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-1.1) Set Request - with non-B-long token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-1.1) Set Request - with non-B-long token for MethodID; Response - Session Abort (A6-1-3-1.1) Set Request - with invalid token type of StartL	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-18-6-1) Get Request - with 3rd Status token = 41h(byte); Response - Session Abort (A6-18-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-14-2-1) Get Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with none atom for MethodID; Response - Pass (A6-0-1-1-2) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-2-1) Set Request - with none-byte token for InvokingID; Response - Status Code: (A6-1-1-3-31(2)) Set Request - with none-byte token for InvokingID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-1(2)) Set Request - with none-byte token for MethodID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-1(2)) Set Request - with non-byte token for MethodID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-1(2)) Set Request - with non-byte token for MethodID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-1(2)) Set Request - with non-8-long token for MethodID; Response - Status Code: (D1h(Not_Authorized) (A6-1-3-1-1) Set Request - with non-8-long token for MethodID; Response - Status Code: (D1h(Not_Authorized) (A6-1-3-1-1) Set Request - with invalid token type of EndUst: 0e0h; Response - Session Abort (A6-	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-18-6-1) Get Request - with 3rd Status token - 41h(byte); Response - Session Abort (A6-18-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-14-2-1) Get Request - with invalid token type of StatusCode End. OeD; Response - Session (A6-14-2-21) Get Request - with unexpected token encoded inside the Params; Response - Status Code: Och(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with molium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with molium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with molium atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with invalid token for 'Call'(FBh); Response - Pass (A6-0-1-1) Set Request - with nonexistent InvokingID; Response - Pass (A6-0-1-2) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-2-3) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-2-1) Set Request - with nonexistent MethodID; Response - Status Code: (A6-1-2-1) Set Request - with nonexistent MethodID; Response - Status Code: (A6-1-2-1) Set Request - with nonexistent MethodID; Response - Status Code: (A6-1-2-3) Set Request - with nonexistent for MethodID; Response - Status Code: (A6-1-3-1(2)) Set Request - with nonexistent MethodID; Response - Status Code: (A1(Not, Authorized)) (A6-1-3-1) Set Request - with nonexistent for MethodID; Response - Status Code: (A1(Not, Authorized)) (A6-1-3-1) Set Request - with noneXiong token for MethodID; Response - Session Abort (A6-1-2-1) Set Request - with invalid token type of StatusCode: (A6-1-2-1) Se	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-18-6-1) Get Request - with 3rd Status token = 41h(byte); Response - Session Abort (A6-18-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-14-2-1) Get Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with none atom for MethodID; Response - Pass (A6-0-1-1-2) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-2-1) Set Request - with none-byte token for InvokingID; Response - Status Code: (A6-1-1-3-31(2)) Set Request - with none-byte token for InvokingID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-1(2)) Set Request - with none-byte token for MethodID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-1(2)) Set Request - with non-byte token for MethodID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-1(2)) Set Request - with non-byte token for MethodID; Response - Status Code: (D1h(Not_Authorized) (A6-1-2-3-1(2)) Set Request - with non-8-long token for MethodID; Response - Status Code: (D1h(Not_Authorized) (A6-1-3-1-1) Set Request - with non-8-long token for MethodID; Response - Status Code: (D1h(Not_Authorized) (A6-1-3-1-1) Set Request - with invalid token type of EndUst: 0e0h; Response - Session Abort (A6-	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-18-6-1) Get Request - with and Status token = A1h(byte); Response - Session Abort (A6-19-6-1) Get Request - with and Status token = 91h(integer); Response - Session Abort (A6-19-2-1) Get Request - with invalid token type of StatusCode End. Oe0h; Response - Session (A6-14-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: OCh(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with and tom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with long atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with invalid token for 'Call'(F8h); Response - Session Abort (A6-1-1-3) Set Request - with invalid token for 'Call'(F8h); Response - Session Abort (A6-1-1-3) Set Request - with non-Byte token for InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with non-Byte token for InvokingID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-2.1) Set Request - with non-Byte token for InvokingID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-2.1) Set Request - with non-Byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-3.1(2)) Set Request - with non-Byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-1(2)) Set Request - with non-Byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-1(2)) Set Request - with non-Byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-1(2)) Set Request - with non-Byte token for MethodID; Response - Session Abort (A6-1-3-1) Set Request - with invalid token type of StartList: Oe0h; Resp	PASS PASS
 (A6-18-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-19-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-19-2-1) Get Request - with invalid token type of StatusCode End. Oe0h; Response - Session (A6-14-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: OCh(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with modium atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with long atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with invalid token for 'Call'(F8h); Response - Pass (A6-0-1-1) Set Request - with none for MethodID; Response - Pass (A6-1-1-1) Set Request - with none for MethodID; Response - Session Abort (A6-1-1-3) Set Request - with none-Byte token for InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with non-Byte token for InvokingID; Response - Status Code: Olh(Not_Authorized) (A6-1-2-2) Set Request - with non-Byte token for InvokingID; Response - Status Code: Olh(Not_Authorized) (A6-1-2-2.1) Set Request - with non-Byte token for MethodID; Response - Status Code: Olh(Not_Authorized) (A6-1-2-3.1(2)) Set Request - with non-Byte token for MethodID; Response - Status Code: Olh(Not_Authorized) (A6-1-3-10) Set Request - with non-Byte token for MethodID; Response - Status Code: Olh(Not_Authorized) (A6-1-3-12) Set Request - with non-Byte token for MethodID; Response - Status Code: Olh(Not_Authorized) (A6-1-3-12) Set Request - with non-Byte token for MethodID; Response - Status Code: Olh(Not_Authorized) (A6-1-3-13) Set Request - with non-Byte token for MethodID; Response - Session	PASS PASS
 (A6-18-6-1) Get Request - with 3rd Status token = 41h(byte); Response - Session Abort (A6-18-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-14-2-1) Get Request - with invalid token type of StatusCode End: 0e0h; Response - Session (A6-14-2-21)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with none atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-2-3) Set Request - with nonexistent MethodID; Response - Status Code: (A6-1-2-3) Set Request - with nonexistent MethodID; Response - Status Code: (A6-1-2-3-12)) Set Request - with nonexistent MethodID; Response - Status Code: (A6-1-2-3-12)) Set Request - with nonexistent MethodID; Response - Status Code: (A6-1-2-3-12)) Set Request - with nonexistent MethodID; Response - Status Code: (A6-1-3-1-1) Set Request - no ACE in the ACL; Response - Status Code: (A6-1-3-1-1) Set Request - no ACE in the ACL; Response - Status Code: Naturorized) (A6-1-3-1-1) Set Request - with invalid token type of EndUata: 0e0h; Response - Session Abort (A6-1-3-2-1) Set Request - with invalid token type of EndUata: 0e0h; Response - Session Abort (A6-1-3-2-1) Set Request - with first S	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-18-6-1) Get Request - with 3rd Status token - 41h(byte); Response - Session Abort (A6-18-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-14-2-1) Get Request - with invalid token type of StatusCode End. OeD; Response - Session (A6-14-2-21) Get Request - with unexpected token encoded inside the Params; Response - Status Code: Och(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with modium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with invalid token for Call'(FB); Response - Pass (A6-0-1-1) Set Request - with none atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with invalid token for Call'(FB); Response - Status Code: (A6-1-1-3) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with none-byte token for InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with none-byte token for InvokingID; Response - Status Code: (A6-1-3-31(2)) Set Request - with none-byte token for InvokingID; Response - Status Code: (A1(Not_Authorized) (A6-1-2-31) Set Request - with none-byte token for MethodID; Response - Status Code: (A1(Not_Authorized) (A6-1-2-31(2)) Set Request - with none-8-long token for MethodID; Response - Status Code: (A1(Not_Authorized) (A6-1-2-31(2)) Set Request - with none-8-long token for MethodID; Response - Status Code: (A1(Not_Authorized) (A6-1-2-31(2)) Set Request - with none-8-long token for MethodID; Response - Session Abort (A6-1-2-31(2)) Set Request - with invalid token type of EndUst: 0e0h; Response - Session Abort (A6-1-2-31(2)) Set Request - with none-8-long token for MethodID; Response - Session Abort (A6-1-2-31) Se	PASS PASS
 (A6-18-6-1) Get Request - with and Status token = A1h(byte); Response - Session Abort (A6-19-6-1) Get Request - with and Status token = 91h(integer); Response - Session Abort (A6-19-2-1) Get Request - with invalid token type of StatusCode End. Oe0h; Response - Session (A6-14-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: OCh(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with long atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with invalid token for 'Call'(F8h); Response - Session Abort (A6-1-1-3) Set Request - with invalid token for 'Call'(F8h); Response - Session Abort (A6-1-1-3) Set Request - with non-Byte token for InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with non-Byte token for InvokingID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-2.1) Set Request - with non-Byte token for InvokingID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-2.1) Set Request - with non-Byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-2-3.1(2)) Set Request - with non-Byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-1(2)) Set Request - with non-Byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-1(2)) Set Request - with non-Byte token for MethodID; Response - Status Code: O1h(Not_Authorized) (A6-1-3-1(2)) Set Request - with non-Byte token for MethodID; Response - Session Abort (A6-1-3-1) Set Request - with invalid token type of Startus: Code); Response - Session Abort (A6-1-3-1) Set Request - with invalid token type o	PASS PASS
 (A6-18-6-1) Get Request - with 3rd Status token - 41h(byte); Response - Session Abort (A6-18-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-14-2-1) Get Request - with invalid token type of StatusCode End. OeD; Response - Session (A6-14-2-21) Get Request - with unexpected token encoded inside the Params; Response - Status Code: Och(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with modium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with invalid token for Call'(FB); Response - Pass (A6-0-1-1) Set Request - with none atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with invalid token for Call'(FB); Response - Status Code: (A6-1-1-3) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with none-byte token for InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with none-byte token for InvokingID; Response - Status Code: (A6-1-3-31(2)) Set Request - with none-byte token for InvokingID; Response - Status Code: (A1(Not_Authorized) (A6-1-2-31) Set Request - with none-byte token for MethodID; Response - Status Code: (A1(Not_Authorized) (A6-1-2-31(2)) Set Request - with none-8-long token for MethodID; Response - Status Code: (A1(Not_Authorized) (A6-1-2-31(2)) Set Request - with none-8-long token for MethodID; Response - Status Code: (A1(Not_Authorized) (A6-1-2-31(2)) Set Request - with none-8-long token for MethodID; Response - Session Abort (A6-1-2-31(2)) Set Request - with invalid token type of EndUst: 0e0h; Response - Session Abort (A6-1-2-31(2)) Set Request - with none-8-long token for MethodID; Response - Session Abort (A6-1-2-31) Se	PASS PASS
 (A6-18-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-19-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-19-2-1) Get Request - with invalid token type of StatusCode End. Oe0h; Response - Session (A6-14-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: OCh(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with long atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with invalid token for 'Call'(F8h); Response - Session Abort (A6-1-1-3) Set Request - with none'stent InvokingID; Response - Session Abort (A6-1-1-3) Set Request - with none'stent InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with non-Byte token for InvokingID; Response - Status Code: Oth(Not_Authorized) (A6-1-2-3) Set Request - with non-Byte token for InvokingID; Response - Status Code: Oth(Not_Authorized) (A6-1-2-3) Set Request - with non-Byte token for MethodID; Response - Status Code: Oth(Not_Authorized) (A6-1-2-3) Set Request - with non-Byte token for MethodID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-1(2)) Set Request - with non-Byte token for MethodID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-12) Set Request - with non-Byte token for MethodID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-12) Set Request - with non-Byte token for MethodID; Response - Session Abort (A6-1-3-13) Set Request - with invalid token type of Status Code: Session Abort (A6-1-3-13) Set Request - with invalid token type of Status Code: Session Abort (A6	PASS PASS
 (A6-18-6-1) Get Request - with 3rd Status token = 41h(byte); Response - Session Abort (A6-18-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-14-2-21) Get Request - with unexpected token encoded inside the Params; Response - Status Code: Och(Invalid_Param) (A6-0-1-1) Set Request - with short atom for Invoking[D; Response - Pass (A6-0-1-1) Set Request - with medium atom for Invoking[D; Response - Pass (A6-0-1-1) Set Request - with medium atom for Invoking[D; Response - Pass (A6-0-1-1) Set Request - with medium atom for Invoking[D; Response - Pass (A6-0-1-1) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with invalid token for Call'(FB); Response - Status (A6-1-1-21) Set Request - with nonexistent Invoking[D; Response - Status Code: (A6-1-1-3.1) Set Request - with nonexistent Invoking[D; Response - Status Code: (A6-1-3-1) Set Request - with nonexistent Invoking[D; Response - Status Code: (A6-1-3-12)) Set Request - with nonexistent MethodID; Response - Status Code: (A6-1-2-3.12)) Set Request - with non-8-long token for Invoking[D; Response - Status Code: (A6-1-2-3.12)) Set Request - with nonexistent MethodID; Response - Status Code: (A6-1-2-3.12)) Set Request - with nonexistent MethodID; Response - Status Code: (A6-1-3-3.12)) Set Request - with nonexistent MethodID; Response - Status Code: (A6-1-3-3.12)) Set Request - with noneXistent MethodID; Response - Status Code: (A6-1-3-3.12)) Set Request - with noneXistent MethodID; Response - Status Code: (A6-1-3-3.12)) Set Request - with noneXistent MethodID; Response - Status Code: (A6-1-3-3.12)) Set Request - with non-8-long token for MethodID; Response - Session Abort (A6-1-2-3.15) Set Request - with noneXistent MethodID; Response - Session Abort (A6-1-2-3.15) Set Request - with nox	PASS PASS
 (A6-18-6-1) Get Request - with 3rd Status token - 41h(byte); Response - Session Abort (A6-18-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-14-2-1) Get Request - with unexpected token encoded inside the Params; Response - Status Code: Och(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with modium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with modium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with invalid token for Call'(FB); Response - Pass (A6-0-1-1) Set Request - with nonexistent InvokingID; Response - Pass (A6-0-1-1) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with none-byte token for InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with none-byte token for InvokingID; Response - Status Code: (A6-1-3-3) Set Request - with none-byte token for InvokingID; Response - Status Code: (A6-1-3-3) Set Request - with none-B-long token for InvokingID; Response - Status Code: (A1(Not_Authorized) (A6-1-2-3) Set Request - with none-B-long token for MethodID; Response - Status Code: (A1(Not_Authorized) (A6-1-2-3) Set Request - with none-B-long token for MethodID; Response - Status Code: (A1(Not_Authorized) (A6-1-2-3) Set Request - with invalid token type of Status Code: (A1(Not_Authorized) (A6-1-2-3) Set Request - with invalid token type of Status Code: (A1(Not_Authorized) (A6-1-2-3) Set Request - with invalid token type of EndUst: 0e0h; Response - Session Abort (A6-1-2-1) Set Request - with invalid token type of EndUst: 0e0h; Response - Session Abort (A6-1-2-1) Set Request - with first Status Code = 0h(found in	PASS PASS
 (A6-18-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-19-62-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-19-2-1) Get Request - with unexpected token encoded inside the Params; Response - Status Code: OCh(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with long atom for MethodID; Response - Pass (A6-0-1-1) Set Request - with invalid token for 'Call'(F8h); Response - Session Abort (A6-1-1-3) Set Request - with none'stent InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with none'stent InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with none'stent InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with none'stent InvokingID; Response - Status Code: (A6-1-2-2) Set Request - with none'stent MethodID; Response - Status Code: (A6-1-2-2) Set Request - with none'stent MethodID; Response - Status Code: (A6-1-2-2) Set Request - with none'stent MethodID; Response - Status Code: (A6-1-3-1(2)) Set Request - with none'stent MethodID; Response - Status Code: (A6-1-3-1(2)) Set Request - with none'stent for MethodID; Response - Status Code: (A6-1-3-12) Set Request - with none'stent for MethodID; Response - Status Code: (A6-1-3-12) Set Request - with none'stent for MethodID; Response - Session Abort (A6-1-3-12) Set Request - with invalid token type of Status Code: Session Abort (A6-1-3-13-13) Set Request - with invalid token type of Status Code: Session Abort (A6-1-3-14) Set Request - with first Status Code = 10h(Rot1); Response	PASS PASS
 (A6-18-6-1) Get Request - with 3rd Status token - 41h(byte); Response - Session Abort (A6-18-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-14-2-1) Get Request - with unexpected token encoded inside the Params; Response - Status Code: Och(Invalid_Param) (A6-0-1-1) Set Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with modium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with modium atom for InvokingID; Response - Pass (A6-0-1-1) Set Request - with invalid token for Call'(FB); Response - Pass (A6-0-1-1) Set Request - with nonexistent InvokingID; Response - Pass (A6-0-1-1) Set Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with none-byte token for InvokingID; Response - Status Code: (A6-1-1-3) Set Request - with none-byte token for InvokingID; Response - Status Code: (A6-1-3-3) Set Request - with none-byte token for InvokingID; Response - Status Code: (A6-1-3-3) Set Request - with none-B-long token for InvokingID; Response - Status Code: (A1(Not_Authorized) (A6-1-2-3) Set Request - with none-B-long token for MethodID; Response - Status Code: (A1(Not_Authorized) (A6-1-2-3) Set Request - with none-B-long token for MethodID; Response - Status Code: (A1(Not_Authorized) (A6-1-2-3) Set Request - with invalid token type of Status Code: (A1(Not_Authorized) (A6-1-2-3) Set Request - with invalid token type of Status Code: (A1(Not_Authorized) (A6-1-2-3) Set Request - with invalid token type of EndUst: 0e0h; Response - Session Abort (A6-1-2-1) Set Request - with invalid token type of EndUst: 0e0h; Response - Session Abort (A6-1-2-1) Set Request - with first Status Code = 0h(found in	PASS PASS

(A6-0-1-1-1) Next Request - with long atom for MethodID; Response - Pass	PASS
(A6-1-1-1(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:	DACC
01h(Not_Authorized) (A6-1-1-3-1(2)) Next Request - with non-8-long token for InvokingID; Response - Status Code:	PASS
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 (A6-1-8-2-1) Next Request - with first Status Code != 0h(not in the status code); Response - fail	PASS PASS
(A6-1-8-3-2) Next Request - with second Status Code != Onnot in the status code, Response - Nam (A6-1-8-3-2) Next Request - with second Status Code != On 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 (A6-1-8-6-1) Next Request - with 3rd Status token = 91h(integer); Response - Session Abort	PASS PASS
(A6-1-8-6-1) Next Request - with 3rd Status token = 91n(integer); Response - Session Abort (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 (A6-0-1-1-1) GetACL Request - with medium atom for InvokingID; Response - Pass	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 Holig atom for Mothing), Response - Pass	PASS
(A6-0-1-1-1) GetACL Request - with long atom for MethodID; Response - Pass	PASS
(A6-1-1-1(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:	DACC
01h(Not_Authorized) (A6-1-1-3-1(2)) GetACL Request - with non-8-long token for InvokingID; Response - Status Code:	PASS
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:	0.00-
01h(Not_Authorized)	PASS 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 - Section Abort	
(A6-1-5-2-1) GetACL Request - with invalid token type of EndList: 0e0h; Response - Session Abort (A6-1-6-2-1) GetACL Request - with invalid token type of EndData; 0e0h; Response - Session Abort	PASS
(A6-1-5-2-1) GetACL Request - with invalid token type of EndList: 0e0h; Response - Session Abort (A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2.1) GetACL Request - with invalid token type of StatusCode	PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort	
$(A6-1-6-2-1) \ \mbox{GetACL} \ \mbox{Request} - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) \ \mbox{GetACL} \ \mbox{Request} - with invalid token type of StatusCode Statu: 0e0h; Response - Session (A6-1-8-2) \ \mbox{GetACL} \ \mbox{GetACL} \ \mbox{Request} - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) \ \mbox{GetACL} \ \mbox{Request} - with first Status token = 81h(short); Response - Session (A6-1-8-2-1) \ \mbox{GetACL} \ \mbox{GetACL} \ \mbox{GetACL} \ \mbox{Request} - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) \ \mbox{GetACL} \ \$	PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Start: 0e0h; Response - Session (A6-1-8-1-2) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(found in status code); Response - fail (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(not in the status code); Response - fail	PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Start: 0e0h; Response - Session (A6-1-8-1-2) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status Code = 1e0h(found in status code); Response - fail (A6-1-8-2-1) GetACL Request - with first Status Code = 0h(not in the status code); Response - fail (A6-1-8-3-2) GetACL Request - with status Code = 0h(not in the status code); Response - fail (A6-1-8-3-2) GetACL Request - with second Status Code = 0h; Response - Normal	PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Session (A6-1-8-1-2) GetACL Request - with first Status token = 81h(short); Response - Pasa (A6-1-8-2-1) GetACL Request - with first Status Code = 10h(found in status code); Response - fail (A6-1-8-2-1) GetACL Request - with first Status Code = 10h(not in the status code); Response - fail (A6-1-8-3-2) GetACL Request - with first Status Code = 10h(not in the status code); Response - fail (A6-1-8-3-2) GetACL Request - with first Status Code = 10h; Response - Normal (A6-1-8-3-2) GetACL Request - with third Status Code = 10h; Response - Normal	PASS PASS PASS PASS PASS PASS
$ \begin{array}{l} (A6-1-6-2-1) \; GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort \\ (A6-1-7-2-1) \; GetACL Request - with invalid token type of StatusCode Start: 0e0h; Response - Session \\ (A6-1-8-1-2) \; GetACL Request - with first Status token = 81h(short); Response - Pass \\ (A6-1-8-2-1) \; GetACL Request - with first Status Code = 0h(fround in status code); Response - fail \\ (A6-1-8-2-1) \; GetACL Request - with first Status Code = 0h(not in the status code); Response - fail \\ (A6-1-8-2-1) \; GetACL Request - with first Status Code = 0h(not in the status code); Response - fail \\ (A6-1-8-2-2) \; GetACL Request - with first Status Code = 0h(not in the status code); Response - fail \\ (A6-1-8-2-2) \; GetACL Request - with first Status Code = 0h; Response - Normal \\ (A6-1-8-2-2) \; GetACL Request - with first Status Code = 0h; Response - Normal \\ (A6-1-8-6-1) \; GetACL Request - with 1st Status Code = 0h(response - Normal \\ (A6-1-8-6-1) \; GetACL Request - with 1st Status Code = 0h(response - Normal \\ (A6-1-8-6-1) \; GetACL Request - with 1st Status Code = 0h(response - Session Abort \\ \end{array}$	PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Start: 0e0h; Response - Session (A6-1-8-1-2) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status Code = 0h(found in status code); Response - fail (A6-1-8-2-1) GetACL Request - with first Status Code = 0h(not in the status code); Response - fail (A6-1-8-3-2) GetACL Request - with second Status Code = 0h(not in the status code); Response - fail (A6-1-8-3-2) GetACL Request - with second Status Code = 0h; Response - Normal (A6-1-8-1) GetACL Request - with third Status Code = 0h; Response - Normal (A6-1-8-1) GetACL Request - with 1st Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = 91h(integer); Response - Session Abort	PASS PASS PASS PASS PASS PASS PASS PASS
$ \begin{array}{l} (A6-1-6-2-1) \; GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort \\ (A6-1-7-2-1) \; GetACL Request - with invalid token type of StatusCode Start: 0e0h; Response - Session \\ (A6-1-8-1-2) \; GetACL Request - with first Status token = 81h(short); Response - Pass \\ (A6-1-8-2-1) \; GetACL Request - with first Status Code = 0h(fround in status code); Response - fail \\ (A6-1-8-2-1) \; GetACL Request - with first Status Code = 0h(not in the status code); Response - fail \\ (A6-1-8-2-1) \; GetACL Request - with first Status Code = 0h(not in the status code); Response - fail \\ (A6-1-8-2-2) \; GetACL Request - with first Status Code = 0h(not in the status code); Response - fail \\ (A6-1-8-2-2) \; GetACL Request - with first Status Code = 0h; Response - Normal \\ (A6-1-8-2-2) \; GetACL Request - with first Status Code = 0h; Response - Normal \\ (A6-1-8-6-1) \; GetACL Request - with 1st Status Code = 0h(response - Normal \\ (A6-1-8-6-1) \; GetACL Request - with 1st Status Code = 0h(response - Normal \\ (A6-1-8-6-1) \; GetACL Request - with 1st Status Code = 0h(response - Session Abort \\ \end{array}$	PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Start: 0e0h; Response - Session (A6-1-8-2) GetACL Request - with first Status token = 81h(short); Response - Pasa (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasa (A6-1-8-2-1) GetACL Request - with first Status Code = 0h(fout in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with first Status Code = 0h(fout in the status code); Response - fail (A6-1-8-3-2) GetACL Request - with second Status Code = 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with status Status Code = 0h; Response - Normal (A6-1-8-6-1) GetACL Request - with 1st Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = 41h(byte); Response - Session Abort	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Start: 0e0h; Response - Session (A6-1-8-1-2) GetACL Request - with first Status token = 81h(short); Response - Pasa (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - fail (A6-1-8-2-1) GetACL Request - with first Status Code = 0h(fromt in the status code); Response - fail (A6-1-8-2-3) GetACL Request - with first Status Code = 0h; Response - Normal (A6-1-8-2) GetACL Request - with stord Status Code = 0h; Response - Normal (A6-1-8-2) GetACL Request - with stord Status Code = 0h; Response - Normal (A6-1-8-6-1) GetACL Request - with 1st Status token = 1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 41h(byte); Response - Session Abort	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Start: 0e0h; Response - Session (A6-1-8-2) GetACL Request - with first Status token = 81h(short); Response - Pasa (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - fail (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(not in the status code); Response - fail (A6-1-8-2) GetACL Request - with first Status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with store Status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with third Status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with 1st Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Session (A6-1-8-2-1) GetACL Request - with first Status token = 81h/short); Response - Pasa (A6-1-8-2-1) GetACL Request - with first Status token = 81h/short); Response - Pasi (A6-1-8-2-2) GetACL Request - with first Status Code I= 0h/(rout in status code); Response - fail (A6-1-8-2-2) GetACL Request - with first Status Code I= 0h/(rout in the status code); Response - fail (A6-1-8-2) GetACL Request - with second Status Code I= 0h; Response - Normal (A6-1-8-2) GetACL Request - with status token = 0h; Response - Normal (A6-1-8-2) GetACL Request - with status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(Integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(Integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(Integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(Integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(Integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(Integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(Integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(Integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(Integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(Integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with ard Status token = 91h(Integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with area Status token = 91h(Integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with area Status token = 91h(Integer); Response - Session Abort	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Start: 0e0h; Response - Session (A6-1-8-2) GetACL Request - with first Status token = 81h(short); Response - Pasa (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasa (A6-1-8-2-1) GetACL Request - with first Status Code I= 0h(fout in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with first Status Code I= 0h(fout in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with stord Status Code I= 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with stord Status Code I= 0h; Response - Normal (A6-1-8-6-1) GetACL Request - with 1st Status token = 0h)(finteger); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Session (Abort - 1-4-2-1(1)) GetACL Request - with unexpected token encoded inside the Param; Response - Status Code: 0Ch(invalid -Param)	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Start: 0e0h; Response - Session (A6-1-8-2) GetACL Request - with first Status token = 81h(short); Response - Pasa (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Asa (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(rot in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with first Status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with third Status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with 1st Status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with 1st Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-2-1) (1) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Session (A6-1-4-2-1(1)) GetACL Request - with unexpected token encoded inside the Params; Response - Status Code: C0h(invalid_Param) (A6-0-1-1-1) GenAC Request - with short atom for InvokingID; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Session (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasa (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasi (A6-1-8-2-2) GetACL Request - with first Status Code != 0h(not in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with first Status Code != 0h(not in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with second Status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with status token = A1h(byte); Response - Session Abort (A6-1-8-2) GetACL Request - with 1st Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with andid token type of StatusCode End: 0e0h; Response - Session (Abort (A6-1-8-1)) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Session (Abort (A6-1-1)] GetACL Request - with short atom for InvokingID; Response - Pass (A6-0-1-1)] GenKey Request - with invokingID; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Session (A6-1-8-2) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasi (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(fout in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with first Status Code != 0h(fout in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with stord Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with stord Status Code != 0h; Response - Normal (A6-1-8-6-1) GetACL Request - with 1st Status token = 0h)(finteger); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Session (Abort (A6-1-9-2-1)) GetACL Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(invalid_Param) (A6-0-1-1-1) GenKey Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) GenKey Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) GenKey Request - with short atom for InvokingID; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Session (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasa (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasi (A6-1-8-2-2) GetACL Request - with first Status Code != 0h(not in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with first Status Code != 0h(not in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with second Status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with status token = A1h(byte); Response - Session Abort (A6-1-8-2) GetACL Request - with 1st Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with andid token type of StatusCode End: 0e0h; Response - Session (Abort (A6-1-8-2-1(1)) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Session (Abort (A6-1-1-1)) GetACL Request - with short atom for InvokingID; Response - Pass (A6-0-1-1) GenKey Request - with short atom for InvokingID; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Start: 0e0h; Response - Session (A6-1-8-2) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Asia (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(not in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with first Status Code != 0h(not in the status code); Response - fail (A6-1-8-2) GetACL Request - with second Status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with third Status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with 1st Status token = 0h; Response - Session Abort (A6-1-8-2) GetACL Request - with 1st Status token = 91h(integer); Response - Session Abort (A6-1-8-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-1) GetACL Request - with 3rd Status token = 01h(integer); Response - Session Abort (A6-1-8-2-1) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Session (Abort (A6-1-9-2-1) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Session (Abort (A6-1-1)) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Session (Abort (A6-1-1-1)) GetACR Request - with medium atom for InvokinglD; Response - Pass (A6-0-1-1-1) GenKey Request - with medium atom for InvokinglD; Response - Pass (A6-0-1-1-1) GenKey Request - with medium atom for InvokinglD; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Start: 0e0h; Response - Fasi (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasi (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Fail (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(not in the status code); Response - fail (A6-1-8-2-1) GetACL Request - with first Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with second Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with third Status Code != 0h; Response - Normal (A6-1-8-6-1) GetACL Request - with 1st Status token = 0h; Response - Normal (A6-1-8-6-1) GetACL Request - with 1st Status token = 0h; Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 0h(hyte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 0h(hyte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 0h(hyte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 0h(hyte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 0h(hyte); Response - Session Abort (A6-1-8-2-1) GetACL Request - with 3rd Status token = 0h(hyte); Response - Session Abort (A6-1-8-2-1) GetACL Request - with nuexpected token encoded inside the Params; Response - Status Code: 0Ch(hrvaid); Param) (A6-0-1-1-1) GenKey Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) GenKey Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1-1) GenKey Request - with long atom for MethodID; Response - Pass (A6-0-1-1-1) GenKey Request - with long atom for NethodID; Response - Pass (A6-0-1-1-1) GenKey Request - with long atom for NethodID; Response - Pass (A6-0-1-1-1) GenKey Request - with long atom for NethodID; Response - Pass (A6-0-1-1-1) GenKey Request - with nonexistent InvokingID; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Fail (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasi (A6-1-8-2-2) GetACL Request - with first Status Code != 0h(not in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with first Status Code != 0h(not in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with second Status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with status token = A1h(byte); Response - Session Abort (A6-1-8-2) GetACL Request - with 1st Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Session (Abort (A6-1-1-1) GetACL Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid, Param) (A6-0-1-1-1) GenKey Request - with InvakingID; Response - Pass (A6-0-1-1-1) GenKey Request - wi	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Fasi (A6-1-8-2) GetACL Request - with first Status token = 81h(short); Response - Pasi (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasi (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(fout in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with first Status Code != 0h(fout in the status code); Response - fail (A6-1-8-2-3) GetACL Request - with stord Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with stord Status Code != 0h; Response - Normal (A6-1-8-6-1) GetACL Request - with 1st Status token = 0h)(fintger); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = 1h(intger); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 1h(intger); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(intger); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(intger); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(intger); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(intger); Response - Session Abort (A6-1-9-2-1) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Session (Abort (A6-1-9-2)) GetACL Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) GenKey Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1-1) GenKey Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) GenKey Request - with mong atom for NuchoingID; Response - Pass (A6-0-1-1-1) GenKey Request - with mong atom for MethodID; Response - Pass (A6-0-1-1-1) GenKey Request - with nonexistent InvokingID; Response - Session Abort (A6-1-1-2-1) GenKey Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-2-1) GenKey Reques	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Fasi (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasi (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Fail (A6-1-8-2-1) GetACL Request - with first Status Code = 0h(not in the status code); Response - fail (A6-1-8-2-1) GetACL Request - with first Status Code = 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with second Status Code = 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with status Status Code = 0h; Response - Normal (A6-1-8-6-1) GetACL Request - with 1st Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3nd Status token = 91h(integer); Response - Session Abort (A6-1-8-2-1) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Session (Abort - 1-1) GetACL Request - with sort atom for InvokingID; Response - Pass (A6-0-1-1-1) GenKey Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1-1) GenKey Request - with long atom for MethodID; Response - Pass (A6-0-1-1-1) GenKey Request - with nonexistent InvokingID; Response - Pass (A6-0-1-1-1) GenKey Request - with nonexistent InvokingID; Response - Pass (A6-0-1-1-1) GenKey Request - with nonexistent InvokingID; Response - Status Code: (Dh(Not, Authorized) (A6-1-1-3-1(2)) GenKey Request - with none-8-long token for InvokingID; Response - Status Code: (Dh(Not, Authorize	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Fail (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasi (A6-1-8-2-2) GetACL Request - with first Status Code != 0h(fout in the status code); Response - Fail (A6-1-8-2-2) GetACL Request - with first Status Code != 0h(fout in the status code); Response - Fail (A6-1-8-2-2) GetACL Request - with second Status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with status token = A1h(byte); Response - Session Abort (A6-1-8-2) GetACL Request - with 1st Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Session (Abort (A6-1-1-1)) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Status (A6-0-1-1-1) GenKey Request - with Invalid token for InvokingID; Response - Pass (A6-0-1-1-1) GenKey Request - with Invalid token for MethodID; Response - Pass (A6-0-1-1-1) GenKey Request - with Invalid token for 'Call'(FBH); Response - Pass (A6-0-1-1-1) GenKey Request - with Invalid token for 'Call'(FBH); Response - Status Code: (A6-1-1-1-1) GenKey Request - with InvokingID; Response - Status Code: (A6-1-1-1-1) GenKey Request - with InvokingID; Response - Status Code: (A6-1-1-1-1) GenKey Re	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Fasi (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasis (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasis (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(fout in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with first Status Code != 0h(fout in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with storod Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with storod Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with 1st Status token = 0h(finteger); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 1h(hyte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 1h(hyte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 1h(hyte); Response - Session Abort (A6-1-9-2-1) GetACL Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param) (A6-0-1-1.1) GenKey Request - with noing for InvokingiD; Response - Pass (A6-0-1-1.1) GenKey Request - with medium atom for InvokingiD; Response - Pass (A6-0-1-1.1) GenKey Request - with none for MethodD); Response - Pass (A6-0-1-1.1) GenKey Request - with none for MethodD); Response - Pass (A6-0-1-1.1) GenKey Request - with none for MethodD); Response - Status Code: (A6-1-1-1.1) G	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Fasi (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasis (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasis (A6-1-8-2-1) GetACL Request - with first Status Code = 0h(not in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with first Status Code = 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with second Status Code = 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with 1st Status token = 91h(integer); Response - Session Abort (A6-1-8-1) GetACL Request - with 1st Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Session (A6-0-1-1) GetACR Request - with short atom for InvokingID; Response - Pass (A6-0-1-1.1) GenKey Request - with short atom for InvokingID; Response - Pass (A6-0-1-1.1) GenKey Request - with nong atom for NethodID; Response - Pass (A6-0-1-1.1) GenKey Request - with none Xient for MethodID; Response - Pass (A6-0-1-1.1) GenKey Request - with none-Syten KokingID; Response - Status Code: (D1h(Not, Authorized)) (A6-1-1-3-1(2)) GenKey Request - with none-Sient InvokingID; Response - S	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Fasi (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasis (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasis (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(fout in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with first Status Code != 0h(fout in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with storod Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with storod Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with 1st Status token = 0h(finteger); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 1h(hyte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 1h(hyte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 1h(hyte); Response - Session Abort (A6-1-9-2-1) GetACL Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param) (A6-0-1-1.1) GenKey Request - with noing for InvokingiD; Response - Pass (A6-0-1-1.1) GenKey Request - with medium atom for InvokingiD; Response - Pass (A6-0-1-1.1) GenKey Request - with none for MethodD); Response - Pass (A6-0-1-1.1) GenKey Request - with none for MethodD); Response - Pass (A6-0-1-1.1) GenKey Request - with none for MethodD); Response - Status Code: (A6-1-1-1.1) G	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Fail (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(fout in the status code); Response - Fail (A6-1-8-2-2) GetACL Request - with first Status Code != 0h(fout in the status code); Response - Fail (A6-1-8-2-2) GetACL Request - with second Status Code != 0h; Response - Normal (A6-1-8-2-2) GetACL Request - with second Status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with status token = A1h(byte); Response - Session Abort (A6-1-8-2) GetACL Request - with 1st Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with and toten type of StatusCode End: 0e0h; Response - Sassion (A6-0-1-1.1) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Sassion (A6-0-1-1.1) GetACP Request - with nont atom for InvokingID; Response - Pass (A6-0-1-1.1) GetACP Request - with long atom for MethodID; Response - Pass (A6-0-1-1.1) GetACP Request - with invalid token for 'Call'(FBH); Response - Status Code: (A6-1-1-1.1) GetACP Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-1.1) GetACP Request - with non-8-long token for InvokingID	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Fail (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(fout in the status code); Response - Fail (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(fout in the status code); Response - Fail (A6-1-8-2-1) GetACL Request - with second Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with second Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with status token = A1h(byte); Response - Session Abort (A6-1-8-4-1) GetACL Request - with 1st Status token = A1h(byte); Response - Session Abort (A6-1-8-4-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-4-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-4-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-4-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-4-1) GetACL Request - with and Status token = 91h(integer); Response - Session Abort (A6-1-8-4-11) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Status Code: 0Ch(Invalid, Param) (A6-0-1-1-1) GenKey Request - with nont atom for InvokingID; Response - Pass (A6-0-1-1-1) GenKey Request - with invalid token for Call'(FBh); Response - Pass (A6-1-1-1-1) GenKey Request - with nonexistent InvokingID; Response - Pass (A6-1-1-1-1) GenKey Request - with nonexistent InvokingID; Response - Status Code: (A6-1-1-1) GenKey Request - with non-8-long token for InvokingID; Response - Status Code: (A1-1-1-1) GenKey Request - with non-8-long token	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Fail (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasil (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(fout in the status code); Response - fail (A6-1-8-2-1) GetACL Request - with scond Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with status token = 0h); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-9-2-1) GetACL Request - with short atom for InvokingiD; Response - Pass (A6-0-1-1) GenKey Request - with nonexpected token encoded inside the Params; Response - Status Code: 0Ch(invalid_Param) (A6-0-1-1) GenKey Request - with nong atom for InvokingiD; Response - Pass (A6-0-1-1) GenKey Request - with none for MethodD; Response - Pass (A6-0-1-1) GenKey Request - with nonexistent InvokingiD; Response - Status Code: (A6-1-1-2.1) GenKey Request - with none-8-long token for InvokingiD; Response - Status Code: (Ah(t-1-2	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with Invalid token type of StatusCode Stat: 0e0h; Response - Fail (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status Code I= 0h(found in status code); Response - fail (A6-1-8-2-2) GetACL Request - with first Status Code I= 0h(found in status code); Response - fail (A6-1-8-2) GetACL Request - with second Status Code I= 0h; Response - Normal (A6-1-8-2) GetACL Request - with second Status Code I= 0h; Response - Normal (A6-1-8-2) GetACL Request - with status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with andi token type of StatusCode End. Ce0h; Response - Session (A6-0-1-11) GenKey Request - with none for InvokingID; Response - Pass (A6-0-1-11) GenKey Request - with none for InvokingID; Response - Pass (A6-0-1-11) GenKey Request - with Involid token for Call'(FBH); Response - Pass (A6-0-1-11) GenKey Request - with none for InvokingID; Response - Pass (A6-0-1-11) GenKey Request - with none-byte token for InvokingID; Response - Status Code: (A6-1-1-3-1(2)) GenKey Request - with non-byte token for InvokingID; Response - Status Code: (A6-1-1-3-1(2)) GenKey Request - with non-byte token for InvokingID; Response - Status Code: (A6-1-2-1)(2) GenKey Request - with non-b	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Fail (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasi (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(fout in the status code); Response - fail (A6-1-8-2-1) GetACL Request - with Second Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with second Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with 1st Status token = A1h(byte); Response - Session Abort (A6-1-8-1) GetACL Request - with 1st Status token = A1h(byte); Response - Session Abort (A6-1-8-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with and status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Session (A6-0-1-1.1) GetACL Request - with nonexitent for InvokingID; Response - Pass (A6-0-1-1.1) GenKey Request - with nonexitent for InvokingID; Response - Pass (A6-0-1-1.1) GenKey Request - with nonexitent for MethodID; Response - Pass (A6-1-1-1.1) GenKey Request - with nonexitent for MethodID; Response - Status Code:: Oth(Not, Authorized) (A6-1-2-3-1(2)) GenKey Request - with nonexitent for InvokingID; Response -	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Fail (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasi (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(fout in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with scond Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with third Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with 1st Status token = 0h; Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with and Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with nonexpected token encoded inside the Params; Response - Status Code: 0Ch(invalid_Param) (A6-0-1-1.1) GenKey Request - with none for InvokingiD; Response - Pass (A6-0-1-1.1) GenKey Request - with none for MethodD; Response - Pass (A6-0-1-1.1) GenKey Request - with none with for MethodD; Response - Status Code: 01h(Not_Authorized) (A6-1-1-2.1) GenKey Request - with none-8-long token for InvokingiD; Response - Status Code: 01h(Not_Authorized) (A6-1-2-1.1) GenKey Request - with none-8-long tok	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with Invalid token type of StatusCode Stat: 0e0h; Response - Fail (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status Code I= 0h(found in status code); Response - fail (A6-1-8-2-2) GetACL Request - with first Status Code I= 0h(found in status code); Response - fail (A6-1-8-2) GetACL Request - with second Status Code I= 0h; Response - Normal (A6-1-8-2) GetACL Request - with second Status Code I= 0h; Response - Normal (A6-1-8-2) GetACL Request - with status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-2-1) GetACL Request - with andi token type of StatusCode End: 0e0h; Response - Session (A6-0-1-1) GenKey Request - with novingID; Response - Pass (A6-0-1-1) GenKey Request - with novingID; Response - Pass (A6-0-1-1) GenKey Request - with novingID; Response - Pass (A6-0-1-1) GenKey Request - with novisitent InvokingID; Response - Status Code: (A6-1-1-1) GenKey Request - with non-Byte token for InvokingID; Response - Status Code: (A6-1-1-1) GenKey Request - with non-Byte token for InvokingID; Response - Status Code: (A6-1-2-1) GenKey Request - with non-Byte token for InvokingID; Response - Status Code: (A6-1-2-1) GenKey Request - with non-byte token for InvokingID; Response - Status Code: (A6	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Fail (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasi (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(not in the status code); Response - fail (A6-1-8-2-1) GetACL Request - with first Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with second Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with status token = A1h(byte); Response - Session Abort (A6-1-8-4-1) GetACL Request - with 1st Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with and status token = 01h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Status Code: 0Ch(lowalid_Param) (A6-0-1-1.1) GenKey Request - with nonexitent for InvokingID; Response - Pass (A6-0-1-1.1) GenKey Request - with long atom for InvokingID; Response - Pass (A6-0-1-1.1) GenKey Request - with nonexitent InvokingID; Response - Pass (A6-0-1-1.1) GenKey Request - with nonexitent InvokingID; Response - Status Code: 01h(Not, Authorized) (A6-1-2-3-1(2)) GenKey Request - with non-8-long token for InvokingID; Response - Status Code: 01h(Not, Authorized) (A6-1-2-3-1(2)) GenKey Request - with non-8-long token for MethodID; Response - Status Code: 01h(Not, Authorized) (A6-1-2-3-1) GenKey Reque	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with Invalid token type of StatusCode Stat: 0e0h; Response - Fail (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status Code I= 0h(found in status code); Response - fail (A6-1-8-2-2) GetACL Request - with first Status Code I= 0h(found in status code); Response - fail (A6-1-8-2) GetACL Request - with second Status Code I= 0h; Response - Normal (A6-1-8-2) GetACL Request - with second Status Code I= 0h; Response - Normal (A6-1-8-2) GetACL Request - with status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = A1h(byte); Response - Session Abort (A6-1-2-1) GetACL Request - with andi token type of StatusCode End: 0e0h; Response - Session (A6-0-1-1) GenKey Request - with novingID; Response - Pass (A6-0-1-1) GenKey Request - with novingID; Response - Pass (A6-0-1-1) GenKey Request - with novingID; Response - Pass (A6-0-1-1) GenKey Request - with novisitent InvokingID; Response - Status Code: (A6-1-1-1) GenKey Request - with non-Byte token for InvokingID; Response - Status Code: (A6-1-1-1) GenKey Request - with non-Byte token for InvokingID; Response - Status Code: (A6-1-2-1) GenKey Request - with non-Byte token for InvokingID; Response - Status Code: (A6-1-2-1) GenKey Request - with non-byte token for InvokingID; Response - Status Code: (A6	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Fail (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasi (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(fout in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with second Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with third Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with 1st Status token = 0h)(finteger); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 1h(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3nd Status token = 91h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with and to token type of StatusCode End: 0e0h; Response - Session (A6-0-1-1) GetACL Request - with nonexpected token encoded inside the Params; Response - Status Code: 0Ch(invalid_Param) (A6-0-1-1) GenKey Request - with nong atom for InvokinglD; Response - Pass (A6-0-1-1) GenKey Request - with none for InvokinglD; Response - Pass (A6-0-1-1) GenKey Request - with nonexistent InvokinglD; Response - Status Code: (Ih(Not_Authorized) (A6-1-1-2.1) GenKey Request - with none-8-long token for InvokinglD; Response - Status Code: (Ih(Not_Authorized) (A6-1-2-1.1	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Fail (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasil (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(fout in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with third Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with storod Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with 1st Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 1h(hyte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 1h(hyte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with nonexpected token encoded inside the Params; Response - Status Code: 0Ch(invalid_Param) (A6-0-1-1.1) GenKey Request - with none for InvokinglD; Response - Pass (A6-0-1-1.1) GenKey Request - with none for InvokinglD; Response - Pass (A6-0-1-1.1) GenKey Request - with none for MethodD; Response - Pass (A6-0-1-1.1) GenKey Request - with none stent InvokinglD; Response - Status Code: (A6-1-1-1.1) GenKey Request - with none-8-long token for InvokinglD; Response - Status Code: (A6-1-1-1.2.1) GenKey Request - with none-8-long token for InvokinglD; Response - Status Code: (PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with first Status token = Shi/short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status token = Shi/short); Response - Pasi (A6-1-8-2-2) GetACL Request - with first Status Code != 0h(found in status code); Response - fail (A6-1-8-2) GetACL Request - with first Status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with second Status Code != 0h; Response - Normal (A6-1-8-2) GetACL Request - with status token = Ahl(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = Ahl(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = Ahl(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = Ahl(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = Ahl(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = Ahl(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = Ahl(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = Ahl(byte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with novalid token type of StatusCode End: 0e0h; Response - Status Code: COh(Invalid, Param) (A6-0-1-1) GenKey Request - with novalingID; Response - Pass (A6-0-1-1) GenKey Request - with novalid token for 'LovekingID; Response - Pass (A6-0-1-1) GenKey Request - with novalid token for 'LovekingID; Response - Status Code: (A6-1-1-1) GenKey Request - with non-B-long token for InvokingID; Response - Status Code: (A6-1-1-1) GenKey Request - with non-B-long token for InvokingID; Response - Status Code: (A6-1-1-1) GenKey Request - with non-B-long token for InvokingID; Response - Status Code:	PASS PASS PASS PASS PASS PASS PASS PASS
 (A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort (A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Stat: 0e0h; Response - Fail (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pass (A6-1-8-2-1) GetACL Request - with first Status token = 81h(short); Response - Pasil (A6-1-8-2-1) GetACL Request - with first Status Code != 0h(fout in the status code); Response - fail (A6-1-8-2-2) GetACL Request - with third Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with storod Status Code != 0h; Response - Normal (A6-1-8-3-2) GetACL Request - with 1st Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 1st Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 1h(integer); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 2nd Status token = 1h(hyte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with 3rd Status token = 1h(hyte); Response - Session Abort (A6-1-8-6-1) GetACL Request - with nonexpected token encoded inside the Params; Response - Status Code: 0Ch(invalid_Param) (A6-0-1-1.1) GenKey Request - with none for InvokinglD; Response - Pass (A6-0-1-1.1) GenKey Request - with none for InvokinglD; Response - Pass (A6-0-1-1.1) GenKey Request - with none for MethodD; Response - Pass (A6-0-1-1.1) GenKey Request - with none stent InvokinglD; Response - Status Code: (A6-1-1-1.1) GenKey Request - with none-8-long token for InvokinglD; Response - Status Code: (A6-1-1-1.2) GenKey Request - with none-8-long token for InvokinglD; Response - Status Code: (A6	PASS PASS PASS PASS PASS PASS PASS PASS

(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 != Oh(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 != Oh; 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) 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) (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: OCh (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: OCh (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 N/A Status Code: 0Ch (Invalid Param) (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) 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: OCh (Invalid_Param) PASS (A12-0-1-1-2) SPInfo RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: OCh (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: OCh (A12-3-1-6-6) SPInfo RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code:	PASS
0Ch (Invalid_Param)	PASS
(A12-3-1-6-10) SPInfo RequiredParams: Get without 'StartCol' component; Get response - Pass (A12-3-1-7-6) SPInfo RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code:	PASS
	PASS PASS
(A12-3-1-7-10) SPInfo RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response -	
Status Code: OCh (Invalid_Param) (A12-3-1-7-10(2)) SPInfo RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response	PASS
- Status Code: 0Ch (Invalid_Param) (A12-0-1-1-1) SPTemplates RequiredParams: Get with 'Table' component; Get response - Status	PASS
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) (A12-3-1-5-1) SPTemplates RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch	PASS
(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 - (A12-3-1-7-6) SPTemplates RequiredParams: Get with 'EndCol' > maximum; Get response - Status	PASS
Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-9) SPTemplates RequiredParams: Get without 'EndCol' component; Get response - Pass (A12-3-1-7-10) SPTemplates RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get	PASS
response - Status Code: 0Ch (Invalid_Param) (A12-3-1-7-10(2)) SPTemplates RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get	PASS
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	DACC
response - Status Code: 0Ch (Invalid_Param) (A12-0-1-1-2) MethodID RequiredParams: Get with 'EndColumn' component encoded twice; Get	PASS
response - Status Code: 0Ch (Invalid_Param) (A12-3-1-4-1) MethodID RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch	PASS
(Invalid_Param)	PASS
(A12-3-1-5-1) MethodID RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch (A12-3-1-6-6) MethodID RequiredParams: Get with 'StartCol' > maximum; Get response - Status	PASS
Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-6-10) MethodID RequiredParams: Get without 'StartCol' component; Get response - Pass (A12-3-1-7-6) MethodID RequiredParams: Get with 'EndCol' > maximum; Get response - Status	PASS
	PASS PASS
(A12-3-1-7-10) MethodID RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get	
response - Status Code: 0Ch (Invalid_Param) (A12-3-1-7-10(2)) MethodID RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get	PASS
response - Status Code: OCh (Invalid_Param) (A12-0-1-1-1) ACE RequiredParams: Get with 'Table' component; Get response - Status Code: OCh	PASS
(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 -	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: OCh (A12-3-1-6-6) ACE RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: OCh	PASS
(Invalid_Param)	PASS
(A12-3-1-6-10) ACE RequiredParams: Get without 'StartCol' component; Get response - Pass (A12-3-1-7-6) ACE RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: OCh	PASS
(PASS PASS
(A12-3-1-7-10) ACE RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response -	
Status Code: 0Ch (Invalid_Param) (A12-3-1-7-10(2)) ACE RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response -	PASS
Status Code: 0Ch (Invalid_Param)	PASS
	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 Required Params: Get with 'EndColumn' component encoded twice; Get	
response - Status Code: 0Ch (Invalid_Param) (A12-3-1-4-1) Authority RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch	PASS
	PASS PASS
(A12-3-1-6-6) Authority Required Params: Get with 'StartCol' > maximum; Get response - Status	
	PASS PASS
(A12-3-1-7-6) Authority Required Params: Get with 'EndCol' > maximum; Get response - Status Code:	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 Required Params: Get with the number of 'StartCol' > 'EndCol'; Get	
response - Status Code: 0Ch (Invalid_Param) (A12-0-1-1-1) C_PIN RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch	PASS
	PASS
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 PASS
(A12-3-1-6-6) C_PIN RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code:	
	PASS PASS

(A12-3-1-7-6) C_PIN RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code:	
OCh (Invalid_Param) (A12-3-1-7-9) C_PIN RequiredParams: Get without 'EndCol' component; Get response - Pass	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:	17100
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	FA33
response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-4-1) TPerInfo RequiredParams: Get with 'StartRow'; Get response - Status Code: OCh	PASS
(A12-3-1-5-1) TPerInfo RequiredParams: Get with 'EndRow'; Get response - Status Code: OCh (A12-3-1-6-6) TPerInfo RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code:	PASS
OCh (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) (A12-3-1-7-9) TPerInfo RequiredParams: Get without 'EndCol' component; Get response - Pass	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	DACC
response - Status Code: 0Ch (Invalid_Param) (A12-0-1-1-1) Template RequiredParams: Get with 'Table' component; Get response - Status Code:	PASS
OCh (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: OCh (Invalid Param)	PASS
(A12-3-1-4-1) Template RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch	17,55
(Invalid_Param)	PASS
(A12-3-1-5-1) Template Required Params: Get with 'EndRow'; Get response - Status Code: OCh	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 Required Params: 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 (A12-3-1-7-10) Template RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get	PASS
response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-10(2)) Template RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get	17100
response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-1) SP RequiredParams: Get with 'Table' component; Get response - Status Code: OCh	DACC
(Invalid_Param) (A12-0-1-1-2) SP RequiredParams: Get with 'StartColumn' component encoded twice; Get response -	PASS
Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) SP RequiredParams: Get with 'EndColumn' component encoded twice; Get response -	
Status Code: OCh (Invalid_Param)	PASS
(A12-3-1-4-1) SP RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch (A12-3-1-5-1) SP RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch	PASS PASS
(A12-3-1-6-6) SP RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: Och	17,55
(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 PASS
(A12-3-1-7-9) SP RequiredParams: Get without 'EndCol' component; Get response - Pass (A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response -	
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid_Param)	
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: OCh (Invalid_Param) (A12-3-1-7-10(2)) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response -	PASS PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: GCh (Invalid_Param) (A12-3-1-7-10(2)) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param)	PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: OCh (Invalid_Param) (A12-3-1-7-10(2)) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response -	PASS PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: GCh (Invalid_Param) (A12-3-1-7-10(2)) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) 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 (A13-2-1-2-6) DataStore OptParams-where: Set with 'Where' > limit of the table; Set response -	PASS PASS PASS PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: OCh (Invalid_ Param) (A12-3-1-7-10(2)) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_ Param) 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 (A13-2-1-2-6) DataStore OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param)	PASS PASS PASS PASS PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: OCh (Invalid_ Param) (A12-3-1-7-10(2)) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_ Param) 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 (A13-2-1-2-6) DataStore OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: OCh (Invalid_ Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass	PASS PASS PASS PASS PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: OCh (Invalid_ Param) (A12-3-1-7-10(2)) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_ Param) 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 (A13-2-1-2-6) DataStore OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param)	PASS PASS PASS PASS PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: CDCh (Invalid_ Param) (A12-3-1-7-10(2)) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) 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 (A13-2-1-2-6) DataStore OptParams-where: Set with "Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Status Code: Och (Invalid_Param) (A13-2-1-3-5) DataStore OptParams-value: Set with data whthin limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Status Code: OCh (Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: GCh (Invalid_Param) (A12-3-1-7-10)(S) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) A13: Stet) - Byte Table Grammar check (A13-2-1-2-5) DataStore OptParams-where: Set with the limit of the byte table; Set response - pass (A13-2-1-2-6) DataStore OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-5) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthin limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: OCh (Invalid_Param) (A12-3-1-7-10(2)) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) A13: Set() - Byte Table Grammar check (A13-2-1-2-5) DataStore OptParams-where: Set with the limit of the byte table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-6) DataStore OptParams-value: Set with data whthin limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: GCh (Invalid_Param) (A12-3-1-7-10)(S) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) A13: Stet) - Byte Table Grammar check (A13-2-1-2-5) DataStore OptParams-where: Set with the limit of the byte table; Set response - pass (A13-2-1-2-6) DataStore OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-5) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthin limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: CDCh (Invalid_ Param) (A12-3-1-7-10(2)) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: CDCh (Invalid_Param) 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 (A13-2-1-2-6) DataStore OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: CDCh (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthin limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with helmst = 0 for 'Values' parameter; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with the limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with the limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with the limit of the table; Set response - (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - Status Code: CDC (Invalid_ Param)	PASS PASS PASS PASS PASS PASS PASS PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: CDCh (Invalid_Param) (A12-3-1-7-10(2)) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: CDCh (Invalid_Param) 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 (A13-2-1-2-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Status Code: CDCh (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set with data whthin limit of the table; Set response - (A13-2-1-3-5) DataStore OptParams-where: Set with data whthin limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with he limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-1-3-9) DataStore OptParams-value: Set with he limit of the byte table; Set response - (A13-2-1-3-1-3-9) DataStore OptParams-value: Set with he limit of the table; Set response - (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the table; Set response - pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the table; Set response - Status Code: COCh (Invalid_Param) (A13-2-1-2-3) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-2-3) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-2-3) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) MBR OptParams-where: Set with data whthin limit of the table; Set response - Pass	PASS PASS PASS PASS PASS PASS PASS N/A N/A
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: CCh (Invalid_Param) (A12-3-1-7-10) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: CCh (Invalid_Param) A13: Stet] - Byte Table Grammar check (A13-2-1-2-5) DataStore OptParams-where: Set with the limit of the table; Set response - pass (A13-2-1-2-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-2-1-3-5) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthin limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: CCh (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - (A13-2-1-2-6) MBR OptParams-where: Set with Where' > limit of the table; Set response - Status Code: CCh (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set with Where' > limit of the table; Set response - Status Code: CCh (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set with Where' parameter; Set response - Status (A13-2-1-3-5) MBR OptParams-value: Set with data whthin limit of the table; Set response - Status (A13-2-1-3-5) MBR OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-5) MBR OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-5) MBR OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-5) MBR OptParams-value: Set with data whthint limit of the table; Set response - Pass	PASS PASS PASS PASS PASS PASS PASS N/A N/A N/A N/A
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: CDCh (Invalid_Param) (A12-3-1-7-10(2)) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: CDCh (Invalid_Param) 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 (A13-2-1-2-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Status Code: CDCh (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set with data whthin limit of the table; Set response - (A13-2-1-3-5) DataStore OptParams-where: Set with data whthin limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with he limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-1-3-9) DataStore OptParams-value: Set with he limit of the byte table; Set response - (A13-2-1-3-1-3-9) DataStore OptParams-value: Set with he limit of the table; Set response - (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the table; Set response - pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the table; Set response - Status Code: COCh (Invalid_Param) (A13-2-1-2-3) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-2-3) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-2-3) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) MBR OptParams-where: Set with data whthin limit of the table; Set response - Pass	PASS PASS PASS PASS PASS PASS PASS N/A N/A N/A
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: CCh (Invalid_Param) (A12-3-1-7-10)(S) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) 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 (A13-2-1-2-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-2-1-2-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set with Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set with Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS N/A N/A N/A N/A N/A
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: OCh (Invalid_Param) (A12-3-1-7-10)(2) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) A13: Set() - Byte Table Grammar check (A13-2-1-2-5) DataStore OptParams-where: Set with the limit of the table; Set response - pass (A13-2-1-2-6) DataStore OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-5) DataStore OptParams-value: Set with data whthin limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with be limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-5) MBR OptParams-where: Set with Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-9) MBR OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-1) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-1) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-1-3-1) MBR OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS N/A N/A N/A N/A N/A
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: CDCh (Invalid_Param) (A12-3-1-7-10(2)) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: CDCh (Invalid_Param) 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 (A13-2-1-2-9) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-2-1-2-9) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-2-1-2-9) DataStore OptParams-where: Set with data whthiu limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: CDC (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: CDC (Invalid_Param) (A13-2-1-2-5) MBR OptParams-value: Set with he limit of the byte table; Set response - pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the table; Set response - pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-where: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-5) MBR OptParams-where: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-9) MBR OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-1) MBR OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (PASS PASS PASS PASS PASS PASS PASS N/A N/A N/A N/A N/A
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: OCh (Invalid_Param) (A12-3-1-7-10)(2) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) A13: Set() - Byte Table Grammar check (A13-2-1-2-5) DataStore OptParams-where: Set with the limit of the table; Set response - pass (A13-2-1-2-6) DataStore OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-5) DataStore OptParams-value: Set with data whthin limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with be limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-5) MBR OptParams-where: Set with Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-9) MBR OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-1) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-1) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-1-3-1) MBR OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS N/A N/A N/A N/A N/A
(A12-31-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: CDCh (Invalid_Param) (A12-31-7-10)(S) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) A13: Set() - Byte Table Grammar check (A13-21-25) DataStore OptParams-where: Set with the limit of the byte table; Set response - pass (A13-21-2-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-21-2-9) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-21-2-9) DataStore OptParams-where: Set with data whthin limit of the table; Set response - (A13-21-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-21-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-21-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-21-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-21-2-5) MBR OptParams-where: Set with the limit of the table; Set response - pass (A13-21-3-6) MBR OptParams-where: Set with the limit of the table; Set response - Pass (A13-21-3-5) MBR OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-21-3-5) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-21-3-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-21-3-1-3) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-21-3-1) MBR OptParams-value: Set with Where' parameter; Set response - Status Code: OCh (Invalid_Param) (A13-41-2-1) Authority OptParams-where: Set with Where' parameter; Set response - Status Code: OCh (Invalid_Param) (A13-41-2-1) Authority OptParams-where: Set with ColumName-Value which indicate the same ce	PASS PASS PASS PASS PASS PASS PASS N/A N/A N/A N/A N/A N/A PASS PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: CCh (Invalid_Param) (A12-3-1-7-10(2)) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) 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 (A13-2-1-2-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-2-1-3-5) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-2-1-3-5) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set with Where' parameter; Set response - Pass (A13-2-1-2-9) MBR OptParams-where: Set with Out 'Where' parameter; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-1) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-2-1) Authority OptParams-where: Set with Vbere' parameter; Set response - Status Code: OCh (Invalid_Param) (A13-4-1-4-1) Authority OptParams-where: Set with Vbere' parameter; Set response - Status Code: OC	PASS PASS PASS PASS PASS PASS PASS PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: OCh (Invalid_Param) (A12-3-1-7-10) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) A13: Sett) - Byte Table Grammar check (A13-2-1-2-6) DataStore OptParams-where: Set with the limit of the table; Set response - pass (A13-2-1-2-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-2-1-2-6) DataStore OptParams-where: Set with UWhere' parameter; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthin limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with Mere' > limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-6) MBR OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-2-1-2-6) MBR OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-5) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-4-1-2-1) Authority OptParams-where: Set with Where' parameter; Set response - Status Code: OCh (Invalid_Param) (A13-4-1-2-1) Juthority OptParams-where: Set with Where' parameter; Set response - Status Code: OCh (Invalid_Param) (A13-4-1-2-1) Jucking OptParams-where: Set with Where' parameter; Set response - Status Code: OCh (Invalid_Param) (A13-4	PASS PASS PASS PASS PASS PASS PASS N/A N/A N/A N/A N/A N/A N/A PASS PASS PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: CCh (Invalid_Param) (A12-3-1-7-10(2)) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) 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 (A13-2-1-2-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-2-1-3-5) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-2-1-3-5) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set with Where' parameter; Set response - Pass (A13-2-1-2-9) MBR OptParams-where: Set with Out 'Where' parameter; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-1) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-2-1) Authority OptParams-where: Set with Vbere' parameter; Set response - Status Code: OCh (Invalid_Param) (A13-4-1-4-1) Authority OptParams-where: Set with Vbere' parameter; Set response - Status Code: OC	PASS PASS PASS PASS PASS PASS PASS N/A N/A N/A N/A N/A N/A PASS PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: OCh (Invalid_Param) (A12-3-1-7-10) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) A13: Sett) - Byte Table Grammar check (A13-2-1-2-6) DataStore OptParams-where: Set with the limit of the table; Set response - pass (A13-2-1-2-6) DataStore OptParams-where: Set with 'Where' > limit of the table; Set response - Pass (A13-2-1-2-6) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthin limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-6) MBR OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with Where' parameter; Set response - Pass (A13-2-1-2-6) MBR OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2+1-3-1) MBR OptParams-value: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: OCh (Invalid_Param) (A13-4+1-2+1) Locking OptParams-where: Set with Where' parameter; Set response - Status Code: OCh (Invalid_Param) (A13-4+1-4+1) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set respo	PASS PASS PASS PASS PASS PASS PASS N/A N/A N/A N/A N/A N/A N/A PASS PASS PASS
(A12-31-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: CCh (Invalid_Param) (A12-31-7-10)(2) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) A13: Set() - Byte Table Grammar check (A13-21-2-5) DataStore OptParams-where: Set with the limit of the byte table; Set response - pass (A13-21-2-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-21-3-5) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-21-3-5) DataStore OptParams-where: Set with data whthin limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-21-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-21-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-21-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-21-3-6) MBR OptParams-where: Set with the limit of the table; Set response - Pass (A13-21-3-5) MBR OptParams-where: Set with Where' Parameter; Set response - Pass (A13-21-3-5) MBR OptParams-where: Set with data whthout limit of the table; Set response - Pass (A13-21-3-5) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-21-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-21-3-9) MBR OptParams-value: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: OCh (Invalid_Param) (A13-41-41) Authority OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: OCh (Invalid_Param) (A13-41-41-41) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: OCh (Invalid_Param) (A13-41-41-	PASS PASS PASS PASS PASS PASS PASS PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol', Get response - Status Code: CCh (Invalid_Param) (A12-3-1-7-10)(S) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) 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 (A13-2-1-2-6) DataStore OptParams-where: Set with 'Where' > limit of the table; Set response - Pass (A13-2-1-2-6) DataStore OptParams-where: Set with Out 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set with Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set with Out 'Where' parameter; Set response - Pass (A13-2-1-2-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-1) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-1) MBR OptParams-value: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: OCh (Invalid_Param) (A13-4-1-4-15) Authority OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: OCh (Invalid_P	PASS PASS PASS PASS PASS PASS PASS N/A N/A N/A N/A N/A N/A PASS PASS PASS PASS PASS PASS
(A12-31-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: CCh (Invalid_Param) (A12-31-7-10)(2) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) A13: Set() - Byte Table Grammar check (A13-21-2-5) DataStore OptParams-where: Set with the limit of the byte table; Set response - pass (A13-21-2-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-21-3-5) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-21-3-5) DataStore OptParams-where: Set with data whthin limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-21-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-21-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-21-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-21-3-6) MBR OptParams-where: Set with the limit of the table; Set response - Pass (A13-21-3-5) MBR OptParams-where: Set with Where' Parameter; Set response - Pass (A13-21-3-5) MBR OptParams-where: Set with data whthout limit of the table; Set response - Pass (A13-21-3-5) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-21-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-21-3-9) MBR OptParams-value: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: OCh (Invalid_Param) (A13-41-41) Authority OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: OCh (Invalid_Param) (A13-41-41-41) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: OCh (Invalid_Param) (A13-41-41-	PASS PASS PASS PASS PASS PASS PASS PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol', Get response - Status Code: OCh (Invalid_Param) (A12-3-1-7-10) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) A13: Set[) - Byte Table Grammar check (A13-2-1-2-6) DataStore OptParams-where: Set with the limit of the byte table; Set response - pass (A13-2-1-2-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set with Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set with Out 'Where' parameter; Set response - Pass (A13-2-1-2-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-2-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-2-1) MBR OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-4-1-4-1) Authority OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: OCh (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response -	PASS PASS PASS PASS PASS PASS PASS N/A N/A N/A PASS PASS PASS PASS PASS PASS PASS PA
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol', Get response - Status Code: OCh (Invalid_Param) (A13-31-7-10) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) (A13-21-25) DataStore OptParams-where: Set with the limit of the byte table; Set response - pass (A13-21-2-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-21-2-5) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-21-3-2) DataStore OptParams-where: Set without 'Where' parameter; Set response - (A13-21-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-21-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: CCh (Invalid_Param) (A13-21-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - (A13-21-3-6) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Status Code: CCh (Invalid_Param) (A13-21-3-6) MBR OptParams-where: Set with Where' > Inmit of the table; Set response - Status Code: CCh (Invalid_Param) (A13-21-3-6) MBR OptParams-where: Set with Out 'Where' parameter; Set response - Pass (A13-21-3-6) MBR OptParams-value: Set with data whthut limit of the table; Set response - Pass (A13-21-3-9) MBR OptParams-value: Set with data whthut limit of the table; Set response - Status Code: CCh (Invalid_Param) (A13-41-41-3) Authority OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: CCh (Invalid_Param) (A13-41-41-41) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: CCh (Invalid_Param) (A13-41-41-41) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: CCh (Invalid_Param) (A13-41-41-41) Locking	PASS PASS PASS PASS PASS PASS PASS N/A N/A N/A N/A N/A N/A N/A N/A PASS PASS PASS PASS PASS PASS PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol', Get response - Status Code: OCh (Invalid_Param) (A12-3-1-7-10) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: OCh (Invalid_Param) A13: Set[) - Byte Table Grammar check (A13-2-1-2-6) DataStore OptParams-where: Set with the limit of the byte table; Set response - pass (A13-2-1-2-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set with Where' > limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set with Out 'Where' parameter; Set response - Pass (A13-2-1-2-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-2-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-2-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-2-1) MBR OptParams-value: Set with data whthout limit of the table; Set response - Status Code: OCh (Invalid_Param) (A13-4-1-4-1) Authority OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: OCh (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response -	PASS PASS PASS PASS PASS PASS PASS PASS

(A14-1-3-2-8) Table OptParams-where: Next with nonexistent UID; Next response - Status Code: 0Ch PASS (Invalid Param) (A14-1-3-2-11) Table OptParams-where: Next with omitted 'Where' parameter; Next response - first PASS UID in the table (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 OCh (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: OCh (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 DASS (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 OCh (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: OCh (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 OCh (Invalid Param) PASS (A14-1-3-2-11) Template OntParams-where: Next with omitted 'Where' parameter: Next response -PASS first UID in the table (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 PASS snonse - 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 RegParams: 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 PASS response - Pase (A15-1-2-2-1(2)) SPTemplates ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pas PASS (A15-1-2-2-1(2)) SPTemplates RegParams-methodID: GetACL with long atom for MethodID: GetACL PASS response - Pass (A15-1-2-3-1) SPTemplates RegParams; 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
$({\tt A15-1-2-2-1(2)}) \ {\tt MethodID} \ {\tt ReqParams-methodID}: {\tt GetACL} \ {\tt with} \ {\tt medium} \ {\tt atom} \ {\tt for} \ {\tt MethodID}; \ {\tt GetACL} \ {\tt optimal} \ {\tt GetACL} \ {\tt atom} \ {\tt atom} \ {\tt for} \ {\tt MethodID}; \ {\tt GetACL} \ {\tt for} \ {\tt atom} \ $	PASS
response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with long atom for MethodID; GetACL	
response - Pass (A15-1-2-3-1) MethodID ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID';	PASS
GetACL response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status	PASS
Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL	PASS
response - Pass (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL	PASS PASS
(A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with medium atom for MethodID; GetACL (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response -	PASS 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	PASS
response - Pass (A15-1-2-2-1(2)) Authority ReqParams-methodID: GetACL with long atom for MethodID; GetACL	
response - Pass (A15-1-2-3-1) Authority ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID';	PASS
GetACL response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) C_PIN Condition: GetACL without UID of access control table; GetACL response -	PASS
Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) C_PIN ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL	PASS
response - Pass (A15-1-2-1-1(2)) C_PIN ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL	PASS 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 (A15-1-2-3-1) C_PIN ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL	PASS
response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) TPerInfo Condition: GetACL without UID of access control table; GetACL response -	PASS
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	PASS
response - Pass (A15-1-2-1-1(2)) Template ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL	
response - Pass (A15-1-2-2-1(2)) Template ReqParams-methodID: GetACL with medium atom for MethodID; GetACL	PASS
response - Pass (A15-1-2-2-1(2)) Template ReqParams-methodID: GetACL with long atom for MethodID; GetACL	PASS
response - Pass (A15-1-2-3-1) Template ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID';	PASS
GetACL response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) SP Condition: GetACL without UID of access control table; GetACL response - Status	PASS
Code: 01h (Not_Authority)	PASS
(A15-1-2-1-1(2)) SP ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL (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 - (A15-1-2-3-1) SP ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL	PASS
response - Status Code: 01h (Not_Authority) A19: ReverSP() Grammar check	PASS
(A19-1-3-1-10) KeepGlbRange: RevertSP to LockingSP with the omitted KeepGlobalRangeKey;	
RevertSP Response - Pass Revert LockingSP	NM
Start Session with HostChallenge - AdminSP Sync Session - AdminSP	PASS PASS
LockingSP.Revert - Request	PASS
LockingSP.Revert - Response End Session - Request	PASS PASS
End Session - Response Activating the Locking SP	PASS
Start Session with HostChallenge - AdminSP	PASS
Sync Session - AdminSP	PASS
Activate_LockingSP	PASS
Activate_LockingSP - Response	PASS
Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response	PASS 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	1 435

(C2) Properties Parameter and Host Properties Parameter	PASS
(C2) Check TPer properties (C2(1)) Check Host properties	PASS PASS
C3: Get() Byte Table: DataStore and MBR Contents check	FA33
(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 (C3-24) Data Comparison from MBR table	N/A N/A
C4: Next() Table Contents (AdminSP)	N/A
(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 (C4-6) Verify UIDs for ACE Table	PASS 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 (C4-20) Next() - C_PIN Table	PASS 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	
(D1-1-1-9) ACE.Set Grammar: Request with right params; Set response - pass	PASS
(D1-1-1-10) ACE.Set Grammar: Request with non-parsed boolean expression form; Set response -	
Session abort	PASS
(D1-1-1-11) 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
Code: 0Ch (Invalid_Param)	N/A PASS
Code: 0Ch (Invalid_Param) (D1-1-2-1-1) ACE.Get is issued to verify; Data comparison - Matching	PASS
Code: OCh (Invalid_Param) (D1-1-2-1-1) ACE.Get is issued to verify; Data comparison - Matching (D1-1-2-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching	PASS PASS PASS
Code: OCh (Invalid_Param) (D1-1-2-1-1) ACE.Get is issued to verify; Data comparison - Matching (D1-1-2-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-1-3-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-1-3-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value	PASS PASS
Code: OCh (Invalid_Param) (D1-12-1-1) ACE.Get is issued to verify; Data comparison - Matching (D1-12-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing	PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-1-2-1-1) ACE. Get is issued to verify; Data comparison - Matching (D1-1-2-1-2) ACE. Set with different UIDs; ACE. Get is issued to verify data - Matching (D1-1-3-1) ACE. Set in a transaction with endTransaction status = 0; The new value retains the set (D1-1-3-1-2) ACE. Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-1-2-1-1) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column	PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-1-2-1-1) ACE.Get is issued to verify; Data comparison - Matching (D1-1-2-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-1-3-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-1-3-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-1-2-1-1) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (D2-1-2-2-1) Authenticate - User1[Enabled=1]; Authenticate Response - Success (AuthStatus = 01h)	PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-12) ACE.Set vith different UIDs; ACE.Get is issued to verify; data - Matching (D1-12-12) ACE.Set vith different UIDs; ACE.Get is issued to verify data - Matching (D1-13-12) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-11) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (D2-12-2-3-1) Authenticate - User1[Enabled=1]; Authenticate Response - Fail	PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-1-2-1) ACE. Set is issued to verify; Data comparison - Matching (D1-1-2-12) ACE. Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-1-3-1) ACE. Set in a transaction with endTransaction status = 0; The new value retains the set (D1-1-3-1-2) ACE. Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-1-2-1-1) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (D2-1-2-2) Authenticate - User1[Enabled=1); Authenticate Response - Fail (D2-1-2-3-2) The previous successful authentication result with this authority in this session shall not	PASS PASS PASS PASS PASS PASS PASS
Code: 0Ch (Invalid_Param) (D1-12-1-1) ACE.Get is issued to verify; Data comparison - Matching (D1-12-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (D2-1-2-3-1) Authenticate - User1[Enabled=0]; Authenticate Response - Fail (D2-12-2-3) Authenticate - User1[Enabled=0]; Authenticate Response - Fail (D2-12-2-3) The previous successful authentication result with this authority in this session shall not be affected	PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-1-2-1) ACE. Set is issued to verify; Data comparison - Matching (D1-1-2-12) ACE. Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-1-3-1) ACE. Set in a transaction with endTransaction status = 0; The new value retains the set (D1-1-3-1-2) ACE. Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-1-2-1-1) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (D2-1-2-2) Authenticate - User1[Enabled=1); Authenticate Response - Fail (D2-1-2-3-2) The previous successful authentication result with this authority in this session shall not	PASS PASS PASS PASS PASS PASS N/A
Code: 0Ch (Invalid_Param) (D1-12-1-1) ACE.Get is issued to verify; Data comparison - Matching (D1-12-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Authenticate - User1{Enabled'column (05h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1{Enabled'=1}; Authenticate Response - Fail (D2-12-2-1) Authenticate - User1{Enabled'=0}; Authenticate Response - Fail (D2-12-2-3) Authenticate - User1{Enabled=1}; Sync Session - Pass (D2-12-2-3) Start Session - as User1[Enabled=1]; Sync Session - Pass (D2-12-2-3) Start Session - as User1[Enabled=0]; Sync Session - Fail (D2-13-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set	PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-12) ACE.Set et is issued to verify; Data comparison - Matching (D1-12-12) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-12) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-12) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-11) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (D2-12-2-11) Authenticate - User1(Enabled=1); Authenticate Response - Fail (D2-12-2-31) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-12-2-31) The previous successful authentication result with this authority in this session shall not be affected (D2-12-2-31) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-12-3-12) Authority.Set in a transaction and endTran' status = 0; The value changes back to the GD2-13-12) Authority.Set in a transaction and endTran' status = 1; The value changes back to the	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-1-1) ACE. Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1(Enabled=1); Authenticate Response - Fail (D2-12-3-2) The previous successful authentication result with this authority in this session shall not be affected (D2-1-2-3-1) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-1-2-3-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-1-2-3-1) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value	PASS PASS PASS PASS PASS PASS PASS PASS
Code: 0Ch (Invalid_Param) (D1-12-21) ACE.Get is issued to verify; Data comparison - Matching (D1-12-12) ACE. Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-11) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-12) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-21) Authenticate - User1[Enabled'column (05h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled'-0]; Authenticate Response - Fail (D2-12-2-1) Authenticate - User1[Enabled'-0]; Authenticate Response - Fail (D2-12-2-1) for previous successful authentication result with this authority in this session shall not be affected (D2-12-2-1) Start Session - as User1[Enabled=1]; Sync Session - Pass (D2-13-1) Authority.Set in a transaction and endTran' status = 0; The env value retains the set (D2-13-1.2) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set()	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-12) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-3-12) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-3-12) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-3) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1(Enabled=1); Authenticate Response - Fail (D2-12-2-3) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-12-2-3) The previous successful authentication result with this authority in this session shall not be affected (D2-12-2-3.1) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-12-3-1.2) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-12) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-12-12) Start Request: PIN = Null; Response: Pass	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-12) ACE. Set tis issued to verify; Data comparison - Matching (D1-12-12) ACE. Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-11) ACE. Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-12) ACE. Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-13) Get Request (User1) - 'Enabled' column (Ob(h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1(Enabled=1); Authenticate Response - Fall (D2-12-3-2) The previous successful authentication result with this authority in this session shall not be affected (D2-12-2-1) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-12-2-1) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-12-3-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set original value D2-12-3-1; Start Session - as User1(Enabled=1); Sync Session - Fail (D2-13-12-1) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-12-12; Set Request: PIN = Null; Response: Pass (D3-12-12; Set Request: PIN = Null; Response: Pass	PASS PASS PASS PASS PASS PASS PASS PASS
Code: 0Ch (Invalid_Param) (D1-12-1-1) ACE.Get is issued to verify; Data comparison - Matching (D1-12-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled'=1); Authenticate Response - Fail (D2-12-2-1) Authenticate - User1[Enabled'=0); Authenticate Response - Fail (D2-12-2-1) Authenticate - User1[Enabled=1); Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=1); Sync Session - Pass (D2-12-2-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1-1) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-12-1-2) Set Request: PIN = Null; Response: Pass (D3-12-1-2) Set Request: PIN = Null; Response: Pass (D3-12-12) S	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-12) ACE. Set tis issued to verify; Data comparison - Matching (D1-12-12) ACE. Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-11) ACE. Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-12) ACE. Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-13) Get Request (User1) - 'Enabled' column (Ob(h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1(Enabled=1); Authenticate Response - Fall (D2-12-3-2) The previous successful authentication result with this authority in this session shall not be affected (D2-12-2-1) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-12-2-1) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-12-3-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set original value D2-12-3-1; Start Session - as User1(Enabled=1); Sync Session - Fail (D2-13-12-1) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-12-12; Set Request: PIN = Null; Response: Pass (D3-12-12; Set Request: PIN = Null; Response: Pass	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-12) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-12) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-12) ACE.Set is a transaction with endTransaction status = 0; The new value retains the set (D1-13-12) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Get Request (User1) - 'Enabled' column (O5h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled=1]; Authenticate Response - Success (AuthStatus = 01h) (D2-12-2-3) Authenticate - User1[Enabled=0]; Authenticate Response - Fail (D2-12-2-3) The previous successful authentication result with this authority in this session shall not be affected (D2-12-2-3) Start Session - as User1[Enabled=1]; Sync Session - Pass (D2-1-2-3-1) Start Session - as User1[Enabled=1]; Sync Session - Pass (D2-13-12) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-12) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value (D3-12-12) Set Request: PIN with 32 byte; Response: Pass (D3-12-12) Set Request: PIN with 32 byte; Response: Pass (D3-13-12) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The Value share set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The Value share set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The Value share set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original Value Sace Sace Sace Sace Sace Sace Sace Sac	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-12) ACE. Set tis issued to verify; Data comparison - Matching (D1-12-12) ACE. Set with different UIDs; ACE. Get is issued to verify data - Matching (D1-13-12) ACE. Set tin a transaction with endTransaction status = 0; The new value retains the set (D1-13-12) ACE. Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-13) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1(Enabled=1); Authenticate Response - Fall (D2-12-3-2) The previous successful authentication result with this authority in this session shall not be affected (D2-12-2-1) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-12-2-1) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-13-2-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set original value D3: C_PIN.Set() (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN with 32 byte; Response: Pass (D3-12-12) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The VIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-12) ACE. Set vith different UIDs; ACE.Get is issued to verify data - Matching (D1-12-12) ACE. Set vith different UIDs; ACE.Get is issued to verify data - Matching (D1-13-12) ACE. Set vith different UIDs; ACE.Get is issued to verify data - Matching (D1-13-12) ACE. Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-12) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-11) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (D2-12-2-11) Authenticate - User1(Enabled=1); Authenticate Response - Fail (D2-12-2-12) Authenticate - User1(Enabled=1); Authenticate Response - Fail (D2-12-2-13) Authenticate - User1(Enabled=1); Sync Session - Pass (D2-12-2-13) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-12-2-13) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-12-2-13) Authority.Set in a transaction and endTran' status = 0; The new value retains the set original value D3: C_PIN.Set() (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-12-12) Set new PIN in a transaction with endTransaction status = 1; The VIN changes back to the original value D4: Locking.Set() for 'RangeStart' and 'RangeLength' (D4-12-12) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: 0Ch (Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-1-1) ACE.Get is issued to verify. Data comparison - Matching (D1-12-1-2) ACE. Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Authenticate - User1[Enabled'column (OSh); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled'-D]; Authenticate Response - Fail (D2-12-2-1) Authenticate - User1[Enabled'-D]; Authenticate Response - Fail (D2-12-2-1) Authenticate - User1[Enabled'-D]; Surt Session - Pass (D2-12-2-1) Start Session - as User1[Enabled'D]; Sync Session - Pass (D2-12-2-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1-2) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-13-12) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The VIN changes back to the original value D4: Locking.Set() for 'RangeStart' and 'RangeLength' (D4-12-12). RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: OCh (Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-12) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-12) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-12) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-12) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-12) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Get Request (User1) - 'Enabled' column (O5h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled=1]; Authenticate Response - Success (AuthStatus = 01h) (D2-12-2-1) Authenticate - User1[Enabled=0]; Authenticate Response - Fail (D2-12-2-1) Start Session - as User1[Enabled=1]; Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=1]; Sync Session - Pass (D2-13-12) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-12) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-12-12) Set Request: PIN with 32 byte; Response: Pass (D3-12-12) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The Value changes back to the original value D4-12-12) Set new PIN in a transaction with endTransaction status = 1; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The PIN retains the set (D3-13-12) RangeStart' and	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-12) ACE. Set with different UIDs; ACE. Get is issued to verify data - Matching (D1-12-12) ACE. Set with different UIDs; ACE. Get is issued to verify data - Matching (D1-13-12) ACE. Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-12) ACE. Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority. Set() testing (D2-12-2-11) Get Request (User1) - 'Enabled' column (O5h); Get Response - 1/0 in 'Enabled' column (D2-12-2-11) Authenticate - User1(Enabled=1); Authenticate Response - Fail (D2-12-2-3) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-12-2-3) The previous successful authentication result with this authority in this session shall not be affected (D2-12-2-1) Start Session - as User1(Enabled=0); Sync Session - Pass (D2-12-2-3) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-12-3-1) Authority. Set in a transaction and endTran' status = 0; The new value retains the set original value D3: C_PIN.Set() (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN with 32 byte; Response: Pass (D3-13-12) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-13-13) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The PIN retains the set (D3-13-13-13) Set new PIN in a transaction with endTransaction status = 1; The PIN retains the set (D3-13-13-13) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-13-1	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-1-1) ACE.Get is issued to verify, Data comparison - Matching (D1-12-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Get Request (User1) - 'Enabled' column (OSh); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled'=1); Authenticate Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled'=0); Authenticate Response - Fail (D2-12-2-1) Authenticate - User1[Enabled'=1); Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=1); Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=0); Sync Session - Fail (D2-13-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1-2) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-13-12) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The Value changes back to the original value D4: Locking.Set() for 'RangeStart' and 'RangeLength' (D4-13-12) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value D4: Locking.Set() for 'RangeStart' and 'RangeLength' (D4-13-11) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: OCh (Invalid_Param) (D4-13-11) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass (D4-13-11) RangeStart/Len Effect: With 'RangeStart' and 'RangeLength'; Get() retrieves the values indicated by Set() (D4-13-2-1) RangeStart/Len Effect: With 'RangeSt	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-1-1) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-2) ACE.Set vith different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Get Request (User1) - 'Enabled' column (O5h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1(Enabled=1); Authenticate Response - Success (AuthStatus = 01h) (D2-12-2-3) Authenticate - User1(Enabled=1); Authenticate Response - Fail (D2-12-2-1) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-12-2-1) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-13-2-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-12) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-12-12) Set Request: PIN with 32 byte; Response: Pass (D3-13-12) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The Value changes back to the original value D4: Locking.Set() for 'RangeStart' and 'RangeLength' (D4-12-12) RangeStart/Len Effect: Set with right Name-Value's values; Response - Status Code: OCh (Invalid_Param) (D4-13-11) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass (D4-13-11) RangeStart/Len Effect: With 'RangeStart'=changed and 'RangeLength'; Get() retrieves the values indicated by Set()	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-1-1) ACE.Get is issued to verify, Data comparison - Matching (D1-12-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Get Request (User1) - 'Enabled' column (OSh); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled'=1); Authenticate Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled'=0); Authenticate Response - Fail (D2-12-2-1) Authenticate - User1[Enabled'=1); Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=1); Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=0); Sync Session - Fail (D2-13-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1-2) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-13-12) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The Value changes back to the original value D4: Locking.Set() for 'RangeStart' and 'RangeLength' (D4-13-12) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value D4: Locking.Set() for 'RangeStart' and 'RangeLength' (D4-13-11) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: OCh (Invalid_Param) (D4-13-11) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass (D4-13-11) RangeStart/Len Effect: With 'RangeStart' and 'RangeLength'; Get() retrieves the values indicated by Set() (D4-13-2-1) RangeStart/Len Effect: With 'RangeSt	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-1-1) ACE.Get is issued to verify. Data comparison - Matching (D1-12-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Get Request (User1) - 'Enabled' column (OSh); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled'=1]; Authenticate Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled'=1]; Authenticate Response - Fail (D2-12-2-1) Authenticate - User1[Enabled'=1]; Sync Session - Fail (D2-12-2-1) Start Session - as User1[Enabled]; Sync Session - Fail (D2-12-2-1) Start Session - as User1[Enabled]; Sync Session - Fail (D2-13-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1-2) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1-2) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN with 32 byte; Response: Pass (D3-12-13-1) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-13-1) Set new PIN in a transaction with endTransaction status = 1; The Value changes back to the original value D4: Locking.Set() (Or 'AngeStart' and 'RangeLength' (D4-13-1-1) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: OCh (Invalid_Param) (D4-13-1-1) RangeStart/Len Effect: Get the values of 'RangeStart' and 'RangeLength', Set() retrieves the values indicated by Set() (D4-13-3-1) RangeStart/Len Effect: With 'RangeStart'=changed and 'RangeLength', Set() retrieves the values indicated by Set() (D4-13-3-1) RangeStart/Len Effect: With 'RangeStart'=changed and 'RangeLength'=0; Response with Get - the values as intended by Set()	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-1-1) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-2) ACE.Set vith a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-1-1) Get Request (User1) - 'Enabled' column (D5h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled=1); Authenticate Response - Success (AuthStatus = 01h) (D2-12-2-3.1) Authenticate - User1[Enabled=0); Authenticate Response - Fail (D2-12-2-3.1) Authenticate - User1[Enabled=1); Sync Session - Pass (D2-12-2-3.1) Start Session - as User1[Enabled=0); Sync Session - Pass (D2-12-2-1.1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1.2) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-12-1-2) Set Request: PIN = Nuli; Response: Pass (D3-12-12) Set Request: PIN = Nuli; Response: Pass (D3-12-12) Set Request: PIN = Nuli; Response: Pass (D3-12-12) Set Request: PIN = Nuli; Response: Pass (D3-13-12) Set Request: PIN = Nuli; Response: Pass (D3-13-12) Set Request: PIN = Nuli; Response: Pass (D3-12-12) Set Request: PIN = Nuli; Response: Pass (D3-13-12) Set Request: PIN = Nuli; Response: Pass (D4-13-12) RangeStart/Len Effect: Set with right Name-Value's values;	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-1-1) ACE.Get is issued to verify Data comparison - Matching (D1-12-1-2) ACE. Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Get Request (User1) - 'Enabled' column (D5h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled=1); Authenticate Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled=0); Authenticate Response - Fail (D2-12-2-1) Authenticate - User1[Enabled=0); Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=1); Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=0); Sync Session - Fail (D2-13-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1-1) Authority.Set in a transaction and endTran' status = 0; The PIN retains the set (D2-13-1-2) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-13-12) Set Re	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-1-1) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1(Enabled=1); Authenticate Response - Success (AuthStatus = 01h) (D2-12-2-1) Authenticate - User1(Enabled=1); Authenticate Response - Fail (D2-12-2-1) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-12-2-1) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-12-2-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-12) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-12) Authority.Set in a transaction and endTran' status = 0; The PIN retains the set (D3-12-12) Set Request: PIN with 32 byte; Response: Pass (D3-12-12) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The Value changes back to the original value D4: Locking,Set() for 'RangeStart' and 'RangeLength' (D4-12-12) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: OCh (Invalid_Param) (D4-13-11) RangeStart/Len Effect: Set with 'RangeStart'=changed and 'RangeLength' =0; Response with Get - no LBA covered by Set() (D4-13-3-1) RangeStart/Len Effect: with 'RangeStart'=changed and 'RangeLength' =0; Response with Get - no LBA covered by Set() (D4-13-3-1) RangeStart/Len Effect: with 'RangeLength'=0; Response with Get - no LBA covered by (D4-13-1-1) RangeStart/Len Effect: with 'RangeLength'=0; Response with Get - no LBA covered by (D4-13-1-	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-1-1) ACE.Get is issued to verify, Data comparison - Matching (D1-12-1-2) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Authenticate - User1[Enabled'column (OSh); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled'-D]; Authenticate Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled'-D]; Authenticate Response - Fail (D2-12-2-1) Authenticate - User1[Enabled'-D]; Authenticate Response - Fail (D2-12-2-1) Authenticate - User1[Enabled']; Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled]; Sync Session - Pass (D2-12-2-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1.2) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1.2) Start Session - as User1[Enabled]; Sync Session - Fail (D2-13-2.1) Start Session - as User1[Enabled]; Sync Session - Fail (D2-13-2.2) Ster Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN with 32 byte; Response: Pass (D3-13-1.1) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-1.2) Set Request: PIN with 32 byte; Response: Pass (D3-13-1.2) Set Request: PIN with 32 byte; Response: Pass (D3-13-1.2) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value D4: Locking.Set() for 'RangeStart' and 'RangeLength' (D4-13-3-1.1) RangeStart/Len Effect: Get the values of 'RangeStart' and 'RangeLength', Set() (D4-13-3-1.1) RangeStart/Len Effect: With 'RangeStart'=changed and 'RangeLength', Set() retrieves the values incided by Set() (D4-13-3-1.1) RangeStart/Len Effect: With 'RangeLength'=0; Response with Get - no LBA covered by that range (D4-13-3-1.1	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-1-1) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-1-2) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Authenticate - User1[Enabled'=1]; Authenticate Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled=0]; Authenticate Response - Fail (D2-12-2-1) Authenticate - User1[Enabled=0]; Authenticate Response - Fail (D2-12-2-1) Authenticate - User1[Enabled=0]; Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=0]; Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=0]; Sync Session - Pass (D2-12-2-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1-1) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-13-12) Set Request: PIN = Null; Response: Pass (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value D4: Locking.Set() for 'RangeStart' and 'RangeLength' (D4-12-12) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: OCh (Invalid_Param) (D4-13-11) RangeStart/Len Effect: Set with right Name-Value's values; Response - Status Code: OCh (Invalid_Param) (D4-13-3-1) RangeStart/Len Effect: With 'RangeStart'=changed and 'RangeLength']=0; Response with Get - no LBA covered by Net() (D4-13-3-1) RangeStart/Len Effect: with 'RangeStart'=changed and 'RangeLength']=0; Response with Get - no LBA covered by that range (D4-13-3-1) RangeStart/Len Effect: with 'RangeLength'=0; Response with Get - no LBA cove	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-1-1) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-2) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1(Enabled=1); Authenticate Response - Success (AuthStatus = 01h) (D2-12-2-3) The previous successful authentication result with this authority in this session shall not be affected (D2-12-2-1) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-13-2-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-12) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D3-13-12) Set Request: PIN with 32 byte; Response: Pass (D3-13-2-12) Set Request: PIN with 32 byte; Response: Pass (D3-13-2-12) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The VIN changes back to the original value D4: Locking,Set(1) for 'RangeStart' and 'RangeLength' (D4-13-2-12) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: OCh (Invalid_Param) (D4-13-2-13) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass (D4-13-13) RangeStart/Len Effect: with 'RangeStart'=changed and 'RangeLength' =0; Response with det - no LBA covered by Set(1) (D4-13-3-11) RangeStart/Len Effect: with 'RangeLength'=0; Response with Get - no LBA covered by (D4-1-3-3-13) RangeStart/Len Effect: with 'RangeLength'=0; Response with Get - no LBA covered by (D4-1-12) RangeStar	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-1-1) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-1-2) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Authenticate - User1[Enabled'=1]; Authenticate Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled=0]; Authenticate Response - Fail (D2-12-2-1) Authenticate - User1[Enabled=0]; Authenticate Response - Fail (D2-12-2-1) Authenticate - User1[Enabled=0]; Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=0]; Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=0]; Sync Session - Pass (D2-12-2-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1-1) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-13-12) Set Request: PIN = Null; Response: Pass (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value D4: Locking.Set() for 'RangeStart' and 'RangeLength' (D4-12-12) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: OCh (Invalid_Param) (D4-13-11) RangeStart/Len Effect: Set with right Name-Value's values; Response - Status Code: OCh (Invalid_Param) (D4-13-3-1) RangeStart/Len Effect: With 'RangeStart'=changed and 'RangeLength']=0; Response with Get - no LBA covered by Net() (D4-13-3-1) RangeStart/Len Effect: with 'RangeStart'=changed and 'RangeLength']=0; Response with Get - no LBA covered by that range (D4-13-3-1) RangeStart/Len Effect: with 'RangeLength'=0; Response with Get - no LBA cove	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-1-1) ACE.Get is issued to verify, Data comparison - Matching (D1-12-1-2) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Authenticate - User1[Enabled'=1]; Authenticate Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled'=1]; Authenticate Response - Fail (D2-12-2-1) Authenticate - User1[Enabled'=1]; Authenticate Response - Fail (D2-12-2-1) Authenticate - User1[Enabled'=1]; Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=1]; Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=1]; Sync Session - Pass (D2-12-2-1) Authenticate - User1[Enabled=1]; Sync Session - Fail (D2-13-2-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1-2) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-2-1) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN with 32 byte; Response: Pass (D3-13-13) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-13) Set new PIN in a transaction with endTransaction status = 1; The VIN changes back to the original value D4: Locking.Set() for 'RangeStart' and 'RangeLength' (D4-13-3-1) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: OCh (Invalid_Param) (D4-13-3-1) RangeStart/Len Effect: Get the values of 'RangeStart' and 'RangeLength' -G; Response with Get - the values as intended by Set() (D4-13-3-1) RangeStart/Len Effect: with 'RangeStart'=changed and 'RangeLength' -G; Response with Get - the values as intended by Set() (D4-13-3-1) RangeStart/Len Effect: with 'RangeLength'=O; Response with Get - the values as intended by Set() (D4-13-3-1) RangeStart/Len Effect: With 'RangeLength	PASS PASS PASS PASS PASS PASS PASS PASS
Code: 0Ch (Invalid_Param) (D1-12-1-1) ACE.Get is issued to verify, Data comparison - Matching (D1-12-1-2) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Authenticate - User1[Enabled'=1]; Authenticate Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled'=1]; Authenticate Response - Fail (D2-12-2-1) Authenticate - User1[Enabled'=1]; Sync Session - Pass (D2-12-2-1) Authenticate - User1[Enabled=1]; Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=1]; Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=1]; Sync Session - Fail (D2-13-2-1) Start Session - as User1[Enabled=0]; Sync Session - Fail (D2-13-2-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1-1) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-13-13) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-13) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value D4: Locking.Set() for 'RangeStart' and 'RangeLength' (D4-12-12) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: OCh (Invalid_Param) (D4-13-11) RangeStart/Len Effect: Get the values' or 'RangeStart' and 'RangeLength'=0; Response with Get - the values as intended by Set() (D4-13-3-1) RangeStart/Len Effect: with 'RangeStart'=changed and 'RangeLength'=0; Response with Get - no LBA covered by that range (D4-13-3-1) RangeStart/Len Effect: with 'RangeLength'=0; Response with Get - no LBA covered by (D4-13-3-1) RangeStart/Len Effect: with 'RangeLength'=0; Response with Get - no LBA covered by (D4-13-3-	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-12) ACE. Set with different UIDs; ACE. Get is issued to verify data - Matching (D1-12-12) ACE. Set with different UIDs; ACE. Get is issued to verify data - Matching (D1-13-1-1) ACE. Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-2) ACE. Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority. Set() testing (D2-12-2-1) Qut Request (User1) - 'Enabled' column (D5h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled=1); Authenticate Response - Success (AuthStatus = 01h) (D2-12-2-1) Authenticate - User1[Enabled=0); Authenticate Response - Fail (D2-12-2-1) Start Session - as User1[Enabled=1); Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=0); Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=0); Sync Session - Fail (D2-13-1-1) Authority. Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1-1) Authority. Set in a transaction and endTran' status = 0; The PIN retains the set (D3-12-1-2) Set Request: PIN = Null; Response: Pass (D3-12-1-2) Set Request: PIN = Null; Response: Pass (D3-12-1-2) Set Request: PIN = Null; Response: Pass (D3-12-1-2) Set Request: PIN = Null; Response: Pass (D3-13-1-1) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-1-2) Set Request: PIN = Null; Response: Pass (D3-13-1-1) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value D4: Locking. Sett] for 'RangeStart' and 'RangeLength' (D4-13-2-12) RangeStart/Lene Effect: Set with right Name-Value's values; Response - Status Code: OCh (Invalid_Param) (D4-13-1-1) RangeStart/Lene Effect: with 'RangeStart'=changed and 'RangeLength']=0; Response with Get - no LBA covered by Yet() (D4-13-3-1) RangeStart/Lene Effect: with 'RangeStart'=changed and 'RangeLength']=0; Response with Get - no LBA covered by Yet() (D4-13-3	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-1-2) ACE.Set vith different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-2) ACE.Set vith a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-2-1) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1(Enabled=1); Authenticate Response - Success (AuthStatus = 01h) (D2-12-2-1) Authenticate - User1(Enabled=1); Authenticate Response - Fail (D2-12-2-1) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-12-2-1) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-12-2-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-12) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-12) Authority.Set in a transaction and endTran' status = 0; The PIN retains the set (D3-12-12) Set Request: PIN with 32 byte; Response: Pass (D3-12-12) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The Value changes back to the original value D4: Locking.Set() for 'RangeStart' and 'RangeLength' (D4-13-21) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: OCh (Invalid_Param) (D4-13-21) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass (D4-13-3-11) RangeStart/Len Effect: with 'RangeStart'=changed and 'RangeLength' =0; Response with det - the values as intended by Set() (D4-13-3-11) RangeStart/Len Effect: with 'RangeLength' =0; Response with Get - no LBA covered by Mat range (D4-1-3-4-12) RangeStart/Len Effect: with 'RangeLength']=0; Response with Get - no LBA covered by (D4-1-3-4-12) RangeStart/Len Effect: with	PASS PASS PASS PASS PASS PASS PASS PASS
Code: 0Ch (Invalid_Param) (D1-12-1-1) ACE.Get is issued to verify; Data comparison - Matching (D1-12-1-2) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-1-1) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1{Enabled'=1}; Authenticate Response - Fail (D2-12-2-1) Authenticate - User1{Enabled'=0}; Authenticate Response - Fail (D2-12-2-1) Authenticate - User1{Enabled'=1}; Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=1); Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=1); Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=0); Sync Session - Fail (D2-13-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1-2) Start Session - as User1[Enabled=0]; Sync Session - Fail (D2-13-1-2) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN with 32 byte; Response: Pass (D3-13-13) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-12) Set Request: PIN with 32 byte; Response: Pass (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value D4: Locking.Set() for 'RangeStart' and 'RangeLength' (D4-13-3-1) RangeStart/Len Effect: Get the values' or 'RangeStart' and 'RangeLength': Get() retrieves the values indicated by Set() (D4-13-3-1) RangeStart/Len Effect: With 'RangeStart'=changed and 'RangeLength'=0; Response with Get - the values as intended by Set() (D4-13-3-1) RangeStart/Len Effect: With 'RangeLength'=0; Response with Get - ne LBA covered by (D4-13-3-1) RangeStart/Len Effect: With 'RangeLength'=0; Response with Get - ne LBA covered by (D4-1-	PASS PASS PASS PASS PASS PASS PASS PASS
Code: 0Ch (Invalid_Param) (D1-12-12) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-12) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-2) ACE.Set vith a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-1-1) Get Request (User1) - 'Enabled' column (D5h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled=1); Authenticate Response - Success (AuthStatus = 01h) (D2-12-2-1) Authenticate - User1[Enabled=0); Authenticate Response - Fail (D2-12-2-1) Start Session - as User1[Enabled=1); Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=0); Sync Session - Fail (D2-13-2-1) Start Session - as User1[Enabled=0); Sync Session - Fail (D2-13-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1-2) Set Request: PIN = Null; Response: Pass (D3-12-1-2) Set Request: PIN = Null; Response: Pass (D3-12-1-2) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-13-1-1) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-1-2) Set Request: PIN = Null; Response: Pass (D3-13-1-1) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value D4: Locking.Sett] for 'RangeStart' and 'RangeLength' (D4-13-2-12) RangeStart/Lene Effect: Set with right Name-Value's values; Response - Status Code: OCh (Invalid_Param) (D4-13-3-1) RangeStart/Lene Effect: with 'RangeStart'=changed and 'RangeLength']=0; Response with Get - no LBA covered by Yet() (D4-13-3-1) RangeStart/Lene Effect: with 'RangeStart'=changed and 'RangeLength']=0; Response with Get - no LBA covered by Yet() (D4-13-3-1) RangeStart/Lene Effect: with 'RangeLength']=0;	PASS PASS PASS PASS PASS PASS PASS PASS
Code: OCh (Invalid_Param) (D1-12-1-1) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-1-2) ACE.Set vith different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-2) ACE.Set vith different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-2) ACE.Set vith different UIDs; ACE.Get is issued to verify data - Matching (D2-12-2-1) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1(Enabled=1); Authenticate Response - Success (AuthStatus = 01h) (D2-12-2-1) Authenticate - User1(Enabled=1); Authenticate Response - Fail (D2-12-2-1) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-12-2-1) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-13-12) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-12) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D3-12-12) Set Request: PIN with 32 byte; Response: Pass (D3-12-12) Set Request: PIN with 32 byte; Response: Pass (D3-13-12) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-12) Set new PIN in a transaction with endTransaction status = 1; The Value changes back to the original value D4: Locking,Set(1) for 'RangeStart' and 'RangeLength' (D4-13-2-12) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: OCh (Invalid_Param) (D4-13-2-13) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass (D4-13-3-11) RangeStart/Len Effect: with 'RangeStart'=changed and 'RangeLength' =0; Response with Get - the values as intended by Set(1) (D4-13-3-11) RangeStart/Len Effect: with 'RangeLength' =0; Response with Get - the values as intended by Set(1) (D4-13-3-11) RangeStart/Len Effect: with 'RangeLength' =0; Response with Get - the values as intended by Set(1) (D4-13-3-11) RangeStart/L	PASS PASS PASS PASS PASS PASS PASS PASS
Code: 0Ch (Invalid_Param) (D1-12-12) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-12-12) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set (D1-13-1-2) ACE.Set vith a transaction with endTransaction status = 1; The value changes back to the original value D2: Authority.Set() testing (D2-12-1-1) Get Request (User1) - 'Enabled' column (D5h); Get Response - 1/0 in 'Enabled' column (D2-12-2-1) Authenticate - User1[Enabled=1); Authenticate Response - Success (AuthStatus = 01h) (D2-12-2-1) Authenticate - User1[Enabled=0); Authenticate Response - Fail (D2-12-2-1) Start Session - as User1[Enabled=1); Sync Session - Pass (D2-12-2-1) Start Session - as User1[Enabled=0); Sync Session - Fail (D2-13-2-1) Start Session - as User1[Enabled=0); Sync Session - Fail (D2-13-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set (D2-13-1-2) Set Request: PIN = Null; Response: Pass (D3-12-1-2) Set Request: PIN = Null; Response: Pass (D3-12-1-2) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-13-1-1) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set (D3-13-1-2) Set Request: PIN = Null; Response: Pass (D3-13-1-1) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value D4: Locking.Sett] for 'RangeStart' and 'RangeLength' (D4-13-2-12) RangeStart/Lene Effect: Set with right Name-Value's values; Response - Status Code: OCh (Invalid_Param) (D4-13-3-1) RangeStart/Lene Effect: with 'RangeStart'=changed and 'RangeLength']=0; Response with Get - no LBA covered by Yet() (D4-13-3-1) RangeStart/Lene Effect: with 'RangeStart'=changed and 'RangeLength']=0; Response with Get - no LBA covered by Yet() (D4-13-3-1) RangeStart/Lene Effect: with 'RangeLength']=0;	PASS PASS PASS PASS PASS PASS PASS PASS

(D4-2-2-5) 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 (D4-2-2-3-3) RdLockEnabled/Locked=1/0: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and	N/A
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; (D4-2-2-4-1(2)) RdLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Read with multiple ranges	N/A
(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	N/A
(range2); Response - abortion/pass(if rangeCrossing=1/0) (D4-2-2-4-2) RdLockEnabled/Locked=0/0 w/ active MBR shadowing: Read with LBA covered by this	N/A
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) (D4-2-2-4-1(2)) RdLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Read with multiple ranges	N/A
(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 (D4-2-2-4-3) RdLockEnabled/Locked=0/1: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and	N/A
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 (D4-2-3-1-2) ReadLock Effect in Trans: Set ReadLockEnabled in a transaction and endTran's status=1;	PASS
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 (D4-3-2-1-1) WrLockEnabled/Locked: Get the contents of 'WriteLockEnabled' and 'WriteLocked';	PASS
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 (D4-3-2-2-1(2)) WrLockEnabled/Locked=1/1 w/ inactive MBR shadowing: Write with other range;	N/A
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 (D4-3-2-2-3) WrLockEnabled/Locked=1/1: Locked bit = 0 in Level 0 Discovery	N/A 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 (D4-3-2-3-1(2)) WrLockEnabled/Locked=1/0 w/ inactive MBR shadowing: Write with multiple ranges	N/A
(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
	N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (D4-3-2-3-3) WrlockEnabled/Locked=1/0: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and	N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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 N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (D4-3-2-3-3) WrlockEnabled/Locked=1/0: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and	N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-3-2) WrtockEnabled/Locked=1/0 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-3-3) WrtockEnabled/Locked=1/0: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges) (D4-3-2-3-5) WrtockEnabled/Locked=1/0: Power-on reset; Response - 'WriteLocked' = 1 (D4-3-2-4-1) WrtockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1(2)) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges	N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-3-5) WrLockEnabled/Locked=1/0: Power-on reset; Response - 'WriteLocked' = 1 (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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 N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-3-5) WrLockEnabled/Locked=1/0: Power-on reset; Response - 'WriteLocked' = 1 (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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) (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 N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-3-2) WtlockEnabled/Locked=1/0 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-3-3) WtlockEnabled/Locked=1/0: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges) (D4-3-2-3-5) WrtlockEnabled/Locked=1/0: Power-on reset; Response - 'WriteLocked' = 1 (D4-3-2-4-1) WtlockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1(2)) WtlockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-1(2)) WtlockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WritockEnabled/Locked=0/0 w/ active MBR shadowing: Write with BA covered by this (D4-3-2-4) WritockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with BA covered by this	N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4) WrLockEnabled/Locked=1/0: Power-on reset; Response - 'WriteLocked' = 1 (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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) (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) (D4-3-2-4-2) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with BLB covered by this range and not by MBR; Response - Pass	N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-5) WrLockEnabled/Locked=1/0: Power-on reset; Response - 'WriteLocked' = 1 (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1(2)) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (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) (D4-3-2-4-2) WrLockEnabled/Locked=0/0 w/ active MBR shadowing: Write with BA covered by this range and not by MBR; Response - Pass (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 N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=1/0: Power-on reset; Response - 'WriteLocked' = 1 (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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) (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) (D4-3-2-4-3) WrLockEnabled/Locked=0/0 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (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 N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=1/0: Power-on reset; Response - 'WriteLocked' = 1 (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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) (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) (D4-3-2-4-12) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with BA covered by this range and not by MBR; Response - Pass (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/1: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges) (D4-3-2-4-1) WrLockEnabled/Locked=0/1: Locked BR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1: w/ inactive MBR shadowing: Write with this range;	N/A N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=1/0: Power-on reset; Response - 'WriteLocked' = 1 (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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) (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) (D4-3-2-4-3) WrLockEnabled/Locked=0/0 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (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 N/A N/A N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=1/0: Power-on reset; Response - 'WriteLocked' = 1 (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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) (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) (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 (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with this range; (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1(2)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (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 N/A N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=1/0: Power-on reset; Response - 'WriteLocked' = 1 (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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) (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) (D4-3-2-4-3) WrLockEnabled/Locked=0/0 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (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 N/A N/A N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=1/0: Power-on reset; Response - 'WriteLocked' = 1 (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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) (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) (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 (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response	N/A N/A N/A N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-3-2) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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) (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) (D4-3-2-4-2) WrLockEnabled/Locked=0/0 w/ iactive MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-1) WrLockEnabled/Locked=0/0 Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges) (D4-3-2-4-1(2)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1(2)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (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) (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with BA covered by this range and not by MBR; Response - Pass (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with LBA covered by this range	N/A N/A N/A N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=1/0: Power-on reset; Response - 'WriteLocked' = 1 (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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) (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) (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 (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response	N/A N/A N/A N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1(2)) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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) (D4-3-2-4-2) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-12) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1(2)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1(2)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with BA covered by this range and not by MBR; Response - Pass (D4-3-3-1-1) WriteLock Effect in Trans. Set WriteLockEnabled in a transaction and endTran's status=0; The value reta	N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=1/0: Power-on reset; Response - 'WriteLocked' = 1 (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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) (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) (D4-3-2-4-3) WrLockEnabled/Locked=0/0 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with BA covered by this range and not by MBR; Response - Pass (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write w	N/A N/A N/A N/A N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1(2)) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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) (D4-3-2-4-2) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-12) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1(2)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1(2)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with BA covered by this range and not by MBR; Response - Pass (D4-3-3-1-1) WriteLock Effect in Trans. Set WriteLockEnabled in a transaction and endTran's status=0; The value reta	N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=1/0: Power-on reset; Response - 'WriteLocked' = 1 (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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) (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) (D4-3-2-4-3) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-1)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-1)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-1)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with BA covered by this range and not by MBR; Response - Pass (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-3-1)) WrtLockEnabled/Locked=0/1 w/	N/A N/A N/A N/A N/A N/A N/A N/A PASS N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12)) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-1)) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/0 w/ clucked Bk shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-2) WrLockEnabled/Locked=0/0 w/ clucked Bk shadowing: Write with this range; (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-12) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (range1); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with multiple ranges (range1); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (D4-3-3-1-1) WrLockEnabled/Locked=0/1: Locked bit = 0/1 in Level 0 Discovery (unlocked-write	N/A N/A N/A N/A N/A N/A N/A N/A N/A PASS N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=1/0: Power-on reset; Response - 'WriteLocked' = 1 (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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) (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) (D4-3-2-4-3) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-1)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-1)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-1)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with BA covered by this range and not by MBR; Response - Pass (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-3-1)) WrtLockEnabled/Locked=0/1 w/	N/A N/A N/A N/A N/A N/A N/A N/A PASS N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-12) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/0 w/ clicked Bs shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-2) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with multiple ranges (range3-1-1) WriteLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with multiple ranges (range3-1-1) WriteLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with multiple ranges (range3-1-1) WriteLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with multiple ranges (rang-	N/A N/A N/A N/A N/A N/A N/A N/A N/A PASS N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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) (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) (D4-3-2-4-2) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with EBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with BA covered by this range and not by MBR; Response - Pass (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 Si MBRControl.Set() Grammar and Effect (D5-1-2-2-1) Set Enable/Done = True (OII); Response - Pass (D5-1-2-2-1) Set Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2	N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-12) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/0 w/ clicked Bs shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-2) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with multiple ranges (range3-1-1) WriteLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with multiple ranges (range3-1-1) WriteLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with multiple ranges (range3-1-1) WriteLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with multiple ranges (rang-	N/A N/A N/A N/A N/A N/A N/A N/A N/A PASS N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Past(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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) (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) (D4-3-2-4-2) WrLockEnabled/Locked=0/0 w/ Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges) (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (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) (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) (D4-3-2-4-1)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with BA covered by this range and not by MBR; Response - Pass (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 D5: MBRControl.Set() Grammar and Effect (D5-1-2-2-1) Set Table/Done=1: Write tommand: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeC	N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D5-1-2-2-3) Enable/Done = True (OIh); Respon	N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Past(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/0: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges) (D4-3-2-4-12) WrLockEnabled/Locked=0/1: w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/1: w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (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) (D4-3-2-4-1)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with BA covered by this range and not by MBR; Response - Pass (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 D5: MBRControl.Set() Grammar and Effect (D5-1-2-2) Enable/Done=1: Write with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2) Enable/Done=1: Write with m	N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-12)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-12)) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-2) Fable/Done= True (OIh); Response - Pass (D5-1-2-2-1) Set Enable/Done = True (OIh); Response - Pass (D5-1-2-2-2) Fable/Done=1: Read command: pass (Read/WriteLockEnabled = 0)	N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Past(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-2) WrLockEnabled/Locked=0/0: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges) (D4-3-2-4-12) WrLockEnabled/Locked=0/1: w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/1: w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-12) WrLockEnabled/Locked=0/1: w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (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) (D4-3-2-4-1)) WrLockEnabled/Locked=0/1: w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-3) WrLockEnabled/Locked=0/1: w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (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 D5: MBRControl.Set() Grammar and Effect (D5-1-2-2) Enable/Done=1: Write with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1	N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A
(globalrange); Response - Abortion/Pass(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abortion/Pass(if rangeCrossing=1/0) (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) (D4-3-2-4-1)) WrLockEnabled/Locked=0/0 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-1)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with this range; (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) (D4-3-2-4-1)) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with multiple ranges (globalRange); Response - Abortion/Pass(if rangeCrossing=1/0) (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with LBA covered by this range and not by MBR; Response - Pass (D4-3-2-4-3) WrLockEnabled/Locked=0/1 w/ active MBR shadowing: Write with BA covered by this range and not by MBR; Response - Pass (D4-3-2-4-3) WrLockE	N/A N/A
(globalrange); Response - Abortion/Past(if rangeCrossing=1/0) (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 (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) (D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1(2)) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; (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) (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) (D4-3-2-4-2) WrLockEnabled/Locked=0/0: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges) (D4-3-2-4-1) WrLockEnabled/Locked=0/1: w/ inactive MBR shadowing: Write with this range; (D4-3-2-4-1(2)) WrLockEnabled/Locked=0/1: w/ inactive MBR shadowing: Write with this range; (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) (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) (D4-3-2-4-3) WrLockEnabled/Locked=0/1: w/ inactive MBR shadowing: Write with BA covered by this range and not by MBR; Response - Pass (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 D5: MBRControl.Set() Grammar and Effect (D5-1-2-2) Enable/Done=1: Write with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1	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 (D5-1-2-3-3(2)) Enable/Done=1/0: Write addressing LBA covered by MBR table and not by MBR;	N/A
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 (D5-1-2-4-1) Enable/Done=0/0: Read command: pass (Read/WriteLockEnabled = 0)	N/A 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) (D5-1-2-4-1) Enable/Done=0/0: Read command: fail (Read/WriteLockEnabled = 1)	N/A 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 (D5-1-2-4-1) Enable/Done=0/1: Read command: pass (Read/WriteLockEnabled = 0)	N/A 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	N/A
(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) (D5-1-2-4-2) Enable/Done=0/1: Write command: fail (Read/WriteLockEnabled = 1)	N/A 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 (D5-1-3-1-1) Set 'Enabled' = 1 in a transaction and endTransaction status = 0; The value retains the	N/A N/A
(D5-1-3-1-2) Set 'Enabled' = 0 in a transaction and endTransaction status = 0; The value retains the	N/A
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(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 D7: DataStore.Set() -Basic Grammar and Effect	N/A
(D7-1-1-1) Set Datastore; Response - Pass	PASS
(D7-1-1-1) 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 (D7-1-2-1-2) Datastore.Set in a transaction with endTransaction status = 1; The data changes back to	PASS
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 (D8-1-3-1-2) GenKey Effect in a transaction with endTransaction status = 1; The range's media	N/A
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
(D9-1-2-1-2) Activate to Eddings? If ATA security is chabled, response 3 status code. 3rth (rail) (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 PASS
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass	PASS PASS PASS
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Read data in sector 1; Compare the data - matching (D9-1-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the	PASS PASS PASS PASS PASS
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Read data in sector 1; Compare the data - matching (D9-1-3-2-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change	PASS PASS PASS PASS
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Read data in sector 1; Compare the data - matching (D9-1-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the	PASS PASS PASS PASS PASS
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Read data in sector 1; Compare the data - matching (D9-1-3-2-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change D10: AdminSP.Revert() Effect check (D10-1-1-1) AdminSP.Revert Grammar: Revert Session to AdminSP; Revert response - pass (D10-1-2-1-1) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be	PASS PASS PASS PASS PASS PASS PASS
(09-13-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (09-1-3-1-5) LockignSP.Activate() Effect: Read data in sector 1; Compare the data - matching (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg:inative - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg:inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect theck (D10-1-1-1-1) AdminSP.Revert Grammar: Revert Session to AdminSP; Revert response - pass (D10-1-2-1-2) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be (D10-1-2-1-2) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be	PASS PASS PASS PASS PASS PASS PASS N/A
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Read data in sector 1; Compare the data - matching (D9-1-3-2-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change D10: AdminSP.Revert() Effect check (D10-1-1-1) AdminSP.Revert Grammar: Revert Session to AdminSP; Revert response - pass (D10-1-2-1-1) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be	PASS PASS PASS PASS PASS PASS PASS
(09-13-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Read data in sector 1; Compare the data - matching (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg:inative - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg:inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect theck (D10-1-1-1-1) AdminSP.Revert Grammar: Revert Session to AdminSP; Revert response - pass (D10-1-2-1-2) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be (D10-1-2-1-2) AdminSP.Revert Effect: LockingEnabled bit = 0 from Level0_Discovery (D10-1-2-1-4) AdminSP.Revert Effect: The state of LockingSP in OFS(Manufactured/Manufactured- Inactivate)	PASS PASS PASS PASS PASS PASS PASS N/A
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Bread data in sector 1; Compare the data - matching (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect theck (D10-1-1-1-1) AdminSP.Revert Effect: The session within the AdminSP.Revert response - pass (D10-1-2-1-1) AdminSP.Revert Effect: Tha devices: check bit of word 82; bit 10 dword 85; word (D10-1-2-1-3) AdminSP.Revert Effect: Tha devices: check bit of word 82; bit 10 dword 85; word (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured- Inactivate) (D10-1-2-1-5) AdminSP.Revert Effect: The state on LockingSP; SyncSession - Status Code: I= 0 or	PASS PASS PASS PASS PASS PASS PASS N/A PASS PASS
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSesion on LockingSP with SID's PIN; SyncSesion - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Read data in sector 1; Compare the data - matching (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg:inative - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg:inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect theck (D10-1-1-1-1) AdminSP.Revert Grammar: Revert Sesion to AdminSP; Revert response - pass (D10-1-2-1-2) AdminSP.Revert Effect: The sesion within the AdminSP: Revert() was issued shall be (D10-1-2-1-2) AdminSP.Revert Effect: LockingEnabled bit = 0 from Level(_Discovery (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured- inattivate) (D10-1-2-1-5) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: I= 0 or no data returned (D10-1-2-1-6) AdminSP.Revert Effect: StartSession on AdminSP with MSID's PIN; SyncSession - pass	PASS PASS PASS PASS PASS PASS PASS N/A PASS
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Bread data in sector 1; Compare the data - matching (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect theck (D10-1-1-1-1) AdminSP.Revert Effect: The session within the AdminSP.Revert response - pass (D10-1-2-1-1) AdminSP.Revert Effect: Tha devices: check bit of word 82; bit 10 dword 85; word (D10-1-2-1-3) AdminSP.Revert Effect: Tha devices: check bit of word 82; bit 10 dword 85; word (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured- Inactivate) (D10-1-2-1-5) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: I= 0 or no data returned (D10-1-2-1-6) AdminSP.Revert Effect: StartSession on AdminSP with MSID's PIN; SyncSession - pass (D10-1-2-1-3) AdminSP.Revert Effect: StartSession on AdminSP with MSID's PIN; SyncSession - pass	PASS PASS PASS PASS PASS PASS PASS PASS
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSesion on LockingSP with SID's PIN; SyncSesion - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Read data in sector 1; Compare the data - matching (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg:inative - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg:inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect theck (D10-1-1-1-1) AdminSP.Revert Grammar: Revert Sesion to AdminSP; Revert response - pass (D10-1-2-1-2) AdminSP.Revert Effect: The sesion within the AdminSP: Revert() was issued shall be (D10-1-2-1-2) AdminSP.Revert Effect: LockingEnabled bit = 0 from Level(_Discovery (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured- inattivate) (D10-1-2-1-5) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: I= 0 or no data returned (D10-1-2-1-6) AdminSP.Revert Effect: StartSession on AdminSP with MSID's PIN; SyncSession - pass	PASS PASS PASS PASS PASS PASS PASS N/A PASS PASS PASS
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Bread data in sector 1; Compare the data - matching (D9-1-3-2-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect theck (D10-1-1-1-1) AdminSP.Revert Effect: The session within the AdminSP.Revert response - pass (D10-1-2-1-1) AdminSP.Revert Effect: Tha devices: check bit of word 82; bit 10 dword 85; word (D10-1-2-1-3) AdminSP.Revert Effect: Tha devices: check bit of word 82; bit 10 dword 85; word (D10-1-2-1-3) AdminSP.Revert Effect: Tha devices: check bit of word 82; bit 10 dword 85; word (D10-1-2-1-4) AdminSP.Revert Effect: Tha devices: check bit of word 82; bit 10 dword 85; word (D10-1-2-1-3) AdminSP.Revert Effect: Tha devices: check bit of word 82; bit 10 dword 85; word (D10-1-2-1-3) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: I= 0 or no data returned (D10-1-2-3-1) AdminSP.Revert Effect: StartSession on AdminSP with MSID's PIN; SyncSession - pass (D10-1-2-3) AdminSP.Revert Effect: LockingSP in inactive: Read data in sector 1; Compare the data - matching (D10-1-2-2) AdminSP.Revert Effect: LockingSP in active: Read data in sector 1; Compare the data - matching	PASS PASS PASS PASS PASS PASS PASS N/A PASS PASS PASS PASS PASS NM PASS
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Bread data in sector 1; Compare the data - matching (D9-1-3-2-1) LockignSP.Activate() Effect: LockingSP in mfg:native - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg:native - PIN for Admin1 does not change D10: AdminSP.Revert() Effect: LockingSP in mfg:state - PIN for Admin1 does not change D10: AdminSP.Revert() Effect: The session vithin the AdminSP.Revert (Psponse - pass (D10-1-2-1-1) AdminSP.Revert Effect: Tor ATA devices: check bit1 of word 82; bit1 of word 85; word (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured- Inactivate) (D10-1-2-1-5) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: I= 0 or no data returned (D10-1-2-1-5) AdminSP.Revert Effect: StartSession on AdminSP with MSID's PIN; SyncSession - pass (D10-1-2-1-3) AdminSP.Revert Effect: LockingSP in anctive: Read data in sector 1; Compare the data - matching (D10-1-2-1-3) AdminSP.Revert Effect: LockingSP in active: Read data in sector 1; Compare the data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Read data in sector 1; Compare the data - matching (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Read data in sector 1; Compare the data - matching	PASS PASS PASS PASS PASS PASS PASS N/A PASS PASS PASS PASS PASS PASS NM
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Bread data in sector 1; Compare the data - matching (D9-1-3-2-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect theck (D10-1-1-1-1) AdminSP.Revert Effect: The session within the AdminSP.Revert response - pass (D10-1-2-1-1) AdminSP.Revert Effect: Tha devices: check bit of word 82; bit 10 dword 85; word (D10-1-2-1-3) AdminSP.Revert Effect: Tha devices: check bit of word 82; bit 10 dword 85; word (D10-1-2-1-3) AdminSP.Revert Effect: Tha devices: check bit of word 82; bit 10 dword 85; word (D10-1-2-1-4) AdminSP.Revert Effect: Tha devices: check bit of word 82; bit 10 dword 85; word (D10-1-2-1-3) AdminSP.Revert Effect: Tha devices: check bit of word 82; bit 10 dword 85; word (D10-1-2-1-3) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: I= 0 or no data returned (D10-1-2-3-1) AdminSP.Revert Effect: StartSession on AdminSP with MSID's PIN; SyncSession - pass (D10-1-2-3) AdminSP.Revert Effect: LockingSP in inactive: Read data in sector 1; Compare the data - matching (D10-1-2-2) AdminSP.Revert Effect: LockingSP in active: Read data in sector 1; Compare the data - matching	PASS PASS PASS PASS PASS PASS PASS N/A PASS PASS PASS PASS PASS NM PASS
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Bread data in sector 1; Compare the data - matching (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect: LockingSP in mfg-state - PIN for Admin1 does not change D10: AdminSP.Revert() Effect: The cession vithin the AdminSP. Revert response - pass (D10-1-2-1-1) AdminSP.Revert Effect: The session vithin the AdminSP.Revert() was issued shall be (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured- Inactivate) (D10-1-2-1-5) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: I= 0 or no data returned (D10-1-2-1-5) AdminSP.Revert Effect: StartSession on AdminSP; NevSpncSession - status Code: I= 0 or no data returned (D10-1-2-1-5) AdminSP.Revert Effect: LockingSP in inactive: Read data in sector 1; Compare the data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Read data in sector 1; Compare the data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Read data in sector 1; Compare the data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Data in DatStore table shall be the value (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in D10: LockingSP.Revert() Effect check (D10-2-1-1) LockingSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in D10: LockingSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in D10-2-1-1) LockingSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in	PASS PASS PASS PASS PASS PASS PASS PASS
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Bread data in sector 1; Compare the data - matching (D9-1-3-2-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect theck (D10-1-1-1-1) AdminSP.Revert Effect: The session within the AdminSP; Revert response - pass (D10-1-2-1-1) AdminSP.Revert Effect: Tha Calvics: Lock bit of word 82; bit 10 word 85; word (D10-1-2-1-3) AdminSP.Revert Effect: Tha Calvics: Lock bit of word 82; bit 10 word 85; word (D10-1-2-1-3) AdminSP.Revert Effect: Tha Calvics: LockingSP is in OFS(Manufactured/Manufactured/ Inactivate) (D10-1-2-1-3) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: I= 0 or no data returned (D10-1-2-3-1) AdminSP.Revert Effect: StartSession on AdminSP with MSID's PIN; SyncSession - pass (D10-1-2-3-1) AdminSP.Revert Effect: LockingSP in inactive: Read data in sector 1; Compare the data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Read data in sector 1; Compare the data - matching (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in D10: LockingSP.Revert Effect: CockingSP in active: Data in MBR table shall be the value in D10: LockingSP.Revert Effect: CockingSP in active: Data in MBR table shall be the value in D10: LockingSP.Revert Effect: CockingSP in active: Data in MBR table shall be the value in D10: LockingSP.Revert Effect: CockingSP in active: Data in MBR table shall be the value in D10: LockingSP.Revert Effect: CockingSP in active: Data in MBR table shall be the value in D10: LockingSP.Revert Effect: CockingSP in acti	PASS PASS PASS PASS PASS PASS PASS PASS
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Bread data in sector 1; Compare the data - matching (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect: The session within the AdminSP. Revert response - pass (D10-1-2-1-1) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured- Inactivate) (D10-1-2-1-5) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: I= 0 or no data returned (D10-1-2-1-5) AdminSP.Revert Effect: LockingSP in anctive: Read data in sector 1; Compare the data - matching (D10-1-2-1-3) AdminSP.Revert Effect: LockingSP in anctive: Read data in sector 1; Compare the data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Data in DBtStore table shall be the value (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in D10: LockingSP.Revert() Effect: LockingSP in active: Data in MBR table shall be the value (D10-1-2-2-1) LockingSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in D10: LockingSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in D10: LockingSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in D10-2-2-1-1) LockingSP.Revert Effect: LockingSP in active: Data in	PASS PASS PASS PASS PASS PASS PASS N/A PASS PASS PASS PASS N/A PASS N/A PASS N/A
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Read data in sector 1; Compare the data - matching (D9-1-3-3-1) LockignSP.Activate() Effect: Read data in sector 1; Compare the data - matching (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect check (D10-1-1-1-1) AdminSP.Revert Effect: The session to AdminSP; Revert response - pass (D10-1-2-1-2) AdminSP.Revert Effect: To ATA devices: check bit1 of word 82; bit1 of word 85; word (D10-1-2-1-3) AdminSP.Revert Effect: To ATA devices: check bit1 of word 82; bit1 of word 85; word (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured- Inativate) (D10-1-2-1-6) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: I= 0 or no data returned (D10-1-2-1-6) AdminSP.Revert Effect: LockingSP in inactive: Read data in sector 1; Compare the data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Read data in sector 1; Compare the data - matching (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-2-2-1-1) LockingSP.Revert Effect: The stassion remains open after issuing Locking.Revert() (D10-2-2-1-3) LockingSP.Revert Effect: CharA advices: check bit1 of word 82; bit1 of word 85; word (D1	PASS PASS PASS PASS PASS PASS PASS PASS
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Bread data in sector 1; Compare the data - matching (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect: The session within the AdminSP. Revert response - pass (D10-1-2-1-1) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured- Inactivate) (D10-1-2-1-5) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: I= 0 or no data returned (D10-1-2-1-5) AdminSP.Revert Effect: LockingSP in anctive: Read data in sector 1; Compare the data - matching (D10-1-2-1-3) AdminSP.Revert Effect: LockingSP in anctive: Read data in sector 1; Compare the data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Data in DBtStore table shall be the value (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in D10: LockingSP.Revert() Effect: LockingSP in active: Data in MBR table shall be the value (D10-1-2-2-1) LockingSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in D10: LockingSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in D10: LockingSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in D10-2-2-1-1) LockingSP.Revert Effect: LockingSP in active: Data in	PASS PASS PASS PASS PASS PASS PASS PASS
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SUS PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Read data in sector 1; Compare the data - matching (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect check (D10-1-1-1-1) AdminSP.Revert Grammar: Revert Session to AdminSP; Revert response - pass (D10-1-2-1-2) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be (D10-1-2-1-2) AdminSP.Revert Effect: To ATA devices: check bit1 of word 82; bit1 of word 85; word (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured- Inativate) (D10-1-2-1-5) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: I= 0 or no data returned (D10-1-2-1-6) AdminSP.Revert Effect: StartSession on AdminSP with MSID's PIN; SyncSession - pass (D10-1-2-3-1) AdminSP.Revert Effect: LockingSP in inactive: Read data in sector 1; Compare the data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Read data in sector 1; Compare the data - matching (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-2-2-1-1) LockingSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-2-2-1-3) LockingSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-2-2-1-3) LockingSP.Revert Effect: LockingSP in active: Data in DataStore table shal	PASS PASS PASS PASS PASS PASS PASS PASS
(D9-13-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-13-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-13-15) LockignSP.Activate() Effect: Read data in sector 1; Compare the data - matching (D9-13-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the (D9-13-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect: Check (D10-1-1-1) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be (D10-1-2-1-3) AdminSP.Revert Effect: Tor ATA devices: check bit 1 of word 82; bit 1 of word 85; word (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured- Inactivate) (D10-1-2-1-5) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: I= 0 or no data returned (D10-1-2-1-5) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: I= 0 or no data returned (D10-1-2-1-3) AdminSP.Revert Effect: LockingSP in anctive: Read data in sector 1; Compare the data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in D10-1-2-2-1) LockingSP.Revert Effect: CockingSP in active: Data in MBR table shall be the value in D10-2-2-1-1) LockingSP.Revert Effect: CockingSP in active: Data in MBR table shall be the value in D10-2-2-1-1) LockingSP.Revert Effect: CockingSP in active: Data in MBR table shall be the value in D10-2-2-1-1) LockingSP.Revert Effect: CockingSP in active: Data in MBR table shall be the value	PASS PASS PASS PASS PASS PASS PASS N/A PASS PASS PASS PASS N/A PASS PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-3-2) LockignSP.Activate() Effect: Bread data in sector 1; Compare the data - matching (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect: LockingSP in mfg state - PIN for Admin1 does not change D10: AdminSP.Revert() Effect: Check (D10-1-1-1) AdminSP.Revert Effect: The session within the AdminSP. Revert response - pass (D10-1-2-1-1) AdminSP.Revert Effect: Tha devices: check bit of word 82; bit 10 word 85; word (D10-1-2-1-3) AdminSP.Revert Effect: Tha devices: check bit of word 82; bit 10 word 85; word (D10-1-2-1-3) AdminSP.Revert Effect: Tha devices: check bit of word 82; bit 10 word 85; word (D10-1-2-1-4) AdminSP.Revert Effect: StartSession on LockingSP is in OFS(Manufactured/Manufactured- Inactivate) (D10-1-2-1-6) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: I= 0 or no data returned (D10-1-2-3-1) AdminSP.Revert Effect: LockingSP in inactive: Read data in sector 1; Compare the data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Read data in sector 1; Compare the data - (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in D10: LockingSP.Revert Effect: Check bit 10 down 82; bit 00 word 82; word (D10-2-2-1-1) LockingSP.Revert Effect: CheckingSP in active: Data in MBR table shall be the value in D10: LockingSP.Revert Effect: Check bit 10 down 82; bit 00 word 82; word (D10-2-2-1-3) LockingSP.Revert Effect: Check bit 10 down 82; bit 00 word 82; word (D10-2-2-1-3) LockingSP.Revert Effect: CheckingSP in active: Data in MBR table shall be the valu	PASS PASS PASS PASS PASS PASS N/A PASS PASS PASS PASS PASS PASS N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA

(D10-2-2-2-3) LockingSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in D10: RevertSP() Effect check (D10-3-1-1-1) LockingSP.RevertSP Grammar: RevertSP wothout parameters; RevertSP Response - (D10-3-2-1-1) LockingSP.RevertSP Condition: RevertSP if 'KeepGlobalRangeKey' = 1 and read- unlocked/write-unlocked for the Locking GlobalRange; RevertSP Response - Pass (D10-3-2-1-1) LockingSP.RevertSP Condition: RevertSP if 'KeepGlobalRangeKey' = 1 and read- unlocked for the Locking GlobalRange; RevertSP Response - Pass (D10-3-2-1-1) LockingSP.RevertSP Condition: RevertSP if 'KeepGlobalRangeKey' = 1 and write- unlocked for the Locking GlobalRange; RevertSP Response - Pass (D10-3-2-1-2) LockingSP.RevertSP Condition: RevertSP if 'KeepGlobalRangeKey' = 1 and write- unlocked for the Locking GlobalRange; RevertSP Response - 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) (D10-3-3-1-1) LockingSP.RevertSP Effect: The session shall be aborted: Get_Rqs for LifeCycleState	PASS N/A PASS PASS PASS PASS
(D10-3-1-1-1) LockingSP.RevertSP Grammar: RevertSP wothout parameters; RevertSP Response - (D10-3-2-1-1) LockingSP.RevertSP Condition: RevertSP If 'KeepGlobalRangeKey' = 1 and read- unlocked/write-unlocked for the Locking GlobalRange; RevertSP Response - Pass (D10-3-2-1-1) LockingSP.RevertSP Condition: RevertSP If 'KeepGlobalRangeKey' = 1 and read- unlocked for the Locking GlobalRange; RevertSP Response - Pass (D10-3-2-1) LockingSP.RevertSP Condition: RevertSP If 'KeepGlobalRangeKey' = 1 and write- unlocked for the Locking GlobalRange; RevertSP Response - 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) (D10-3-2-1-1) LockingSP.RevertSP Effect: The session shall be aborted: Get_Rqs for LifeCycleState	PASS PASS PASS
unlocked/write-unlocked for the Locking GlobalRange; RevertSP Response - Pass (D10-3-2-1-1) LockingSP.RevertSP Condition: RevertSP if 'KeepGlobalRangeKey' = 1 and read- unlocked for the Locking GlobalRange; RevertSP Response - Pass (D10-3-2-1-1) LockingSP.RevertSP Condition: RevertSP if 'KeepGlobalRangeKey' = 1 and write- unlocked for the Locking GlobalRange; RevertSP Response - 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) (D10-3-3-1-1) LockingSP.RevertSP Effect: The session shall be aborted: Get_Rqs for LifeCycleState	PASS PASS
unlocked for the Locking GlobalRange; RevertSP Response - Pass (D10-3-2-1-1) LockingSP.RevertSP Condition: RevertSP if 'KeepGlobalRangeKey' = 1 and write- unlocked for the Locking GlobalRange; RevertSP Response - 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) (D10-3-3-1-1) LockingSP.RevertSP Effect: The session shall be aborted: Get_Rqs for LifeCycleState	PASS
unlocked for the Locking GlobalRange; RevertSP Response - 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) (D10-3-3-1-1) LockingSP.RevertSP Effect: The session shall be aborted: Get_Rqs for LifeCycleState	
write-locked for the Locking GlobalRange; RevertSP Response - Fail(3Fh) (D10-3-3-1-1) LockingSP.RevertSP Effect: The session shall be aborted: Get_Rqs for LifeCycleState	PASS
	PASS
	N/A
(D10-3-3-1-4) LockingSP.RevertSP Effect: LifeCycleState = 08h (Manufactured-Inactivate)	PASS PASS
(D10-3-3-1-5) LockingSP.RevertSP Effect: StartSession on LockingSP; SyncSession - Status Code: I= 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
	NM PASS
(D10-3-3-2-5) LockingSP.RevertSP Effect: LockingSP in active: Data in MBR table shall be the value in D9-D10 Activate and Revert: ATA command check in RestrictedCommands table	N/A
(D9-1-3-1-6) RestrictedCmds: ATA command check after LockingSP.Activate	N/A
	N/A N/A
	N/A
D11: Power Cycle	
(PASS N/A
	N/A
Revert LockingSP	
	PASS PASS
	PASS
	PASS
	PASS
End Session - Response ** OPAL v2.0 - Generic **	PASS
Revert LockingSP	
	PASS PASS
	PASS
	PASS
	DACC
	PASS
	PASS
Protocol 2 Command Test Check Get_ComID command	
Protocol 2 Command Test Check Get_ComID command Check Verify_ComID_Valid command	PASS N/A N/A
Protocol 2 Command Test Check Get_ComID command Check Verify_ComID_Valid command	PASS N/A
Protocol 2 Command Test Check Get_ComID_command Check Verify_ComID_Valid command Check Get_ComID_Rsp command Check SSC information Identify the device type from the TPerInfo table	PASS N/A N/A N/A PASS
Protocol 2 Command Test Check Get_ComID command Check Verity_ComID_Valid command Check Get_ComID_Rsp command Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00	PASS N/A N/A N/A
Protocol 2 Command Test Check Get_ComID_command Check Verify_ComID_Valid command Check Get_ComID_Rsp command Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry information	PASS N/A N/A N/A PASS
Protocol 2 Command Test Check Get_ComID_command Check Verity, ComID_Valid command Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table	PASS N/A N/A N/A PASS N/A N/A PASS
Protocol 2 Command Test Check Get_ComID_command Check Verlfy_ComID_Valid command Check Get_ComID_Rsp command Check Get_ComID_Rsp command Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC V2.00 Verlfy Geometry infornmation Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verlfy Geometry Info between LockingInfo table and Level0_Discovery	PASS N/A N/A N/A PASS N/A N/A
Protocol 2 Command Test Check Get_ComID_command Check Verify_ComID_valid command Check Set_ComID_Rsp command Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery Ther Reset Command Test	PASS N/A N/A N/A PASS N/A N/A PASS
Protocol 2 Command Test Check Get_ComID_command Check Verlfy_ComID_Valid command Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verlfy Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verlfy Geometry Info between LockingInfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted	PASS N/A N/A PASS N/A N/A PASS N/A PASS PASS
Protocol 2 Command Test Check Get_ComID_cvalid command Check Verify_ComID_valid command Check Get_ComID_Rsp command Check SSC information Identify the device type from the TPerinfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset command: set ProgrammaticResetEnable=1 in the TPerInfo table	PASS N/A N/A PASS N/A PASS N/A PASS PASS PASS
Protocol 2 Command Test Check Get_ComID command Check Verify, ComID_Valid command Check SSC information Identify the device type from the TPerInfo table Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset command: set ProgrammaticResetEnable=1 in the TPerInfo table All open session SHALL be aborted on all ComID	PASS N/A N/A PASS N/A N/A PASS N/A PASS PASS PASS PASS
Protocol 2 Command Test Check Get_ComID_valid command Check Verify_ComID_valid command Check Get_ComID_Rsp command Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry Informmation Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset command: set ProgrammaticResetEnable=1 in the TPerInfo table All uncommitted transactions SHAII be aborted on all ComID All uncommitted transactions SHAII be reset to its initial state	PASS N/A N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS
Protocol 2 Command Test Check Get_ComID command Check Verify, ComID_Valid command Check SSC information Identify the device type from the TPerInfo table Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset is disabled; Issue TPer_Reset - aborted All open session SHALL be aborted on all ComID All uncommitted transactions SHAll be aborted on all ComID The synchronous protocol stack for all ComID SHALL be reset to its initial state All related method processing occurring on all ComIDS SHALL be aborted	PASS N/A N/A PASS N/A N/A PASS PASS PASS PASS PASS PASS PASS
Protocol 2 Command Test Check Get_ComID command Check Verify_ComID_Valid command Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset is disabled; Issue TPer_Reset - aborted All open session SHALL be aborted on all ComID All uncommitted transactions SHAIL be aborted on all ComID The synchronous protocol stack for all ComID SHAIL be aborted Hi related method processing occurring on all ComID SHALL be aborted Host's communications capabilities SHAII be reset to the initial state Host's communications capabilities SHAII be reset to the initial minimum assumptions	PASS N/A N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS
Protocol 2 Command Test Check Get_ComID command Check Verity, ComID_Valid command Check SSC information Identify the device type from the TPerInfo table Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the Lockinginfo table Verify Geometry Info between Lockinginfo table and Level0_Discovery Contents of column 07-0Ah returned from the Lockinginfo table Verify Geometry Info between Lockinginfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset command: set ProgrammaticResetEnable=1 in the TPerInfo table All open session SHALL be aborted on all ComID All uncommitted transactions SHAII be aborted on all ComID The synchronous protocol stack for all ComID SHAIL be reset to its initial state All related method processing occurring on all ComIDS SHAIL be aborted Host's communications capabilities SHAII be reset to the initial minimum assumptions Read/WriteLocked = True for all Locking objects if the LockOnReset = Programmatic enumeration Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration	PASS N/A N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
Protocol 2 Command Test Check Get_ComID command Check Verity, ComID_Valid command Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from tevel0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset solutions SHAIL be aborted on all ComID All open session SHALL be aborted on all ComID The synchronous protocol stack for all ComID SHALL be aborted Host's communications capabilities SHAII be reset to the initial state All related method processing occurring on all ComIDS SHALL be aborted Host's communications capabilities SHAII be reset to the initial minimum assumptions Read/WriteLocked = True for all Locking objects if the LockOnReset = Programmatic enumeration value Check Read/WriteLocked for all Locking objects before and after TPer_Reset is disabled	PASS N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
Protocol 2 Command Test Check Get_ComID command Check Verity, ComID_Valid command Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from tevel0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset solutions SHAIL be aborted on all ComID All open session SHALL be aborted on all ComID The synchronous protocol stack for all ComID SHALL be aborted Host's communications capabilities SHAII be reset to the initial state All related method processing occurring on all ComIDS SHALL be aborted Host's communications capabilities SHAII be reset to the initial minimum assumptions Read/WriteLocked = True for all Locking objects if the LockOnReset = Programmatic enumeration value Check Read/WriteLocked for all Locking objects before and after TPer_Reset is disabled	PASS N/A N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
Protocol 2 Command Test Check Get_ComID command Check Verity, ComID_Valid command Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from tevel0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset solutions SHAIL be aborted on all ComID All open session SHALL be aborted on all ComID All related method processing occurring on all ComIDS SHALL be aborted Host's communications capabilities SHAII be reset to the initial state All related method processing occurring on all ComIDS SHALL be aborted Host's communications capabilities SHAII be reset to the initial minimum assumptions Read/WriteLocked = True for all Locking objects if the LockOnReset = Programmatic enumeration Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration value Check Read/WriteLocked for all Locking objects before and after TPer_Reset is disabled Stack Reset Test Check the support of Stack_Reset command	PASS N/A N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS N/A PASS N/A PASS N/A PASS
Protocol 2 Command Test Check Get_ComID_valid command Check Verify_ComID_valid command Check Set_ComID_Rsp command Check SSC information Identify the device type from the TPerinfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted All open session SHALL be aborted on all ComID All uncommitted transactions SHAII be aborted on all ComID The synchronous protocol stack for all ComID SHAIL be reset to its initial state All related method processing occurring on all ComID SHAIL be aborted Host's communications capabilities SHAII be reset to the initial minimum assumptions Read/WriteLocked = True for all Locking objects if the LockOnReset = Programmatic enumeration Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration value Check Read/WriteLocked for all Locking objects before and after TPer_Reset is disabled Stack Reset Test Check the support of Stack_Reset response - Get_ComID_Rsp	PASS N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
Protocol 2 Command Test Check Get_ComID command Check Verify, ComID_Valid command Check SSC information Identify the device type from the TPerInfo table Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the Lockinginfo table Verify Geometry Info between Lockinginfo table and Level0_Discovery Contents of column 07-0Ah returned from the Lockinginfo table Verify Geometry Info between Lockinginfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset command: set ProgrammaticResetEnable=1 in the TPerInfo table All open session SHALL be aborted on all ComID All uncommitted transactions SHAIL be aborted on all ComID The synchronous protocol Stack for all ComID DS SHAIL be reset to the initial state All related method processing occurring on all ComIDS SHAIL be aborted Host's communications capabilities SHAII be reset to the initial istate All related method processing occurring on all ComIDS SHAIL be aborted Host's communications capabilities SHAII be reset to the initial minimum assumptions Read/WriteLocked = True for all Locking objects if the LockNenset = Programmatic enumeration Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration value Check Read/WriteLocked for all Locking objects if the LockNenset = TPer_Reset is disabled Stack Reset Test Check the support of Stack_Reset response - Get_ComID_Rsp All open session for that LOCMID SHAIL be aborted	PASS N/A N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS N/A PASS N/A PASS N/A PASS
Protocol 2 Command Test Check Get_ComID_valid command Check Verify_ComID_valid command Check Set_ComID_Rsp command Check SSC information Identify the device type from the TPerinfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted All open session SHALL be aborted on all ComID All uncommitted transactions SHAII be aborted on all ComID All uncommitted transactions SHAII be aborted on all ComID SHALL be aborted on all ComID SHAIL be reset to its initial state All related method processing occurring on all ComIDS SHALL be aborted Host's communications capabilities SHAII be reset to the initial minimum assumptions Read/WriteLocked = True for all Locking objects if the LockOnReset = Programmatic enumeration Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration value Check Read/WriteLocked for all Locking objects before and after TPer_Reset is disabled Stack Reset Test Check the support of Stack, Reset command The data returned from Stack, Reset rsponse - Get_ComID_Rsp All open session for that ComID SHALL be aborted All related method on that ComID SHALL be aborted	PASS N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
Protocol 2 Command Test Check Get_ComID command Check Verity, ComID_Valid command Check SSC information Identify the device type from the TPerInfo table Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the Lockinginfo table Verify Geometry info between Lockinginfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset command: set ProgrammaticResetEnable=1 in the TPerInfo table All open session SHALL be aborted on all ComID All uncommitted transactions SHAIL be aborted on all ComID All related method processing occurring on all ComIDS SHALL be aborted Hort's communications capabilities SHAII be reset to the initial state All related method processing occurring on all ComIDS SHALL be aborted Host's communications capabilities SHAII be reset to the initial minium assumptions Read/WriteLocked = True for all Locking objects if the LockonReset = Programmatic enumeration Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration value Check Read/WriteLocked for all Locking objects if the LockonReset = Programmatic enumeration Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration value Check Read/WriteLocked for all Locking objects if the LockonReset = Programmatic enumeration Stack Reset Test Check the support of Stack_Reset command The data returned from Stack_Reset orsponse - Get_ComID_Rsp All open session for that ComID SHALL be aborted All related method on that ComID SHALL be aborted The protocol stack for all ComID SHALL be aborted The protocol stack for all ComID SHALL be aborted	PASS N/A N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
Protocol 2 Command Test Check Get_ComID command Check Verity, ComID_Valid command Check SSC information Identify the device type from the TPerInfo table Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset is disabled; Issue TPer_Reset - aborted All open session SHALL be aborted on all ComID All uncommitted transactions SHAII be aborted on all ComID The synchronous protocol stack for all ComID SHAIL be aborted Host's communications capabilities SHAII be reset to the initial state All related method processing occurring on all ComID SHAIL be aborted Host's communications capabilities SHAII be reset to the initial minium assumptions Read/WriteLocked or True for all Locking objects if the LockOnReset = Programmatic enumeration Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration Done = false in MBRControl table before and after TPer_Reset is disabled Check Read/WriteLocked for all Locking objects before and after TPer_Reset is disabled Check the support of Stack_Reset command The data returned from Stack_Reset command The data returned from Stack_Reset retore and after TPer_Reset is disabled All open sension for that ComID SHALL be aborted All uncommitted transactions SHALL be aborted All uncommitted transactions SHALL be neset to its initial state All communications properties SHALL be reset to the initial state All communications properties SHALL be reset to the initial state	PASS N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
Protocol 2 Command Test Check Get_ComID command Check Verify, ComID_Valid command Check SSC information Identify the device type from the TPerInfo table Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the Lockinginfo table Verify Geometry Info between Lockinginfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset to disabled; Issue TPer_Reset - aborted Enable TPer_Reset command: set ProgrammaticResetEnable=1 in the TPerInfo table All open session SHALL be aborted on all ComID All uncommitted transactions SHAIL be aborted on all ComID All related method processing occurring on all ComID SHAIL be aborted Hort's communications capabilities SHAII be reset to the initial state All related method processing occurring on all ComID SHAIL be aborted Host's communications capabilities SHAII be reset to the initial minium assumptions Read/WriteLocked = True for all Locking objects if the LockNneset = Programmatic enumeration Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration Done = false in MBRControl table before and after TPer_Reset is disabled Stack Reset Test Check the support of Stack_Reset roponse - Get_ComID_Rsp All open session for that ComID SHALL be aborted All uncommitted transactions SHAIL be aborted All open session for that ComID SHALL be aborted All related method on that COMID SHALL be aborted All open session for that ComID SHALL be aborted All related method on that C	PASS N/A N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
Protocol 2 Command Test Check Get_ComID_valid command Check Verify_ComID_valid command Check Get_ComID_Rsp command Check Set_ComID_Rsp command Check Description Comments of column 07-0Ah returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery Ther Reset Command Test Check the support of Ther_Reset command If Ther_Reset is disabled; issue Ther_Reset - aborted All open session SHALL be aborted on all ComID The synchronous protocol stack for all ComID SHAIL be aborted All uncommitted transactions SHAID be aborted on all ComID The synchronous protocol stack for all ComID SHAIL be aborted Check Read/WriteLocked or all Locking objects if the LockOnReset = Programmatic enumeration Done - false in MBRControl table if the DoneOnReset = Programmatic enumeration Done - false in MBRControl table if the DoneOnReset = Programmatic enumeration Done - false in MBRControl table if the DoneOnReset = Programmatic enumeration Done - false in MBRControl table before and after TPer_Reset is disabled Check Done in the MBRControl table before and after TPer_Reset is disabled Check Done in the MBRControl table before and after TPer_Reset is disabled Check None in the MBRControl table before and after TPer_Reset is disabled All open session for MIL ComID SHALL be aborted All uncommitted transactions SHALL be aborted All uncommitted transactions SHALL be aborted All open session for MIL ComID SHALL be aborted Check Nead/Drese Nations SHALL be aborted	PASS N/A N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
Protocol 2 Command Test Check Get_ComID command Check Verity, ComID_Valid command Check SSC information Identify the device type from the TPerInfo table Check Bey ComID_Rsp command Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the Lockinginfo table Verify Geometry Info between Lockinginfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset command: set ProgrammaticResetEnable=1 in the TPerInfo table All open session SHALL be aborted on all ComID All uncommitted transactions SHAIL be aborted on all ComID The synchronous protocol tack for all ComID SHAIL be reset to the initial state All related method processing occurring on all ComIDS SHALL be aborted Host's communications capabilities SHAII be reset to the initial minium assumptions Read/WriteLocked = True for all Locking objects if the LockNeeset = Programmatic enumeration Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration Done = false in the MBRControl table before and after TPer_Reset is disabled Stack Reset Test Check Done in the MBRControl table before and after TPer_Reset is disabled All open session for that ComID SHALL be aborted All uncommitted transactions SHALL be aborted All uncommitted transactions SHALL be aborted All uncommitted transactions SHALL be aborted All open session for that ComID SHALL be abor	PASS N/A N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
Protocol 2 Command Test Check Get_ComID_valid command Check Verify_ComID_valid command Check Set_ComID_Rsp command Check SSC information Identify the device type from the TPerinfo table Check the support of OPAL SSC v2.00 Verify Geometry informmation Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery Ther Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset command: set ProgrammaticResetEnable=1 in the TPerInfo table All open session SHALL be aborted on all ComID The synchronous protocol stack for all ComID SHAIL be aborted Not's communications capabilities SHAII be aborted on all ComID The synchronous protocol stack for all ComID SHAIL be aborted Not's communications capabilities SHAI be aborted on all ComID Done - false in MBRControl table if the DoneOnReset = Programmatic enumeration Done - false in MBRControl table if the DoneOnReset = Programmatic enumeration Done - false in MBRControl table if the DoneOnReset = Programmatic enumeration Done - false in MBRControl table if the DoneOnReset = Programmatic enumeration Done - false in MBRControl table before and after TPEr_Reset is disabled Check Reset Test Check Reset Test All ocommunications sprose - Get_ComID_Rsp All open session for that ComID SHALL be aborted All renominated transactions SHALL be aborted All renominated transactions SHALL be aborted All communications properties SHALL be reset to this initial state All communications properties SHALL be reset to the in idefault values No Response Available if no handle_ComID_Request command preceded the Get_ComID_Response Check Yoael value in MBRControl table Stack, Reset with non-zero reserved byte; It shall be ignored by both hos	PASS N/A N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
Protocol 2 Command Test Check Get_ComID_valid command Check Verify_ComID_valid command Check SSC information Identify the device type from the TPerinfo table Check SSC information Identify the device type from the TPerinfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset command: set ProgrammaticResetEnable=1 in the TPerInfo table All open session SHALL be aborted on all ComID All uncommitted transactions SHAII be aborted on all ComID The synchronous protocol stack for all ComID SHAIL be reset to its initial state All related method processing occurring on all ComIDS SHALL be aborted Host's communications capabilities SHAII be reset to the initial minimum assumptions Read/WriteLocked = True for all Locking objects if the LockOnReset = Programmatic enumeration Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration Done = false in MBRControl table before and after TPer_Reset is disabled Stack Reset Test Check the support of Stack, Reset command The data returned from Stack, Reset rsponse - Get_ComID_Rsp All open session for that ComID SHALL be aborted All related method on that ComID SHALL be aborted All related method on SHALL be aborted All related method on Stack, Reset tomine in initial state All communications properties SHALL be reset to tis initial state All communications properties SHALL be aborted All related method on that ComID SHALL be reset to tis initial state All communications properties SHALL be aborted All related method on that ComID SHALL be reset to tis initial state All communications properties SHALL be aborted A	PASS N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
Protocol 2 Command Test Check Get_ComID_command Check Verity, ComID_valid command Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the Lockinginfo table Verify Geometry Info between Lockinginfo table and Level0_Discovery Contents of column 07-0Ah returned from the Lockinginfo table Verify Geometry Info between Lockinginfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset command: set ProgrammaticResetEnable=1 in the TPerInfo table All open session SHALL be aborted on all ComID All uncommitted transactions SHAIL be aborted on all ComID All related method processing occurring on all ComIDS SHALL be aborted Hort's communications capabilities SHAII be reset to the initial state All related method processing occurring on all ComIDS SHALL be aborted Host's communications capabilities SHAII be reset to the initial minium assumptions Read/WriteLocked = True for all Locking objects if the LockNneset = Programmatic enumeration Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration Done = false in the MBRControl table before and after TPer_Reset is disabled Stack Reset Test Check Done in the MBRControl table before and after TPer_Reset is disabled All open session for that ComID SHALL be aborted All related method on that ComID SHALL	PASS N/A N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
Protocol 2 Command Test Check Get_ComID_valid command Check Set, ComID_Rsp command Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset command: set ProgrammaticResetEnable=1 in the TPerInfo table All uncommitted transactions SHAII be aborted on all ComID All uncommitted transactions SHAII be aborted on all ComID The synchronous protocol stack for all ComID SHAIL be reset to its initial state All related method processing occurring on all ComIDS SHAIL be aborted Host's communications capabilities SHAII be reset to the initial minimum assumptions Read/WriteLocked = True for all Locking objects if the LockOnReset = Programmatic enumeration Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration value Check Read/WriteLocked for all Locking objects before and after TPer_Reset is disabled Stack Reset Test Check the support of Stack, Reset command The data returned from Stack, Reset command The data returned from Stack, Reset command All related method on that ComID SHAIL be aborted All related method on Stack, Reset to the initial state All communications properties SHAIL be aborted All related method on that ComID SHAIL be reset to the initial state All communications properties SHAIL be aborted All related method on that ComID SHAIL be reset to the initial state All communications properties SHAIL be aborted All related method on that ComID SHAIL be reset to the initial state All communications properties SHAIL be reset to the ini	PASS N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
Protocol 2 Command Test Check Get_ComID_valid command Check Set, ComID_Rsp command Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset command: set ProgrammaticResetEnable=1 in the TPerInfo table All uncommitted transactions SHAII be aborted on all ComID All uncommitted transactions SHAII be aborted on all ComID The synchronous protocol stack for all ComID SHAIL be reset to its initial state All related method processing occurring on all ComIDS SHAIL be aborted Host's communications capabilities SHAII be reset to the initial minimum assumptions Read/WriteLocked = True for all Locking objects if the LockOnReset = Programmatic enumeration Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration value Check Read/WriteLocked for all Locking objects before and after TPer_Reset is disabled Stack Reset Test Check the support of Stack, Reset command The data returned from Stack, Reset command The data returned from Stack, Reset command All related method on that ComID SHAIL be aborted All related method on Stack, Reset to the initial state All communications properties SHAIL be aborted All related method on that ComID SHAIL be reset to the initial state All communications properties SHAIL be aborted All related method on that ComID SHAIL be reset to the initial state All communications properties SHAIL be aborted All related method on that ComID SHAIL be reset to the initial state All communications properties SHAIL be reset to the ini	PASS N/A N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
Protocol 2 Command Test Check Get_ComID_valid command Check Set_ComID_Rsp command Check Get_ComID_Rsp command Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset command: set ProgrammaticResetEnable=1 in the TPerInfo table All uncommitted transactions SHAII be aborted on all ComID All uncommitted transactions SHAII be aborted on all ComID The synchronous protocol stack for all ComID SHAIL be reset to its initial state All related method processing occurring on all ComID SHAIL be aborted Host's communications capabilities SHAII be reset to the initial minimum assumptions Read/WriteLocked = True for all Locking objects if the LockOnReset = Programmatic enumeration Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration value Check Read/WriteLocked for all Locking objects before and after TPer_Reset is disabled Stack Reset Test Check the support of Stack_Reset command The data returned from Stack_Reset response - Get_ComID_Rsp All open session for that ComID SHALL be aborted All related method on that ComID SHALL be reset to its initial state All communications properties SHALL be aborted All communications properties SHALL be reset to its initial state All communications properties SHALL be aborted All related method on that ComID SHALL be reset to its initial state All communications properties SHALL be reset to its initial state All communications properties SHALL be reset to its initial state All communications properties SHA	PASS N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
Protocol 2 Command Test Check Get_ComID_command Check Verity, ComID_valid command Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the Lockinginfo table Verify Geometry Info between Lockinginfo table and Level0_Discovery Contents of column 07-0Ah returned from the Lockinginfo table Verify Geometry Info between Lockinginfo table and Level0_Discovery Ther Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset command: set ProgrammaticResetEnable=1 in the TPerInfo table All open session SHALL be aborted on all ComID All uncommitted transactions SHAIL be aborted on all ComID The synchronous protocol Tack for all ComID SHAIL be reset to its initial state All related method processing occurring on all ComIDS SHALL be aborted Host's communications capabilities SHAII be reset to the initial minium assumptions Read/WriteLocked = True for all Locking objects if the LockNneset = Programmatic enumeration Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration Done = false in the MBRControl table before and after TPer_Reset is disabled Stack Reset Test Check Done in the MBRControl table before and after TPer_Reset is disabled All open session for that ComID SHALL be aborted All open session for that ComID SHALL be aborted All related method on that ComID SHALL be aborted All related method on that ComID SHALL be aborted All related method on that ComID SHALL be aborted All reparted mothaset, Reset raponse - Get_ComID_Rsp All reparted method on that ComID SHALL be aborted All reparted method on that ComID SHALL be aborted Check 'ReadIocked' and 'WriteLocked' va	PASS N/A N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
Protocol 2 Command Test Check Get_ComID_valid command Check Set_ComID_kapi dommand Check Set_ComID_kapi command Check SSC information Identify the device type from the TPerinfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery Contents of column 07-0Ah returned from the LockingInfo table Verify Geometry Info between LockingInfo table and Level0_Discovery TPer Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset command: set ProgrammaticReseEnable=1 in the TPerInfo table All open session SHALL be aborted on all ComID The synchronous protocol stack for all ComID SHAII be reset to its initial state All related method processing occurring on all ComID SHAIL be aborted Hoat's communications capabilities SHAIL be reset to the initial minimum assumptions Read/WriteLocked = True for all Locking objects if the LockOnReset = Programmatic enumeration Done - false in MBRControl table if the DoneOnReset = Programmatic enumeration Done - false in MBRControl table if the DoneOnReset = Programmatic enumeration Done - false in MBRControl table before and after TPer_Reset is disabled Check Reset Test Check Reset Test Check Reset Test All oronnuications properties SHALL be aborted All related method on that ComID SHALL be reset to its initial state All communications properties SHALL be reset to its initial state All communications properties SHALL be reset to its initial state All communications properties SHALL be reset to its initial state All communications properies SHALL be reset to its initial state All commu	PASS N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
Protocol 2 Command Test Check Get_ComID_command Check Verity, ComID_valid command Check SSC information Identify the device type from the TPerInfo table Check the support of OPAL SSC v2.00 Verify Geometry Information Geometry Reporting Feature returned from Level0_Discovery Contents of column 07-0Ah returned from the Lockinginfo table Verify Geometry Info between Lockinginfo table and Level0_Discovery Contents of column 07-0Ah returned from the Lockinginfo table Verify Geometry Info between Lockinginfo table and Level0_Discovery Ther Reset Command Test Check the support of TPer_Reset command If TPer_Reset is disabled; Issue TPer_Reset - aborted Enable TPer_Reset command: set ProgrammaticResetEnable=1 in the TPerInfo table All open session SHALL be aborted on all ComID All uncommitted transactions SHAIL be aborted on all ComID The synchronous protocol tack for all ComID SHAIL be reset to its initial state All related method processing occurring on all ComIDS SHALL be aborted Host's communications capabilities SHAII be reset to the initial minium assumptions Read/WriteLocked = True for all Locking objects if the LockNneset = Programmatic enumeration Done = false in MBRControl table if the DoneOnReset = Programmatic enumeration Done = false in MBRControl table before and after TPer_Reset is disabled Stack Reset Test Check Done in the MBRControl table before and after TPer_Reset is disabled All open session for that ComID SHALL be aborted All related method on that ComID SHALL be aborted All related method on that ComID SHALL be aborted All related method on that ComID SHALL be aborted All repartemed from Stack, Reset reponse - Get_ComID_Rsp All reparted method on that ComID SHALL be aborted All repartemed from Stack, Reset reponse - Get_ComID_Rsp All reparted method on that ComID SHALL be aborted The protocol stack for all CockID be aborted All reparted method on that ComID SHALL be aborted All reparted method on that ComID SHALL be aborted All reparted method on that ComID SHA	PASS N/A N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA

Check the state of LockingSP	PASS
End Session - Request End Session - Response	PASS PASS
Check Authenticate method	
Check the support of Authenticate method - AdminSP	PASS
Authenticate - SID; Authenticate Response - Success(AuthStatus = 01h) Authenticate - Admin1(non-authorized UID); Authenticate Response - Fail(AuthStatus = 0h)	PASS 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) Authenticate - User2 (non-authority UID); Authenticate Response - Fail(AuthStatus = 0h)	PASS 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 Random Request with count = 20h in AdminSP; Random Response - Success	PASS PASS
Random Request with count > 20h in AdminSP; Random Response - Success or	
StatusCode=0Ch(Invalid_Param)	PASS
Check the support of Random method - LockingSP Random Request with count < 20h in LockingSP; Random Response - Success	PASS PASS
Random Request with count = 20h in LockingSP; Random Response - Success	PASS
Random Request with count > 20h in LockingSP; Random Response - Success or	
StatusCode=OCh(Invalid_Param) Alignment LBA Test	PASS
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:	N/A
OCh(Invalid_Prams)	N/A
Data Alignment Restriction on Byte Table - DataStore	
Get MandatoryWriteGranularity and RecommendedAccessGranularity of DataStore from Table table MandatoryWriteGranularity of DataStore SHALL be less than or equal to 8192	PASS
Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: OCh(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 Get and Compare data from DataStore - Matching	PASS PASS
Data Alignment Restriction on Byte Table - MBR	FA33
Get MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table	N/A
MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 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: OCh(Invalid_Param)	N/A 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: OCh(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 End Session - Request	PASS 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) (C3-3) SPTemplates: Next() method for table (AdminSP)	PASS 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) (C3-4) MethodID: Get the entries from table (AdminSP)	PASS 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) (C3-6) ACE: Verify the table contents (AdminSP)	PASS
(C3-7) Authority: Next() method for table (AdminSP)	PASS 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) (C3-8) C_PIN: Get the entries from table (AdminSP)	PASS 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) (C3-10) Template: Next() method for table (AdminSP)	PASS 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) (C3-11) SP: Get the entries from table (AdminSP)	PASS PASS
(C3-11) SP: Get the entries from table (AdminSP) (C3-11) SP: Verify the table contents (AdminSP)	PASS
Activating the Locking SP	
Start Session with HostChallenge - AdminSP	PASS
Sync Session - AdminSP Activate _LockingSP	PASS PASS
Activate_LockingSP - Response	PASS
Get - LifeCycle(Locking SP) - Request	PASS PASS
Get - LifeCycle(Locking SP) - Response	r M33

Check the state of LockingSP	PASS
End Session - Request End Session - Response	PASS PASS
C3: Get() Object Table Contents to LockingSP	1,435
(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) (C3-14) SPTemplates: Next() method for table (LockingSP)	PASS PASS
(C3-14) SPTemplates: Next() method for table (LockingSP) (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) (C3-18) ACE: Verify the table contents (LockingSP)	PASS 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 PASS
(C3-21) LockingInfo: Get the entries from table (LockingSP) (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 N/A
(C3-) SecretProtect: Get the entries from table (LockingSP) (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)	PASS
(C5-1) Next() - Table Table (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 (C5-4) Next() - MethodID Table	PASS 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 PASS
(C5-6) Verify ACL values for Authority Table (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 (C5-9) Verify ACL values for Template Table	PASS 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 PASS
(C5-11) Verify ACL values for Table Table (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 (C5-16) Next() - ACE Table	PASS 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 (C5-18) Verify ACL values for C_PIN Table	PASS PASS
(C5-18) Verify ACL values for C_PIN Table (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 (C5-22) GetACL() - MBR Table	N/A N/A
(C5-22) GetAct() - MBR Table (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 (C5-24) Verify ACL values for DataStore Table	PASS 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 (C5-26) GetACL() - RestrictedCmds Table	N/A N/A
Revert LockingSP	N/A
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 Activating the Locking SP	PASS
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 Check the state of LockingSP	PASS 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 Check the new entries added to the AccessControl table	N/A 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 Reactivate() method with DataStore size (<= maxDSSize); Response - Pass	N/A 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 N/A
Check the support of ReActivate and Erase methods in the MethodID table 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 Activate() method with RangeN(N=LockingInfo.MaxRanges); Response - Pass	N/A 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 Activate() method with empty ObjList and 'RangeStartLenPolicy'=1; Response - pass	N/A 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 Verify 'Policy'=0; 'All'=0; 'Any'=1 from Level0_Discovery	N/A 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)	
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 Activate w/ entireLocking: Range1-GlobalRange.Set Request in LockingSP as User1-(N+1); Response -	N/A
StatusCode = 01h(Not_Authorized)	N/A
Activate w/ entireLocking: GlobalRange-RangeN.Set Request in LockingSP as User1-(N+1); Response -	
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 Activate w/ allLockingObj: Range1-GlobalRange.Set Request in LockingSP as User1-(N+1); Response -	N/A
StatusCode = 01h(Not_Authorized)	N/A
Activate w/ allLockingObj: GlobalRange-RangeN.Set Request in LockingSP as User1-(N+1); Response -	
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) ReActivate() method with LockingObject not included in Locking table: Response -	N/A
ReActivate() method with LockingObject not included in Locking table; Response - StatusCode=OCh(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 -	
Verify: StartSession to Locking SP as User(N+1)(N=MaxRanges/2); SyncSession - Pass	N/A
	N/A
ReActivate() method with RangeN(N=LockingInfo.MaxRanges); Response - Pass	N/A N/A
ReActivate() method with RangeN(N=LockingInfo.MaxRanges); Response - Pass Verify: StartSession to Locking SP as UserN(N=MaxRanges); SyncSession -	N/A N/A N/A
ReActivate() method with RangeN(N=LockingInfo.MaxRanges); Response - Pass Verify: StartSession to Locking SP as UserN(N=MaxRanges); SyncSession - Verify: StartSession to Locking SP as User(N+1)(N=MaxRanges); SyncSession - Pass	N/A N/A N/A N/A
ReActivate() method with RangeN(N=LockingInfo.MaxRanges); Response - Pass Verify: StartSession to Locking SP as UserN(N=MaxRanges); SyncSession -	N/A N/A N/A
ReActivate() method with RangeN(N=LockingInfo.MaxRanges); Response - Pass Verify: StartSession to Locking SP as UserN(N=MaxRanges); SyncSession - Verify: StartSesion to Locking SP as User(N+1)(N=MaxRanges); SyncSession - Pass ReActivate() with Admin1PIN=omitted - Succed	N/A N/A N/A N/A 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 Clear Events: Revert AdminSP	N/A
	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 PASS
Check SID Blocked State(=0) after Hardware Reset Subsequent invocation of Block SID Authentication command: Fail with 'Other Invalid Command	PASS
Revert LockingSP	PASS
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