ULINK TCG/IEEE1667 Opal Family SSC Protocol Test Result Script REV 10.0 (License ULINK_TW) Tested by ULINK DriveMaster Enterprise (NVME+DRV) (x64) Version 9.2.1800 (6 2 1 ACR) HBA NAME: PHISON(N) BUS=1 DEV VID=1987 [NVME 2.0.0 PCIE 5.0 SSD PS5026-E26 Model Number 6A91193C154900000091 Serial Number: FW Revision: EQFM22.3 Start Date: Mon 2024 February 0 Time: 12:11:20 PM 3907029168 (0xE8E088B0) Total LBA: 2000 G PASS Check PSID support Start Session - AdminSP PASS PASS Sync Session - AdminSP PASS Next Request - Authority table Next Response - Authority table PASS Check the PSID support PASS PASS End Session - Request PASS End Session - Response Revert LockingSP PASS Start Session with HostChallenge - AdminSP PASS Sync Session - AdminSF PASS PASS LockingSP.Revert - Request LockingSP.Revert - Response PASS End Session - Request **PASS** End Session - Response PASS ** Opal V1.0 - I1667 Test Cases ** N/A (A0-1-1-1-1) Word 48: Identify Device: bit 0 of word 48 shall be set to 1 N/A (A0-1-1-1-2) Word 119: Identify Device: bit 6 of word 119 = 1 - TPer supports Sense Data Reporting N/A A1: Test Trusted Send/Receive cases PASS $(A1\hbox{-} 1\hbox{-} 1\hbox{-} 1\hbox{-} 1)\,X fer Length: Trusted Send with SP=00h; Spcf=ComID; X fer=00h: Abort$ 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 N/A (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 N/A $(A1-2-3-2-3)\,Spcf; Trusted\,Receive\,with\,SP=01h; Spcf=Unsupported\,ComID (0-0FFFh); Xfer=01h; Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Abort\,Ab$ PASS A2: Test Protocol ID = 0 related cases PASS (A2-1-1-1-2) Spcf=0 DataXfer: TCG-Receive with SP=00h; Spcf=00h; Xfer=00h: Pass PASS (A2-1-1-1-1) Spcf=0 DataXfer: TCG-Receive with SP=00h; Spcf=00h; Xfer=01h: Pass PASS (A2-1-2-1-2) Spcf=0 DataContent: TCG-Receive with SP=00h; Spcf=00h; Xfer=01h: SP List-Byte6-7 >= 02h PASS $(A2-1-2-1-3(1))\ Spcf=0\ DataContent:\ TCG-Receive\ with\ SP=00h;\ Spcf=00h;\ Xfer=01h:\ SP\ list-Byte8=00h$ PASS $(A2-1-2-1-3(2)) \ Spcf=0 \ DataContent: \ TCG-Receive \ with \ SP=00h; \ Spcf=00h; \ Xfer=01h: \ SP \ list-Byte9=01h$ PASS $(A2-1-2-1-3(3)) \ Spcf=0 \ Data Content: \ TCG-Receive \ with \ SP=00h; \ Spcf=00h; \ Xfer=01h: \ SP \ list-Byte 10=02h (if \ supported) \ or \ 00h \ or \ 10h \ or \ or \ 10h \ or \ or \ 10h \ or$ PASS (A2-2-1-1-2) Spcf=1 DataXfer: TCG-Receive with SP=00h; Spcf=01h; Xfer=00h: Pass PASS (A2-2-1-1-1) Spcf=1 DataXfer: TCG-Receive with SP=00h; Spcf=01h; Xfer=01h: Pass PASS (A2-2-2-1-2) Spcf=1 DataXfer: TCG-Receive with SP=00h; Spcf=01h; Xfer=01h: Certificate-Byte2-3 = 00h or a value PASS A3: Test Level 0 Discovery Protocol **PASS** (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 PASS (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 (A4-1-1-1-3) IF Send: TPer in awaiting IF Send state - IF Send with SP=01h; Spcf=ComID; Xfer=01h; pass PASS $(A4-1-3-1-1)\ IF_Send:\ TPer\ in\ awaiting\ IF_Recv\ state\ -\ IF_Send\ with\ SP=01h;\ Spcf=ComID;\ Xfer=01h:\ abort$ PASS $(A4-2-1-2-1)\ IF_Recv: TPer\ in\ awaiting\ IF_Send\ state-IF_Recv(Level0\ discovery)\ with\ SP=01h;\ Spcf=01h;\ Xfer=01h:\ pass$ PASS $(A4-2-1-2-3)\ IF_Recv: TPer\ in\ awaiting\ IF_Send\ state-IF_Recv\ with\ SP=01h;\ Spcf=ComID;\ Xfer=01h:\ no\ data\ returned$ PASS (A4-2-3-2-1) IF_Recv: TPer in awaiting IF_Recv state - IF_Recv(Level0 discovery) with SP=01h; Spcf=01h; Xfer=01h: pass PASS $(A4-2-3-2-2)\ IF_Recv: TPer\ in\ awaiting\ IF_Recv\ state-IF_Recv\ with\ SP=01h;\ Spcf=ComID;\ Xfer=01h:\ All\ response\ returned\ no\ further\ data$ PASS (A4-2-3-2-3) IF_Recv: TPer in awaiting IF_Recv state - IF_Recv with Xfer=insufficient; TPer stays in awaiting IF_Recv state PASS A5: Check ComPacket/Packet/SubPacket PASS (A5-1-1-1-2) IF Send ComPacket - Reserved field != 0: IF Send: pass PASS (A5-1-2-2-2) IF_Send ComPacket - ComID != current ID; TPer in awaiting IF_Send state PASS (A5-1-2-5-1) IF_Send ComPacket - ComID Extension != 0; TPer in awaiting IF_Send state PASS (A5-1-3-1-2) IF_Send ComPacket - OutstandingData != 0; IF_Send: pass PASS (A5-1-4-1-2) IF_Send ComPacket - MinTransfer! = 0; IF_Send: pass PASS (A5-1-5-1-2) IF_Send ComPacket - Length > Xfer-data length; TPer in awaiting IF_Send state PASS (A5-1-5-1-2(2)) IF Send ComPacket - Length < 24; TPer in awaiting IF Send state PASS (A5-1-5-1-3) IF_Send ComPacket - Padding byte != 0; IF_Send: pass PASS

(A5-2-3-1-2) IF_Send Packet - Reserved field != 0; IF_Send: pass	PASS
(A5-2-6-1-2) IF_Send Packet - Length > Xfer-data length of Compacket; no data returned	PASS
(A5-2-6-1-2(2)) IF_Send Packet - Length < 12 of Subpacket; no data returned	PASS
(A5-3-1-1-1(1)) IF Send Packet - non-aligned with 4 byte in the start point of Subpacket; no data returned	N/A
(A5-3-1-1-2) IF Send SubPacket - Reserved field != 0; IF, Send: pass	PASS
(A5-3-3-1-2) IF_Send SubPacket - Length > Packet; no data returned	PASS
(A5-2-1-1-2) IF_Send Packet in regular session - Session ID != open session's 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 abort	PASS
(A5-2-6-1-2(2)) IF_Send Packet in regular session - Length < 12 of Subpacket; Session abort	PASS
(A5-3-3-1-2) IF Send SubPacket - Length > Packet; Session abort	PASS
(A0-0-0-1-2) IT_Sellid Subracket - Leligiti > racket, Session about	FASS
A7: Transaction check	PASS
(A7-1-1-2-1(2)) StartTransaction Request: status != 0; StartTransaction Response: Pass with status = 0	PASS
(A7-1-1-2-1(2)) StartTransaction Request: status = 0 with short atom(81h); StartTransaction Response: Pass	PASS
(A7-1-1-2-1(2)) StartTransaction Request: status = 0 with medium atom(C001h); StartTransaction Response: Pass	PASS
(A7-1-1-2-1(2)) StartTransaction Request: status = 0 with long atom(E0000001h); StartTransaction Response: Pass	PASS
(A7-1-1-2-5) StartTransaction Request: status = 0 with byte atom; StartTransaction Response: Session Abort	PASS
(A7-1-1-2-5) StartTransaction Request: status = 0 with integer atom; StartTransaction Response: Session Abort	PASS
(A7-1-1-2-6) StartTransaction Request: no status encoded; StartTransaction Response: Session Abort	PASS
(A7-1-2-2-1(1)) EndTransaction Request: status = 0 with short atom(81h); EndTransaction Response: Pass	PASS
(A7-1-2-2-1(1)) EndTransaction Request: status = 0 with medium atom(C001h); EndTransaction Response: Pass	PASS
(A7-1-2-2-1(1)) EndTransaction Request: status = 0 with long atom(E0000001h); EndTransaction Response: Pass	PASS
(A7-1-2-2-1(2)) EndTransaction Request: status != 0 with short atom(81h); EndTransaction Response: Pass with status != 0	PASS
(A7-1-2-2-1(2)) EndTransaction Request: status != 0 with medium atom(C001h); EndTransaction Response: Pass with status != 0	PASS
(A7-1-2-2-1(2)) EndTransaction Request: status != 0 with long atom(E0000001h); EndTransaction Response: Pass with status != 0	PASS
(A7-1-2-2-5) EndTransaction Request: status = 0 with byte atom; EndTransaction Response: Session Abort	PASS
(A7-1-2-2-5) EndTransaction Request: status = 0 with integer atom; EndTransaction Response: Session Abort	PASS
(A7-1-2-2-6) EndTransaction Request: no status encoded; EndTransaction Response: Session Abort	PASS
(A7-1-3-1-1) Trans-Start attempt: StartTransaction <= MaxTransLimit; Response: pass	PASS
(A7-1-3-1-2) Trans-Start attempt: StartTransaction > MaxTransLimit; Response: Session Abort	PASS
(A7-1-3-2-1) Trans-End attempt: EndTransaction Request: outside of a transaction with status = 0; Response: Session Abort	PASS
(A7-1-3-2-2) Trans-End attempt: EndTransaction Request: within a transaction with status = 0; Response (commit): pass	PASS
(A7-1-3-3-1) Trans-Abort attempt: EndTransaction Request: outside of a transaction with status = 1; Response: Session Abort	PASS
(A7-1-3-3-2) Trans-Abort attempt: EndTransaction Request: within a transaction with status = 0; Response (abort): pass	PASS
(A7-1-3-4-1) Stand-Alone: StartTransaction Request: only with Start_Trans token and status token; Response: Pass	PASS
(A7-1-3-5-1) Stand-Alone: EndTransaction Request: only with End_Trans token and status token; Response: Pass	PASS
(A7-1-3-6-1) Multiple Trans: Trans-Start request after one or more; Response: Pass	PASS
(A7-1-3-8-1) Trans-attempt in a CtrlSession: Trans-Start request outside of methodinvoke: Token shall be discarded	PASS
(A7-1-3-8-1) Trans-attempt in a CtrlSession: Trans-End request outside of methodInvoke: Token shall be discarded	PASS
	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
(A7-1-7-1-1) Trans+Session Close: Transaction is aborted after session gets closed (A7-1-8-1-1) Trans+Session Close: Transaction is aborted after power cycle	PASS
(A7-1-8-1-1) Trans+Session Close: Transaction is aborted after power cycle	PASS
(A7-1-8-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session	PASS PASS
(A7-1-8-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned	PASS PASS PASS
(A7-1-8-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans and EndTrans; Session shall be closed	PASS PASS PASS PASS
(A7-1-8-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned	PASS PASS PASS
(A7-1-8-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans and EndTrans; Session shall be closed	PASS PASS PASS PASS
(A7-1-8-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 PASS PASS PASS PASS PASS
(A7-1-8-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-2-1-1) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-2-1-1) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A7-1-8-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-2-1-1) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-2-1-1) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-2-1-1) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1)(2)) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-2-1-1) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (A8-2-3-1-1) Session after CloseSession: Start a new session shall pass after the Session is aborted	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1)(2)) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-2-1-1) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1)(2)) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-2-1-1) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (A8-2-3-1-1) Session after CloseSession: Start a new session shall pass after the Session is aborted	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-2-1-1) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 (A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1)(2)) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-2-1-1) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-1-1-1) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-1) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 (A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-2-1-1) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-1) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-2-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 (A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (A9-1-1-1-1) StartSession - '0xFF' before a call token(0xF8); SyncSession: pass	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-2-1-1) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 (A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (A9-1-1-1-1) StartSession - '0xFF' before a call token (0xF8); SyncSession: pass (A9-1-1-2-1) StartSession - '0xFF' between a call token and an 'InvokingID'; SyncSession: pass	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-2-1-1) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session affer EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 (A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (A9-1-1-1-1) StartSession - '0xFF' before a call token (0xF8); SyncSession: pass (A9-1-1-2-1) StartSession - '0xFF' between a call token and an 'InvokingID'; SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded outside of a method invocation in a control session; End token shall be closed (A8-1-1-1-1(3)) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-1) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 (A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (A9-1-1-1-1) StartSession - '0xFF' between a call token (0xF8); SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass (A9-1-1-4-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass (A9-1-1-4-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass (A9-1-1-4-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass (A9-1-1-4-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-2-1-1) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session affer EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 (A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (A9-1-1-1-1) StartSession - '0xFF' before a call token (0xF8); SyncSession: pass (A9-1-1-2-1) StartSession - '0xFF' between a call token and an 'InvokingID'; SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded outside of a method invocation in a control session; End token shall be closed (A8-1-1-1-1(3)) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-1) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 (A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (A9-1-1-1-1) StartSession - '0xFF' between a call token (0xF8); SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass (A9-1-1-4-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass (A9-1-1-4-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass (A9-1-1-4-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass (A9-1-1-4-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-2-1-1) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-1) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-2-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 (A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (A9-1-1-1-1) StartSession - '0xFF' between a call token (0xF8); SyncSession: pass (A9-1-1-2-1) StartSession - '0xFF' between a call token and an 'InvokingID'; SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between a 'MethodID' and a 'MethodID'; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' among HostSID and SPUID parameters; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' among SPUID and Write paramters; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' among SPUID and Write paramters; SyncSession: pass	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + Methodinvoke and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded within StartTrans + Methodinvoke and EndTrans; Session shall be closed (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 (A8-1-2-1-1) EndSession Effect: Tere sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 infer 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 (A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (A9-1-1-1-1) StartSession - '0xFF' between a call token (0xF8); SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between a 'MethodID' and 'Po'; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' among HotstID and SPUID parameters; SyncSession: pass (A9-1-1-6-1) StartSession - '0xFF' among SPUID and Write parameters; SyncSession: pass (A9-1-1-6-1) StartSession - '0xFF' among SPUID and Write parameters; SyncSession: pass	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-1(3)) EndSession Grammar: End Session is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-2-1-1) EndSession Effect: Terr sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 iffer 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 (A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (A9-1-1-1-1) StartSession - '0xFF' before a call token (0xF8); SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between a 'MethodID and 'F0'; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' among SPUID and Write parameters; SyncSession: pass (A9-1-1-6-1) StartSession - '0xFF' between endList('F1') and endData('F9'); SyncSession: pass (A9-1-1-7-1) StartSession - '0xFF' between endList('F1') and endData('F9'); SyncSession: pass	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded outside of a method invocation in a control session; End token shall be closed (A8-1-1-1-1(3)) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-1) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-2-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 (A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (A9-1-1-1-1) StartSession - '0xFF' between a call token (0xF8); SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' between an 'MethodID' and 'F0'; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' between and Mrite parameters; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' between end List('F1') and endData('F9'); SyncSession: pass (A9-1-1-6-1) StartSession - '0xFF' between end List('F1') and endData('F9'); SyncSession: pass (A9-1-1-8-1) StartSession - '0xFF' between end List('F1') and endData('F9'); SyncSession: pass (A9-1-1-8-1) StartSession - '0xFF' between endData('F9') and endData('	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-1(3)) EndSession Grammar: End Session is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-2-1-1) EndSession Effect: Terr sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 iffer 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 (A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (A9-1-1-1-1) StartSession - '0xFF' before a call token (0xF8); SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between a 'MethodID and 'F0'; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' among SPUID and Write parameters; SyncSession: pass (A9-1-1-6-1) StartSession - '0xFF' between endList('F1') and endData('F9'); SyncSession: pass (A9-1-1-7-1) StartSession - '0xFF' between endList('F1') and endData('F9'); SyncSession: pass	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded outside of a method invocation in a control session; End token shall be closed (A8-1-1-1-1(3)) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-1) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-2-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 (A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (A9-1-1-1-1) StartSession - '0xFF' between a call token (0xF8); SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' between an 'InvokingID' and a 'MethodID'; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' between an 'MethodID' and 'F0'; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' between and Mrite parameters; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' between end List('F1') and endData('F9'); SyncSession: pass (A9-1-1-6-1) StartSession - '0xFF' between end List('F1') and endData('F9'); SyncSession: pass (A9-1-1-8-1) StartSession - '0xFF' between end List('F1') and endData('F9'); SyncSession: pass (A9-1-1-8-1) StartSession - '0xFF' between endData('F9') and endData('	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded outside of a method invocation in a control session; End token shall be discarded (A8-1-1-1-1) EndSession Effect: Ther sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 (A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (A9-1-1-1-1) StartSession - '0xFF' between a call token (0xF8); SyncSession: pass (A9-1-1-2-1) StartSession - '0xFF' between a 'MethodID' and 'FO'; SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between a 'MethodID' and 'FO'; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' among HostSID and SPUID parameters; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' between endList('F1') and endData('F9') SyncSession: pass (A9-1-1-8-1) StartSession - '0xFF' between endData('F9') and statusCode('F0'); SyncSession: pass (A9-1-1-8-1) StartSession - '0xFF' between endData('F9') and statusCode('F0'); SyncSession: pass (A9-1-1-9-1) StartSession - '0xFF' among tokens and statusCode list;	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded outside of a method invocation in a control session; End token shall be discarded (A8-1-1-1-1(3)) EndSession Effect: Ter sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 (A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (A9-1-1-2-1) StartSession - '0xFF' before a call token (0xF8); SyncSession: pass (A9-1-1-2-1) StartSession - '0xFF' between a call token (0xF8); SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between a 'MethodID' and 'Fo'; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' between endList('F1') and endData('F9'); SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' between endList('F1') and endData('F9'); SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' between endData('F9'); And endData('F9'); SyncSession: pass (A9-1-1-1-1) StartSession - '0xFF' between endData('F9') and statusCode('F0'); SyncSession: pass (A9-1-1-1-1) StartSession - '0xFF' between endData('F9	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded outside of a method invocation in a control session; End token shall be discarded (A8-1-2-1-1) EndSession Effect: Per sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 (A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (A9-1-1-2-1) StartSession - '0xFF' between a call token (0xF8); SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between a milnvokingID'; SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between a 'MethodID' and 'FD'; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' among SPUID and Write paramters; SyncSession: pass (A9-1-1-6-1) StartSession - '0xFF' between endList('FD') and endData('FD'); SyncSession: pass (A9-1-1-8-1) StartSession - '0xFF' between endData('FD') and endData('FD'); SyncSession: pass (A9-1-1-8-1) StartSession - '0xFF' between endData('FD') and endData('FD'); SyncSession: pass (A9-1-1-1-1) StartTransaction - '0xFF' between a Transactio	PASS PASS PASS PASS PASS PASS PASS PASS
(A7-1.8-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(3)) EndSession Grammar: EndSession is encoded within StartTrans + Methodinvoke and EndTrans; Session shall be closed (A8-1-1-1-1(3)) EndSession Grammar: End Session is encoded within StartTrans + Methodinvoke and EndTrans; Session shall be closed (A8-1-1-1) EndSession Effect: TPer sends an End of Session is encoded outside of a method invocation in a control session; End token shall be discarded (A8-1-2-1-2) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (A8-2-2-1-1) 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 (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 - '0xFF' between a call token(0xF8); SyncSession: pass (A9-1-1-2-1) StartSession - '0xFF' between a call token and an 'InvokingID'; SyncSession: pass (A9-1-1-4-1) StartSession - '0xFF' between a MethodID' and 'Fo'; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' mong HotSID and SPUID parameters; SyncSession: pass (A9-1-1-1-1) StartSession - '0xFF' between endData('F9') and statusCode ('F0'); SyncSession: pass (A9-1-1-1-1) StartSession - '0xFF' between endData('F9') and statusCode ('F0'); SyncSession: pass (A9-1-1-1-1) StartSession - '0xFF' bet	PASS PASS PASS PASS PASS PASS PASS PASS
A8: Test Ending Session (A8-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded outside of a method invocation in a control session; End token shall be discarded (A8-1-2-1-1) EndSession Effect: Per sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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 (A8-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (A9-1-1-2-1) StartSession - '0xFF' between a call token (0xF8); SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between a milnvokingID'; SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF' between a 'MethodID' and 'FD'; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' among SPUID and Write paramters; SyncSession: pass (A9-1-1-6-1) StartSession - '0xFF' between endList('FD') and endData('FD'); SyncSession: pass (A9-1-1-8-1) StartSession - '0xFF' between endData('FD') and endData('FD'); SyncSession: pass (A9-1-1-8-1) StartSession - '0xFF' between endData('FD') and endData('FD'); SyncSession: pass (A9-1-1-1-1) StartTransaction - '0xFF' between a Transactio	PASS PASS PASS PASS PASS PASS PASS PASS
(A7-1.8-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(3)) EndSession Grammar: EndSession is encoded within StartTrans + Methodinvoke and EndTrans; Session shall be closed (A8-1-1-1-1(3)) EndSession Grammar: End Session is encoded within StartTrans + Methodinvoke and EndTrans; Session shall be closed (A8-1-1-1) EndSession Effect: TPer sends an End of Session is encoded outside of a method invocation in a control session; End token shall be discarded (A8-1-2-1-2) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (A8-2-2-1-1) 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 (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 - '0xFF' between a call token(0xF8); SyncSession: pass (A9-1-1-2-1) StartSession - '0xFF' between a call token and an 'InvokingID'; SyncSession: pass (A9-1-1-4-1) StartSession - '0xFF' between a MethodID' and 'Fo'; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' mong HotSID and SPUID parameters; SyncSession: pass (A9-1-1-1-1) StartSession - '0xFF' between endData('F9') and statusCode ('F0'); SyncSession: pass (A9-1-1-1-1) StartSession - '0xFF' between endData('F9') and statusCode ('F0'); SyncSession: pass (A9-1-1-1-1) StartSession - '0xFF' bet	PASS PASS PASS PASS PASS PASS PASS PASS
(A7-1-8-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-(3)) EndSession Grammar: End Session is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-(3)) EndSession Grammar: End Session is encoded outside of a method invocation in a control session; End token shall be discarded (A8-1-2-1-1) EndSession Effect: TPer sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: TendSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (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# AmxSessions and a session is timeout; Start/Sync Session - pass (A8-3-4-1-1) Session Timeout: If session# AmxSessions and a session is timeout; Start/Sync Session - pass (A8-4-1-1-1) StartSession - '0xFF' between a call token (0xF8): SyncSession: pass (A9-1-1-2-1) StartSession - '0xFF between a full token and an 'InvokingID'; SyncSession: pass (A9-1-1-3-1) StartSession - '0xFF among HostSID and SPUID parameters; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF among SPUID and Write parameters; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF among SPUID and Write parameters; SyncSession: pass (A9-1-1-6-1) StartSession - '0xFF between endData(FP3): SyncSession: pass (A9-1-1-1-1) StartTransaction - '0xFF between endData(FP3): SyncSession: pass (A9-1-1-1-1) StartTransaction - '0xFF between a TransactionEnd token; Response: pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A7-1-8-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + Methodinvoke and EndTrans; Session shall be closed (A8-1-1-1-1(3)) EndSession Grammar: EndSession is encoded outside of a method invocation in a control session; End token shall be closed (A8-1-2-1-1) EndSession Effect: Ter sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (A8-2-3-1-1) Session after CloseSession: Start a new session shall pass after the Session is aborted (A8-3-1-1) Session Timeout: If sessionie = MaxSessions and a session is timeout; Start/Sync Session - pass (A8-3-4-1-1) Session Timeout: If sessionie = MaxSessions and a session is timeout; Start/Sync Session - pass (A9-1-1-1-1) StartSession - "0xFF" before a call token(0xF8); SyncSession: pass (A9-1-1-2-1) StartSession - "0xFF" before a call token(0xF8); SyncSession: pass (A9-1-1-1-1) StartSession - "0xFF" between a call token and an 'Invoking[D'; SyncSession: pass (A9-1-1-5-1) StartSession - "0xFF" between a 'MethodID' and 'F0'; SyncSession: pass (A9-1-1-5-1) StartSession - "0xFF" between endList('F1') and endData('F9'); SyncSession: pass (A9-1-1-5-1) StartSession - "0xFF" between endList('F1') and endData('F9); SyncSession: pass (A9-1-1-1-1) StartSession - "0xFF" after statusCode list; ending; SyncSession: pass (A9-1-1-1-1) StartTession - "0xFF" between endList('F1') and endData('F9); SyncSession: pass (A9-1-1-1-1) StartTession - "0xFF" between a Transact	PASS PASS PASS PASS PASS PASS PASS PASS
(A8-1-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-1-1-1) EndSession Effect: TPar sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-1) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession: Effect: Verify the session is aborted after TPer sends a CloseSession (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 (A8-3-4-1-1) Session Timeout: If session# = MaxSessions and a session is timeout; Start/Sync Session - pass (A9-1-1-1-1) StartSession - '0xFF' before a call token(0xF8); SyncSession: pass (A9-1-1-1-1) StartSession - '0xFF' between a call token and an 'Invoking(0)'; SyncSession: pass (A9-1-1-4-1) StartSession - '0xFF' between a 'Method(0' and 'F0; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' among SPUID and Write paramters; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' between endList(F1) and endData(F9); SyncSession: pass (A9-1-1-6-1) StartSession - '0xFF' between endData(F9) and adatatacCode(F0); SyncSession: pass (A9-1-1-1-1) StartTsession - '0xFF' between endData(F9) and startSecsion - pass (A9-1-1-1-1) StartTsession - '0xFF' between a TransactionEnd token; Response: pass (A9-1-1-1-1) StartTransaction - '0xFF' between a TransactionE	PASS PASS PASS PASS PASS PASS PASS PASS
(A7-1-8-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + Methodinvoke and EndTrans; Session shall be closed (A8-1-1-1-1(3)) EndSession Grammar: EndSession is encoded outside of a method invocation in a control session; End token shall be closed (A8-1-2-1-1) EndSession Effect: Ter sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (A8-2-3-1-1) Session after CloseSession: Start a new session shall pass after the Session is aborted (A8-3-1-1) Session Timeout: If sessionie = MaxSessions and a session is timeout; Start/Sync Session - pass (A8-3-4-1-1) Session Timeout: If sessionie = MaxSessions and a session is timeout; Start/Sync Session - pass (A9-1-1-1-1) StartSession - "0xFF" before a call token(0xF8); SyncSession: pass (A9-1-1-2-1) StartSession - "0xFF" before a call token(0xF8); SyncSession: pass (A9-1-1-1-1) StartSession - "0xFF" between a call token and an 'Invoking[D'; SyncSession: pass (A9-1-1-5-1) StartSession - "0xFF" between a 'MethodID' and 'F0'; SyncSession: pass (A9-1-1-5-1) StartSession - "0xFF" between endList('F1') and endData('F9'); SyncSession: pass (A9-1-1-5-1) StartSession - "0xFF" between endList('F1') and endData('F9); SyncSession: pass (A9-1-1-1-1) StartSession - "0xFF" after statusCode list; ending; SyncSession: pass (A9-1-1-1-1) StartTession - "0xFF" between endList('F1') and endData('F9); SyncSession: pass (A9-1-1-1-1) StartTession - "0xFF" between a Transact	PASS PASS PASS PASS PASS PASS PASS PASS
(A8-1-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-1-1-1) EndSession Effect: TPar sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-1) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession: Effect: Verify the session is aborted after TPer sends a CloseSession (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 (A8-3-4-1-1) Session Timeout: If session# = MaxSessions and a session is timeout; Start/Sync Session - pass (A9-1-1-1-1) StartSession - '0xFF' before a call token(0xF8); SyncSession: pass (A9-1-1-1-1) StartSession - '0xFF' between a call token and an 'Invoking(0)'; SyncSession: pass (A9-1-1-4-1) StartSession - '0xFF' between a 'Method(0' and 'F0; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' among SPUID and Write paramters; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' between endList(F1) and endData(F9); SyncSession: pass (A9-1-1-6-1) StartSession - '0xFF' between endData(F9) and adatatacCode(F0); SyncSession: pass (A9-1-1-1-1) StartTsession - '0xFF' between endData(F9) and startSecsion - pass (A9-1-1-1-1) StartTsession - '0xFF' between a TransactionEnd token; Response: pass (A9-1-1-1-1) StartTransaction - '0xFF' between a TransactionE	PASS PASS PASS PASS PASS PASS PASS PASS
(A8-1-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1) EndSession Grammar: End Session - '0xFA' returned (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (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 (A8-1-1-1-1) EndSession Effect: TPar sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-1) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-10-1) CloseSession: Effect: Verify the session is aborted after TPer sends a CloseSession (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 (A8-3-4-1-1) Session Timeout: If session# = MaxSessions and a session is timeout; Start/Sync Session - pass (A9-1-1-1-1) StartSession - '0xFF' before a call token(0xF8); SyncSession: pass (A9-1-1-1-1) StartSession - '0xFF' between a call token and an 'Invoking(0)'; SyncSession: pass (A9-1-1-4-1) StartSession - '0xFF' between a 'Method(0' and 'F0; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' among SPUID and Write paramters; SyncSession: pass (A9-1-1-5-1) StartSession - '0xFF' between endList(F1) and endData(F9); SyncSession: pass (A9-1-1-6-1) StartSession - '0xFF' between endData(F9) and adatatacCode(F0); SyncSession: pass (A9-1-1-1-1) StartTsession - '0xFF' between endData(F9) and startSecsion - pass (A9-1-1-1-1) StartTsession - '0xFF' between a TransactionEnd token; Response: pass (A9-1-1-1-1) StartTransaction - '0xFF' between a TransactionE	PASS PASS PASS PASS PASS PASS PASS PASS
(A8-1-1-11) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-12)) EndSession Grammar: End Session is conceded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + Methodilrvoke and EndTrans; Session shall be closed (A8-1-1-1-1(2)) EndSession Grammar: End Session is encoded outside of a method invocation in a control session; End token shall be closed (A8-1-1-1-1(3)) EndSession Effect: Ther sends an End of Session token in Regular session; Session shall be closed (A8-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (A8-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-2-2-1-2) CloseSession Effect: Verify the session is aborted after Ther sends a CloseSession (A8-2-3-1-1) Session Timeout: If sessioni = MaxSessions and a session is timeout; Start/Sync Session - pass (A8-3-4-1-1) Session Timeout: If sessioni = MaxSession after a session aborted due to the timeout - pass A9: Check Empty Atom (A9-1-1-1-1) StartSession - "0xFF" between a call token(0xF8); SyncSession: pass (A9-1-1-2-1) StartSession - "0xFF" between a call token and an 'InvokingID'; SyncSession: pass (A9-1-1-5-1) StartSession - "0xFF" between a "MethodID" and "MethodID'; SyncSession: pass (A9-1-1-5-1) StartSession - "0xFF between an 'MethodID' and Tot'; SyncSession: pass (A9-1-1-1-1) StartSession - "0xFF between and SPUID parameters; SyncSession: pass (A9-1-1-1-1) StartSession - "0xFF among HostSID and SPUID parameters; SyncSession: pass (A9-1-1-1-1) StartSession - "0xFF between endList(F1') and endData(F9') syncSession: pass (A9-1-1-1-1) StartSession - "0xFF between endList(F1') and endData(F9'); SyncSession: pass (A9-1-1-1-1) StartSession - "0xFF between endList(F1') and endData(F9'); SyncSession: pass (A9-1-1-1-1) StartSession - "0xFF between a TransactionStart token and the status code; Response: pass (PASS PASS PASS PASS PASS PASS PASS PASS
(AA-1-8-1-1) Trans+Session Close: Transaction is aborted after power cycle AB: Test Ending Session (AB-1-1-1-1] EndSession Grammar: End Session in OxFA' returned (AB-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans and EndTrans; Session shall be closed (AB-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans + Methodinvoke and EndTrans; Session shall be closed (AB-1-1-1-1(2)) EndSession Grammar: End Session is encoded outside of a method invocation in a control session; End token shall be closed (AB-1-1-1-1) EndSession Effect: Ther sends an End of Session token in Regular session; Session shall be closed (AB-1-2-1-2) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (AB-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (AB-2-2-1-1) CloseSession Effect: Verify the session is aborted after TPer sends a CloseSession (AB-2-1-1) Session Timeout: If sessions = MaxSessions and a session is timeout; Start/Sync Session - pass (AB-3-1-1-1) Session Timeout: If sessions = AmxSessions and a session is timeout; Start/Sync Session - pass (AB-1-1-1-1) StartSession - "OxFF" between a call token (OxFB); SyncSession: pass (AB-1-1-1) StartSession - "OxFF" between a call token and an "InvokingID"; SyncSession: pass (AB-1-1-1-1) StartSession - OxFF between a "HerhotolD" and Fry SyncSession: pass (AB-1-1-1-1) StartSession - OxFF between a "HerhotolD" and Fry SyncSession: pass (AB-1-1-1-1) StartSession - OxFF between a "HerhotolD" and Fry SyncSession: pass (AB-1-1-1-1) StartSession - OxFF between an "HerhotolD" and Fry SyncSession: pass (AB-1-1-1-1) StartSession - OxFF between an "HerhotolD" and Fry SyncSession: pass (AB-1-1-1-1) StartSession - OxFF between an Endotal (FP) and status Code (FP); SyncSession: pass (AB-1-1-1-1) StartSession - OxFF between an Endotal (FP) and status Code (FP); SyncSession: pass (AB-1-1-1-1) StartTransaction - OxFF between an Endotal (FP) and status Code (FP);	PASS PASS PASS PASS PASS PASS PASS PASS
(AA-1-1-1) EndSession Grammar: End Session - "OxFA" returned (AB-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (AB-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans - Methodinvoke and EndTrans; Session shall be closed (AB-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans - Methodinvoke and EndTrans; Session shall be closed (AB-1-1-1-1(2)) EndSession Grammar: End Session is encoded outside of a method invocation in a control session; End token shall be closed (AB-1-1-1) EndSession ErdEnt: Terr sends an End of Session token in Regular session; Session shall be closed (AB-1-2-1) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (AB-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (AB-2-2-1-1) Session Timeout: If sessioni = MaxSessions and a session is timeout; Start/Sync Session - pass (AB-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (AB-1-1-1) StartSession - "OxFF" before a call token(0xF8); SyncSession: pass (AB-1-1-1) StartSession - "OxFF" between a call token and an "Invoking[D"; SyncSession: pass (AB-1-1-1) StartSession - "OxFF" between a "MethodID" and "F0"; SyncSession: pass (AB-1-1-1) StartSession - "OxFF" between a "MethodID" and "F0"; SyncSession: pass (AB-1-1-1) StartSession - "OxFF" between and "MethodID"; SyncSession: pass (AB-1-1-1) StartSession - "OxFF" among SPUID and SPUID parameters; SyncSession: pass (AB-1-1-1) StartSession - "OxFF" among SPUID and SPUID parameters; SyncSession: pass (AB-1-1-1) StartSession - "OxFF" among SPUID and SPUID parameters; SyncSession: pass (AB-1-1-1) StartTransaction - "OxFF" among SPUID and SPUID parameters; SyncSession: pass (AB-1-1-1) StartTransaction - "OxFF" after a TransactionStart token; Response: pass (AB-1-1-1) StartTransaction - "OxFF" after a TransactionStart token; Response: p	PASS PASS PASS PASS PASS PASS PASS PASS
(A7-1-8-1-1) Trans+Session Close: Transaction is aborted after power cycle A8: Test Ending Session (A8-1-1-1-1) EndSession Grammar: End Session · OxFA returned (A8-1-1-1-12) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (A8-1-1-1-12) EndSession Grammar: End Session is encoded within StartTrans + MethodInvoke and EndTrans; Session shall be closed (A8-1-1-1-12) EndSession Grammar: End Session is encoded outside of a method invocation in a control session; End token shall be closed (A8-1-1-1-13) EndSession End: Ther sends an End of Session token in Regular session; Session shall be closed (A8-1-2-12) EndSession End End: End Session Request with some tokens which follow the End of Session (Red Session) End (A8-1-1-1) Session after EndSession: Start a new session shall pass after the Session closed (A8-1-1-1) Session after CloseSession: Start a new session shall pass after the Session is aborted (A8-2-1-1) Session Timeout: Start/Sync Session after the Session is aborted (A8-3-2-1) Session Timeout: Start/Sync Session after a session is aborted due to the timeout - pass A8: Check Empty Atom (A9-1-1-1) StartSession · OxFF between a call token(0xF8); SyncSession: pass (A9-1-1-1) StartSession · OxFF between a call token and an 'InvokingID'; SyncSession: pass (A9-1-1-1) StartSession · OxFF between an 'InvokingID' and 'Pri'; SyncSession: pass (A9-1-1-1) StartSession · OxFF between an 'InvokingID' and Pri't parameters; SyncSession: pass (A9-1-1-1) StartSession · OxFF between and 'InvokingID' and endData(F9'); SyncSession: pass (A9-1-1-1) StartSession · OxFF between end List(F1) and endData(F9'); SyncSession: pass (A9-1-1-1) StartSession · OxFF between end Ends(F9) and start (SyncSession: pass (A9-1-1-1) StartSession · OxFF between end Ends(F9) and start (SyncSession: pass (A9-1-1-1) StartSession · OxFF between end Ends(F9) and start (SyncSession: pass (A9-1-1-1) StartTransaction · OxFF between and TransactionEnd token; Response: pass (A9-1-1-1-1) StartTransaction · OxFF	PASS PASS PASS PASS PASS PASS PASS PASS
(AA-1-1-1) EndSession Grammar: End Session - "OxFA" returned (AB-1-1-1-1) EndSession Grammar: End Session is encoded within StartTrans and EndTrans; Session shall be closed (AB-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans - Methodinvoke and EndTrans; Session shall be closed (AB-1-1-1-1(2)) EndSession Grammar: EndSession is encoded within StartTrans - Methodinvoke and EndTrans; Session shall be closed (AB-1-1-1-1(2)) EndSession Grammar: End Session is encoded outside of a method invocation in a control session; End token shall be closed (AB-1-1-1) EndSession ErdEnt: Terr sends an End of Session token in Regular session; Session shall be closed (AB-1-2-1) EndSession Effect: EndSession Request with some tokens which follow the End of Session; EndSession Response - pass (AB-1-4-1-1) Session after EndSession: Start a new session shall pass after the Session closed (AB-2-2-1-1) Session Timeout: If sessioni = MaxSessions and a session is timeout; Start/Sync Session - pass (AB-3-4-1-1) Session Timeout: Start/Sync Session after a session aborted due to the timeout - pass A9: Check Empty Atom (AB-1-1-1) StartSession - "OxFF" before a call token(0xF8); SyncSession: pass (AB-1-1-1) StartSession - "OxFF" between a call token and an "Invoking[D"; SyncSession: pass (AB-1-1-1) StartSession - "OxFF" between a "MethodID" and "F0"; SyncSession: pass (AB-1-1-1) StartSession - "OxFF" between a "MethodID" and "F0"; SyncSession: pass (AB-1-1-1) StartSession - "OxFF" between and "MethodID"; SyncSession: pass (AB-1-1-1) StartSession - "OxFF" among SPUID and SPUID parameters; SyncSession: pass (AB-1-1-1) StartSession - "OxFF" among SPUID and SPUID parameters; SyncSession: pass (AB-1-1-1) StartSession - "OxFF" among SPUID and SPUID parameters; SyncSession: pass (AB-1-1-1) StartTransaction - "OxFF" among SPUID and SPUID parameters; SyncSession: pass (AB-1-1-1) StartTransaction - "OxFF" after a TransactionStart token; Response: pass (AB-1-1-1) StartTransaction - "OxFF" after a TransactionStart token; Response: p	PASS PASS PASS PASS PASS PASS PASS PASS

(A10-1-6-5-1) Optional Params: Check Host Properties - MaxPacketSize = 7ECh: Response value = 7ECh	PASS
(A10-1-6-6-1) Optional Params: Check Host Properties - MaxIndTokenSize < 7C8h: Response value = 7C8h	PASS
(A10-1-6-6-1) Optional Params: Check Host Properties - MaxIndTokenSize = 7C8h: Response value = 7C8h	PASS
(A10-1-6-7-1) Optional Params: Check Host Properties - MaxPackets = a number: Response value <= a number	PASS
(A10-16-8-1) 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 <= a number	PASS
(A10-1-6-15-1) Optional Params: Check Host Properties - Omission of HostParams: no HostParams returned	PASS
A10: Properties response and effect test	PASS
(A10-3-1-1-2) Properties Effect - HostProp: TPer's response would contain data > MaxComPacketSize; Response: StatusCode = 11h	PASS
(A10-3-1-3-2) Properties Effect - HostProp: TPer's response would contain data > MaxPacketSize; Response: StatusCode = 11h	PASS
(A10-3-1-4-2) Properties Effect - HostProp: TPer's response would contain data token > MaxIndTokenSize; Response: Session abort	PASS
Properties Effect - TPerProp in regular session: = TPer's MaxComPacketSize; Response: Pass	PASS
(A10-3-2-1-1) Properties Effect - TPerProp in regular session: > TPer's MaxComPacketSize; Response: ST = 51h at ATA interface level	PASS
(A10-3-2-3-1) Properties Effect - TPerProp in regular session: > TPer's MaxPacketSize; Response: Session abort	N/A
(A10-3-2-4-1) Properties Effect - TPerProp in regular session: > TPer's MaxIndTokenSize; Response: Session abort	N/A
Properties Effect - TPerProp in control session: = TPer's MaxComPackets; Response: Pass	PASS
(A10-3-2-1-1) Properties Effect - TPerProp in control session: > TPer's MaxComPacketSize; Response: ST = 51h at ATA interface level	PASS
(A10-3-2-3-1) Properties Effect - TPerProp in control session: > TPer's MaxPackets; Response: Discarded by TPer	N/A
(A10-3-2-4-1) Properties Effect - TPerProp in control session: > TPer's MaxIndTokenSize; Response: Discarded by TPer	N/A
	PASS
(A10-3-2-6-1) Properties Effect - TPerProp in control session: > TPer's MaxSubPackets; Response: Discarded by TPer	
(A10-3-2-15-1) Properties Effect - TPerProp: MaxAuthentications shall not be 1	PASS
A11: Test Start/SyncSession()	PASS
(A11-1-1-1) StartSession - SessionID: not all 0; SyncSession - Status Code: 01h (Not Authorized)	PASS
(A11-3-2-1-1) StartSession - HostSessionID: 4-byte uinteger(-0FFFFFFFh); SyncSession - Pass and Tries = 0 in C PIN table	PASS
(A11-3-2-1-1) StartSession - HostSessionID: 4-byte uinteger(=0FFFFFFFh); SyncSession - Pass and Tries = 0 in C_PIN table	PASS
(A11-3-2-1-3) StartSession - HostSessionID: > 4-byte; SyncSession - Status Code: no data returned	PASS
(A11-3-2-2-2) StartSession - SPUID: nonexistent in the SP table; SyncSession - Status Code: 0Ch (Invalid_Param)	PASS
(A11-3-2-2-3) StartSession - SPUID: LockingSP in manufactured-inactive; SyncSession - Status Code: 0Ch (Invalid_Param)	PASS
(A11-3-2-3-3(2)) StartSession - Write: 1; SyncSession - Pass and Tries = 0 in C_PIN table	PASS
(A11-3-2-3-4) StartSession - Write: 2; SyncSession - Status Code: 0Ch (Invalid_Param)	PASS
A11: Test Start/SyncSession with Optional Parameters	PASS
	PASS
(A11-3-4-1-5) StartSession - HostChallenge: correct credential; SyncSession - Pass	
(A11-3-4-1-6) StartSession - HostChallenge: correct credential(if Tries=TryLimit); SyncSession - Status Code: 01h or 12h (Not_Authorized/Authority_locked_oi	
(A11-3-4-1-7) StartSession - HostChallenge: incorrect credential; SyncSession - Status Code: 01h (Not_Authorized)	PASS
(A11-3-4-1-10) StartSession - HostChallenge: anybody (explicitly in HostSignAuth); SyncSession - Pass	PASS
(A11-3-4-1-11) StartSession - HostChallenge: omitted (any authority); SyncSession - Status Code: 0Ch (Invalid_Param)	PASS
(A11-3-4-2-6) StartSession - HostSignAuth: nonexistent UID; SyncSession - Status Code: 0Ch (Invalid_Param.)	PASS
(A11-3-4-2-6(2)) StartSession - HostSignAuth: disabled authority's UID; SyncSession - Status Code: 01h (Not_Authorized)	PASS
(144.0.4.0.0(0)) 01.10(1.11.10(1.4.11	PASS
(A11-3-4-2-6(3)) StartSession - HostSignAuth: a class authority UID; SyncSession - Status Code: 0Ch (Invalid_Param)	
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - pass	PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - pass (A11-3-5-6-1-1) StartSession - exceed MaxSessions property; SyncSession - Status Code: 03h or 07h (SP_Busy/No_Sessions_Available)	PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - pass	PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - pass (A11-3-5-6-1-1) StartSession - exceed MaxSessions property; SyncSession - Status Code: 03h or 07h (SP_Busy/No_Sessions_Available)	PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP	PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP	PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP	PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP - Response	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP	PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP - Response	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Response	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for InvokingID; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - 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 PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for InvokingID; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - 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 PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-1-1-1-1(1)) Get Request - with long atom for MethodID; Response - Pass (A6-1-1-1-1(1)) Get Request - with long atom for MethodID; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for InvokingID; Response - 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 MethodID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-1-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-1-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-1-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-1-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-1-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-1-1-1-1) Get Request - with long atom for MethodID; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-1-1-1(1)) Get Request - with none method Me	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - 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 (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with modium atom for MethodID; Response - Pass (A6-1-1-1) Get Request - with invokingID; Response - Pass (A6-1-1-1) Get Request - with invokingID; Response - Pass (A6-1-1-1) Get Request - with invokingID; Response - Pass (A6-1-1-1) Get Request - with invokingID; Response - Pass (A6-1-1-2-1) Get Request - with invokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-1-1-1(1)) Get Request - with none method Me	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - 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 (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with modium atom for MethodID; Response - Pass (A6-1-1-1) Get Request - with invokingID; Response - Pass (A6-1-1-1) Get Request - with invokingID; Response - Pass (A6-1-1-1) Get Request - with invokingID; Response - Pass (A6-1-1-1) Get Request - with invokingID; Response - Pass (A6-1-1-2-1) Get Request - with invokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Response Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for invokingID; Response - 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 MethodID; Response - Pass (A6-0-1-1-1) Get Request - with involid token for 'Call'(F8h); Response - Session Abort (A6-1-1-1-1(1)) Get Request - with nonexistent invokingID; Response - Session Abort (A6-1-1-3-1(2)) Get Request - with nonexistent InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) Get Request - with none-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with none-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with none-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with none-byte token for MethodID; Response - Status Code: 01h(Not_Authorized)	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Response Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with nong atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with monexistent InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Response Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with needium atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with needium atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with none atom for MethodID; Response - Pass (A6-1-1-1) Get Request - with nonexistent InvokingID; Response - Session Abort (A6-1-1-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Response Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with nong atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with monexistent InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Response Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with needium atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with needium atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with none atom for MethodID; Response - Pass (A6-1-1-1) Get Request - with nonexistent InvokingID; Response - Session Abort (A6-1-1-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP - Response Get - LifeCycle(LockingSP) - Response Get - LifeCycle(Locking SP) - Request Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with inon atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with inon atom for MethodID; Response - Pass (A6-1-1-1) Get Request - with inonsystent InvokingID; Response - Session Abort (A6-1-1-2-1) Get Request - with nonexistent InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) Get Request - with nonexistent InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-2-1) Get Request - with nonexistent InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with nonexistent MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with nonexistent MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with nonexistent MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with nonexistent MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with nonexistent MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with nonexistent MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Response Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Request End Session - Request End Session - Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with inong atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with inong atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with inong atom for MethodID; Response - Pass (A6-1-1-1-1) Get Request - with inonabite nor MethodID; Response - Pass (A6-1-1-1-1) Get Request - with inonabite nor MethodID; Response - Pass (A6-1-1-2-1) Get Request - with inonabite nor MethodID; Response - Session Abort (A6-1-1-3-1(2)) Get Request - with nonexistent InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-3-1(2)) Get Request - with non-Byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-Byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Reques	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP Activate_LockingSP, Response Get - LifeCycle(Locking SP) - Response Cet - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with ung atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with midlum atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with midlum atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with invalid token for 'Call('F8h); Response - Session Abort (A6-1-1-1-1) Get Request - with nonexistent invokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) Get Request - with nonexistent invokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1(2)) Get Request - with nonexistent invokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with none-Jong token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with none-Jong token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with none-Sten methodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Request - with none-Jong token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1) Get Request - with none-Jong token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1) Get Request - with none-Jong token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1) Get Reques	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Response Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Request End Session - Request End Session - Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with inong atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with inong atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with inong atom for MethodID; Response - Pass (A6-1-1-1-1) Get Request - with inonabite nor MethodID; Response - Pass (A6-1-1-1-1) Get Request - with inonabite nor MethodID; Response - Pass (A6-1-1-2-1) Get Request - with inonabite nor MethodID; Response - Session Abort (A6-1-1-3-1(2)) Get Request - with nonexistent InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-3-1(2)) Get Request - with non-Byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-Byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Reques	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP Activate_LockingSP, Response Get - LifeCycle(Locking SP) - Response Cet - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with ung atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with midlum atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with midlum atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with invalid token for 'Call('F8h); Response - Session Abort (A6-1-1-1-1) Get Request - with nonexistent invokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) Get Request - with nonexistent invokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1(2)) Get Request - with nonexistent invokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with none-Jong token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with none-Jong token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with none-Sten methodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Request - with none-Jong token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1) Get Request - with none-Jong token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1) Get Request - with none-Jong token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1) Get Reques	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Response Cet - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-11-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-11-1) Get Request - with medium atom for InvokingID; Response - Pass (A6-0-11-1) Get Request - with ing atom for InvokingID; Response - Pass (A6-0-11-1) Get Request - with long atom for MethodID; Response - Pass (A6-0-11-1) Get Request - with ing atom for MethodID; Response - Pass (A6-1-11-1) Get Request - with ing atom for MethodID; Response - Pass (A6-1-11-1) Get Request - with none-blong token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-13-12)) Get Request - with none-blong token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-12)) Get Request - with none-blong token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-12) Get Request - with none-blong token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-12) Get Request - with none-blong token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1) Get Request - with none-blong token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1) Get Request - with none-blong token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1) Get Request - with none-blong token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1) Get Request - with none-blong token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1) G	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Response Get - LifeCycle(Locking SP) - Response Check the state of LockingSP - Response Check the state of LockingSP - Response Check the state of LockingSP - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with invokingID; Response - Pass (A6-0-1-1-1) Get Request - with invokingID; Response - Pass (A6-1-1-1-1(1)) Get Request - with invokingID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-1(2)) Get Request - with non-8-long token for InvokingID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-1(2)) Get Request - with non-8-long token for InvokingID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-1(2)) Get Request - with non-8-long token for InvokingID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-1(2)) Get Request - with non-8-long token for InvokingID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-1(2)) Get Request - with non-8-long token for InvokingID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-1(2)) Get Request - with non-8-long token for InvokingID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-1) Get Request - with non-8-long token for MethodID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-1) Get Request - with non-8-long token for MethodID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-1) Get Request - with non-8-long tok	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-34-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - pass (A11-35-6-1-1) StartSession - exceed MaxSessions property; SyncSession - Status Code: 03h or 07h (SP_Busy/No_Sessions_Available) Activating the Locking SP Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Response Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with involation for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with involation for MethodID; Response - Pass (A6-1-1-1-1) Get Request - with involation for MethodID; Response - Pass (A6-1-1-1-1) Get Request - with involation for MethodID; Response - Pass (A6-1-1-1-1) Get Request - with none-stent invokingID; Response - Session Abort (A6-1-1-2-1) Get Request - with none-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Request - with none-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with none-byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Request - with none-byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Request - with none-byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Request - with none-byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Request - with none-byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get R	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCyclet[Locking SP) - Response Get - LifeCyclet[Locking SP) - Response Check the state of LockingSP - Response Check the state of LockingSP End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with invoking Into InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-1-1-1-1) Get Request - with inon gatom for MethodID; Response - Session Abort (A6-1-1-1-1) Get Request - with nonexistent InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) Get Request - with non-Byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1(2)) Get Request - with non-Byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1(2)) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1(2)) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1(2)) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1(2)) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1(2)) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized)	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-34-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - pass (A11-35-6-1-1) StartSession - exceed MaxSessions property; SyncSession - Status Code: 03h or 07h (SP_Busy/No_Sessions_Available) Activating the Locking SP Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Response Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with involation for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with involation for MethodID; Response - Pass (A6-1-1-1-1) Get Request - with involation for MethodID; Response - Pass (A6-1-1-1-1) Get Request - with involation for MethodID; Response - Pass (A6-1-1-1-1) Get Request - with none-stent invokingID; Response - Session Abort (A6-1-1-2-1) Get Request - with none-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Request - with none-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with none-byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Request - with none-byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Request - with none-byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Request - with none-byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get Request - with none-byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1) Get R	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with HostChallenge - AdminSP Sync Session - AdminSP Activate_LockingSP Activate_LockingSP Activate_LockingSP - Response Get - LifeCyclet[Locking SP) - Response Get - LifeCyclet[Locking SP) - Response Check the state of LockingSP - Response Check the state of LockingSP End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with invoking Into InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-0-1-1-1) Get Request - with long atom for MethodID; Response - Pass (A6-1-1-1-1) Get Request - with inon gatom for MethodID; Response - Session Abort (A6-1-1-1-1) Get Request - with nonexistent InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) Get Request - with non-Byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1(2)) Get Request - with non-Byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1(2)) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1(2)) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1(2)) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1(2)) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-3-1-1(2)) Get Request - with non-Byte token for MethodID; Response - Status Code: 01h(Not_Authorized)	PASS PASS PASS PASS PASS PASS PASS PASS
(A11-3-4-2-9) StartSession - HostChallenge and HostSignAuth omitted: correct credential; SyncSession - 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 Start Session with Host Challenge - AdminSP SyncSession - AdminSP Activate_LockingSP - Response Get - LifeCycle(Locking SP) - Response Get - LifeCycle(Locking SP) - Response Get - LifeCycle(Locking SP) - Response Check the state of LockingSP End Session - Request End Session - Request End Session - Response A6: Grammar Check on Method/InvokeUID in regular session (A6-0-1-1-1) Get Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with medium atom for InvokingID; Response - Pass (A6-0-1-1-1) Get Request - with invokingID; Response - Pass (A6-0-1-1-1) Get Request - with invokingID; Response - Pass (A6-0-1-1-1) Get Request - with invokingID; Response - Pass (A6-1-1-1) Get Request - with invokingID; Response - Pass (A6-1-1-1) Get Request - with non-byte token for invokingID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-1(2)) Get Request - with non-byte token for invokingID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-1(2)) Get Request - with non-byte token for invokingID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-1(2)) Get Request - with non-byte token for invokingID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-1(2)) Get Request - with non-byte token for invokingID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-1-1) Get Request - with non-byte token for invokingID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-1-1) Get Request - with non-byte token for MethodID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-1-1) Get Request - with non-byte token for MethodID; Response - Status Code: Oth(Not_Authorized) (A6-1-3-1-1) Get Request - with non-byte otken for MethodID; Response - Status Code: Oth(Not_Au	PASS PASS PASS PASS PASS PASS PASS PASS

(A6-1-8-6-1) Get Request - with 2nd Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) Get Request - with 2nd Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-8-6-1) Get Request - with 3rd Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) Get Request - with 3rd Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-9-2-1) Get Request - with invalid token type of StatusCode End: 0e0h; Response - Session Abort	PASS
(A6-1-4-2-1(1)) Get Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param)	PASS
(A6-0-1-1-1) Set Request - with short atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) Set Request - with medium atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) Set Request - with long atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) Set Request - with medium atom for MethodID; Response - Pass	PASS
(A6-0-1-1-1) Set Request - with long atom for MethodID; Response - Pass	PASS
(A6-1-1-1-1(1)) Set Request - with invalid token for 'Call' (F8h); Response - Session Abort	PASS
(A6-1-1-2-1) Set Request - with nonexistent Invoking ID; Response - Status Code: 01h(Not Authorized)	PASS
(A6-1-1-3-1(2)) Set Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not Authorized)	PASS
(A6-1-1-3-1(2)) Set Request - with non-8-long token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-2-2-1) Set Request - with nonexistent MethodID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-2-3-1(2)) Set Request - with non-byte token for MethodID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-2-3-1(2)) Set Request - with non-8-long token for MethodID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-3-1-1) Set Request - no ACE in the ACL; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-4-2-1) Set Request - with invalid token type of StartList: 0e0h; Response - Session Abort	PASS
(A6-1-5-2-1) Set Request - with invalid token type of EndList: 0e0h; Response - Session Abort	PASS
(A6-1-6-2-1) Set Request - with invalid token type of EndData: 0e0h; Response - Session Abort	PASS
(A6-1-7-2-1) Set Request - with invalid token type of StatusCode Start: 0e0h; Response - Session Abort	PASS
(A6-1-8-1-2) Set Request - with first Status token = 81h(short); Response - Pass	PASS
(A6-1-8-2-1) Set Request - with first Status Code != 0h(found in status code); Response - fail	PASS
(A6-1-8-2-1) Set Request - with first Status Code != 0h(not in the status code); Response - fail	PASS
(A6-1-8-3-2) Set Request - with second Status Code != 0h; Response - Normal	PASS
(A6-1-8-3-2) Set Request - with third Status Code != 0h; Response - Normal	PASS
(A6-1-8-6-1) Set Request - with 1st Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) Set Request - with 1st Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-8-6-1) Set Request - with 2nd Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) Set Request - with 2nd Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-8-6-1) Set Request - with 3rd Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) Set Request - with 3rd Status token = 91h(integer); Response - Session Abort	PASS
	PASS
(A6-1-9-2-1) Set Request - with invalid token type of StatusCode End: 0e0h; Response - Session Abort	
(A6-1-4-2-1(1)) Set Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param)	PASS
(A6-1-4-2-1(2)) Set Request - with the same optional parameter encoded twice; Response - Status Code: 0Ch(Invalid_Param)	PASS
(A6-1-4-2-1(3)) Set Request - with the descending order of optional parameter; Response - Status Code: 0Ch(Invalid_Param)	PASS
(A6-0-1-1-1) Next Request - with short atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) Next Request - with medium atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) Next Request - with long atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) Next Request - with medium atom for MethodID; Response - Pass	PASS
(A6-0-1-1-1) Next Request - with long atom for MethodID; Response - Pass	PASS
(A6-1-1-1-1(1)) Next Request - with invalid token for 'Calt' (F8h); Response - Session Abort	PASS
(A6-1-1-2-1) Next Request - with nonexistent InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-1-3-1(2)) Next Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-1-3-1(2)) Next Request - with non-8-long token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-2-2-1) Next Request - with nonexistent MethodID; Response - Status Code: 01h(Not_Authorized)	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 StartList. Geon; 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 Abort	PASS
(A6-1-8-1-2) Next Request - with first Status token = 81h(short); Response - Pass	PASS
(A6-1-8-2-1) Next Request - with first Status Code!= 0h(found in status code); Response - fail	PASS
(A6-1-8-2-1) Next Request - with first Status Code != 0h(not in the status code); Response - fail	PASS
(A6-1-8-3-2) Next Request - with second Status Code != 0h; Response - Normal	PASS
(A6-1-8-3-2) Next Request - with third Status Code != 0h; Response - Normal	PASS
(A6-1-8-6-1) Next Request - with 1st Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) Next Request - with 1st Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-8-6-1) Next Request - with 2nd Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) Next Request - with 2nd Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-8-6-1) Next Request - with 3rd Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) Next Request - with 3rd Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-9-2-1) Next Request - with invalid token type of StatusCode End: 0e0h; Response - Session Abort	PASS
(A6-1-4-2-1(1)) Next Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param)	
(A6-1-4-2-1(2)) Next Request - with the same optional parameter encoded twice; Response - Status Code: 0Ch(Invalid_Param)	PASS PASS
	PASS PASS
(A6-1-4-2-1(3)) Next Request - with the descending order of optional parameter; Response - Status Code: OCh(Invalid_Param)	PASS PASS PASS
(A6-1-4-2-1(3)) Next Request - with the descending order of optional parameter; Response - Status Code: OCh(Invalid_Param) (A6-0-1-1-1) GetACL Request - with short atom for InvokingID; Response - Pass	PASS PASS PASS PASS
(A6-1-4-2-1(3)) Next Request - with the descending order of optional parameter; Response - Status Code: OCh(Invalid_Param) (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 PASS PASS PASS
(A6-1-4-2-1(3)) Next Request - with the descending order of optional parameter; Response - Status Code: OCh(Invalid_Param) (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 (A6-0-1-1-1) GetACL Request - with long atom for InvokingID; Response - Pass	PASS PASS PASS PASS PASS PASS
(A6-1-4-2-1(3)) Next Request - with the descending order of optional parameter; Response - Status Code: OCh(Invalid_Param) (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 (A6-0-1-1-1) GetACL Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) GetACL Request - with medium atom for MethodID; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-4-2-1(3)) Next Request - with the descending order of optional parameter; Response - Status Code: OCh(Invalid_Param) (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 (A6-0-1-1-1) GetACL Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) GetACL Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) GetACL Request - with long atom for MethodID; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-4-2-1(3)) Next Request - with the descending order of optional parameter; Response - Status Code: OCh(Invalid_Param) (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 (A6-0-1-1-1) GetACL Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) GetACL Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) GetACL Request - with long atom for MethodID; Response - Pass (A6-0-1-1-1) GetACL Request - with long atom for MethodID; Response - Pass (A6-1-1-1-1) GetACL Request - with invalid token for 'Call' (F8h); Response - Session Abort	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-4-2-1(3)) Next Request - with the descending order of optional parameter; Response - Status Code: OCh(Invalid_Param) (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 (A6-0-1-1-1) GetACL Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) GetACL Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) GetACL Request - with long atom for MethodID; Response - Pass (A6-1-1-1-1) GetACL Request - with invalid token for 'Call'(F8h); Response - Session Abort (A6-1-1-2-1) GetACL Request - with invalid token for 'Call'(F8h); Response - Status Code: 01h(Not_Authorized)	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-4-2-(3)) Next Request - with the descending order of optional parameter; Response - Status Code: OCh(Invalid_Param) (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 (A6-0-1-1-1) GetACL Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) GetACL Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) GetACL Request - with long atom for MethodID; Response - Pass (A6-0-1-1-1) GetACL Request - with invalid token for Call('[8h]; Response - Session Abort (A6-1-1-2-1) GetACL Request - with nonexistent InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) GetACL Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-4-2-1(3)) Next Request - with the descending order of optional parameter; Response - Status Code: OCh(Invalid_Param) (A6-0-1-1-1) GetACL Request - with short atom for InvokingID; Response - Pass (A6-0-1-1-1) GetACL Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) GetACL Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) GetACL Request - with long atom for MethodID; Response - Pass (A6-0-1-1-1) GetACL Request - with long atom for MethodID; Response - Pass (A6-0-1-1-1) GetACL Request - with invalid token for 'Call' (F8h); Response - Session Abort (A6-1-1-1-1-1) GetACL Request - with nonexistent InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) GetACL Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) GetACL Request - with non-8-long token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-4-2-1(3)) Next Request - with the descending order of optional parameter; Response - Status Code: 0Ch(Invalid_Param) (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 (A6-0-1-1-1) GetACL Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) GetACL Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) GetACL Request - with involation atom for MethodID; Response - Pass (A6-1-1-1-1) GetACL Request - with involation to MethodID; Response - Session Abort (A6-1-1-2-1) GetACL Request - with involation to MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) GetACL Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) GetACL Request - with non-B-long token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-2-1) GetACL Request - with non-B-long token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-4-2-1(3)) Next Request - with the descending order of optional parameter; Response - Status Code: OCh(Invalid_Param) (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 (A6-0-1-1-1) GetACL Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) GetACL Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) GetACL Request - with long atom for MethodID; Response - Pass (A6-0-1-1-1) GetACL Request - with invalid token for 'Call' (F8h); Response - Session Abort (A6-1-1-1-1(1)) GetACL Request - with nonexistent InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) GetACL Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) GetACL Request - with non-8-long token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1) GetACL Request - with non-8-long token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-3-1(2)) GetACL Request - with non-byte token for MethodID; Response - Status Code: 01h(Not_Authorized)	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-1-4-2-1(3)) Next Request - with the descending order of optional parameter; Response - Status Code: 0Ch(Invalid_Param) (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 (A6-0-1-1-1) GetACL Request - with long atom for InvokingID; Response - Pass (A6-0-1-1-1) GetACL Request - with medium atom for MethodID; Response - Pass (A6-0-1-1-1) GetACL Request - with involation atom for MethodID; Response - Pass (A6-1-1-1-1) GetACL Request - with involation to MethodID; Response - Session Abort (A6-1-1-2-1) GetACL Request - with involation to MethodID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) GetACL Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-1-3-1(2)) GetACL Request - with non-B-long token for InvokingID; Response - Status Code: 01h(Not_Authorized) (A6-1-2-2-1) GetACL Request - with non-B-long token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS PASS PASS PASS PASS PASS 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 - Session Abort	PASS
(A6-1-6-2-1) GetACL Request - with invalid token type of EndData: 0e0h; Response - Session Abort	PASS
(A6-1-7-2-1) GetACL Request - with invalid token type of StatusCode Start: 0e0h; Response - Session Abort	PASS
(A6-1-8-1-2) GetACL Request - with first Status token = 81h(short); Response - Pass	PASS
(A6-1-8-2-1) GetACL Request - with first Status Code != 0h(found in status code); Response - fail	PASS
(A6-1-8-2-1) GetACL Request - with first Status Code != 0h(not in the status code); Response - fail	PASS
(A6-1-8-3-2) GetACL Request - with second Status Code != 0h; Response - Normal	PASS
(A6-1-8-3-2) GetACL Request - with third Status Code != 0h; Response - Normal	PASS
(A6-1-8-6-1) GetACL Request - with 1st Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) GetACL Request - with 1st Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-8-6-1) GetACL Request - with 2nd Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) GetACL Request - with 2nd Status token = 91h(integer); Response - Session Abort	PASS
	PASS
(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 = 91h(integer); Response - Session Abort	PASS
(A6-1-9-2-1) GetACL Request - with invalid token type of StatusCode End: 0e0h; Response - Session Abort	PASS
(A6-1-4-2-1(1)) GetACL Request - with unexpected token encoded inside the Params; Response - Status Code: 0Ch(Invalid_Param)	PASS
(A6-0-1-1-1) GenKey Request - with short atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) GenKey Request - with medium atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) GenKey Request - with long atom for InvokingID; Response - Pass	PASS
(A6-0-1-1-1) GenKey Request - with medium atom for MethodID; Response - Pass	PASS
(A6-0-1-1-1) GenKey Request - with long atom for MethodID; Response - Pass	PASS
(A6-1-1-1-1(1)) GenKey Request - with invalid token for 'Call'(F8h); Response - Session Abort	PASS
(A6-1-1-2-1) GenKey Request - with nonexistent InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-1-3-1(2)) GenKey Request - with non-byte token for InvokingID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-1-3-1(2)) GenKey Request - with non-8-long token for InvokingID; Response - Status Code: 01h(Not Authorized)	PASS
(A6-1-2-2-1) GenKey Request - with nonexistent MethodID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-2-3-1(2)) GenKey Request - with non-byte token for MethodID: Response - Status Code: 01h(Not Authorized)	PASS
(A6-1-2-3-1(2)) GenKey Request - with non-8-long token for MethodID; Response - Status Code: 01h(Not_Authorized)	PASS
(A6-1-3-1-1) Genkey Request - no ACE in the ACL; Response - Status Code: 01h(Not, Authorized)	PASS
(A6-1-3-1-1/Qenkey Request - nonexistent InvokingID/MethodID in ACL; Response - Status Code: 01h(Not_Authorized)	PASS
	PASS
(A6-1-4-2-1) GenKey Request - with invalid token type of StartList: 0e0h; Response - Session Abort	
(A6-1-5-2-1) GenKey Request - with invalid token type of EndList: 0e0h; Response - Session Abort	PASS
(A6-1-6-2-1) GenKey Request - with invalid token type of EndData: 0e0h; Response - Session Abort	PASS
(A6-1-7-2-1) GenKey Request - with invalid token type of StatusCode Start: 0e0h; Response - Session Abort	PASS
(A6-1-8-1-2) GenKey Request - with first Status token = 81h(short); Response - Pass	PASS
(A6-1-8-2-1) GenKey Request - with first Status Code != 0h(found in status code); Response - fail	PASS
(A6-1-8-2-1) GenKey Request - with first Status Code != 0h(not in the status code); Response - fail	PASS
(A6-1-8-3-2) GenKey Request - with second Status Code != 0h; Response - Normal	PASS
(A6-1-8-3-2) GenKey Request - with third Status Code != 0h; Response - Normal	PASS
(A6-1-8-6-1) GenKey Request - with 1st Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) GenKey Request - with 1st Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-8-6-1) GenKey Request - with 2nd Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) GenKey Request - with 2nd Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-8-6-1) GenKey Request - with 3rd Status token = A1h(byte); Response - Session Abort	PASS
(A6-1-8-6-1) GenKey Request - with 3rd Status token = 91h(integer); Response - Session Abort	PASS
(A6-1-9-2-1) GenKey Request - with invalid token type of StatusCode End: 0e0h; Response - Session Abort	PASS
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
A6: Grammar check on Method/Invokel IID in control session	PASS
A6: Grammar check on Method/InvokeUID in control session (A6:3-1-2-1) Request - with invalid InvokingID: Response - no response prepared	PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared	PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared	PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response prepared	PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-2-1) Request - with nonexistent MethodID; Response - no response prepared	PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-2-1) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared	PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-2-1) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared	PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-2-1) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: integer) in MethodID; Response - no response prepared	PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-2-1) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F6: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-2-1) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-4-2-1) Request - with invalid token type of StartList: 0e0h; Response - no response prepared	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-2-1) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F6: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-2-1) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-4-2-1) Request - with invalid token type of StartList: 0e0h; Response - no response prepared	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-2-1) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: integer) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: uinteger) in MethodID; Response - no response prepared (A6-3-4-2-1) Request - with invalid token type of StartList: 0e0h; Response - no response prepared (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 PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-3-1(2)) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: integer) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-4-2-1) Request - with invalid token type of StartList: 0e0h; Response - no response prepared of StartList: 0e0h; Response - no response prepared or Status Code: 0Ch(invalid_param) (A6-3-4-2-1(1)) Request - without *F2' for the beginning of Mame-Value; Response - no response prepared or Status Code: 0Ch(invalid_param)	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: uinteger) in InvokingID; Response - no response prepared (A6-3-2-3-1(2)) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: integer) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-4-2-1) Request - with invalid token type of StartList: 0e0h; Response - no response prepared (A6-3-4-2-1(1)) Request - without 'F2' for the beginning of Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(1)) Request - with byte atom for value in Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(1)) Request - without 'F3' for the ending of Name-Value; Response - no response prepared or Status Code: OCh(invalid_param)	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-2-1) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F6: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: integer) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-4-2-1) Request - with invalid token type of StartList: OeOh; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(1)) Request - with byte atom for value in Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(1)) Request - without 'F3' for the ending of Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(1)) Request - Without 'F3' for the ending of Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(2)) Request - Without 'F3' for the ending of Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(2)) Request - Without 'F3' for the ending of Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(2)) Request - Without 'F3' for the ending of Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(2)) Request - Without 'F3' for the ending of Name-Value; Response - no response prepared or Status Code: OCh(invalid_param)	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-3-1(2)) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-4-2-1) Request - with invalid token type of StartList: 0e0h; Response - no response prepared or Status Code: 0Ch(invalid_param) (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) (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) (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) (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) (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) (A6-3-4-2-1(1)) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of EndList: 0e0h; Response - no response prepared	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - with xeypected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrIToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: integer) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-4-2-1) Request - with invalid token type of StartList: 0e0h; Response - no response prepared or Status Code: 0Ch(invalid_param) (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) (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) (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) (A6-3-4-2-1(2)) Request - with invalid token type of EndData: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of EndData: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of EndData: 0e0h; Response - no response prepared	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-3-1(2)) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - with xince (88: winteger) in InvokingID; Response - no response prepared (A6-3-2-3-1(2)) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CttTloken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: winteger) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: winteger) in MethodID; Response - no response prepared (A6-3-4-2-1) Request - with invalid token type of StartList: 0e0h; Response - no response prepared of Status Code: 0Ch(invalid_param) (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) (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) (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) (A6-3-4-2-1(2)) Request - with one type of EndData: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of StatusCode; Start: 0e0h; Response - no response prepared (A6-3-8-1-2) Request - with invalid token type of StatusCode Start: 0e0h; Response - no response prepared (A6-3-8-1-2) Request - with invalid token type of StatusCode Start: 0e0h; Response - no response prepared	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: integer) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-4-2-1) Request - with invalid token type of StartList: 0e0h; Response - no response prepared of Status Code: 0Ch(invalid_param) (A6-3-4-2-1(1)) Request - with bit for the beginning of Name-Value; Response - no response prepared or Status Code: 0Ch(invalid_param) (A6-3-4-2-1(1)) Request - with bit around for value in Name-Value; Response - no response prepared or Status Code: 0Ch(invalid_param) (A6-3-4-2-1(1)) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of StatusCode Start: 0e0h; Response - no response prepared (A6-3-8-1-2) Request - with first Status token = 81h(short); Response - poresponse - fail	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: integer) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: uinteger) in MethodID; Response - no response prepared (A6-3-4-2-1) Request - with invalid token type of StartList: 0e0h; Response - no response prepared or Status Code: 0Ch(invalid_param) (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) (A6-3-4-2-1(1)) Request - without 'F2' for the ending of Name-Value; Response - no response prepared or Status Code: 0Ch(invalid_param) (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) (A6-3-4-2-1(2)) Request - without response encoded twice; Response - Status Code: 0Ch(invalid_param) (A6-3-4-2-1(2)) Request - with invalid token type of EndData: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of StatusCode Start: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of StatusCode Start: 0e0h; Response - no response prepared (A6-3-8-2-1) Request - with first Status token = 8th(short); Response - no response prepared (A6-3-8-2-1) Request - with first Status Code! = 0h(found in status code); Response - fail (A6-3-8-3-2) Request - with first Status Code! = 0h(found in status code); Response - fail	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-3-1(2)) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F6: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: integer) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: uinteger) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with invalid token type of StartList: 0e0h; Response - no response prepared or Status Code: 0Ch(invalid_param) (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) (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) (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) (A6-3-4-2-1(2)) Request - without 'F3' for the ending of Name-Value; Response - no response prepared or Status Code: 0Ch(invalid_param) (A6-3-4-2-1(2)) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of EndData: 0e0h; Response - no response prepared (A6-3-8-3-1) Request - with first Status token = 81h(short); Response - no response prepared (A6-3-8-1-2) Request - with first Status Code != 0h; Response - Pass (A6-3-8-3-1) Request - with first Status Code != 0h; Response - Normal (A6-3-8-3-2) Request - with third Status Code != 0h; Response - Normal	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CttlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with invalid token type of StartList: 0e0h; Response - no response prepared (A6-3-4-2-1) Request - with out 'F2' for the beginning of Name-Value; Response - no response prepared or Status Code: 0Ch(invalid_param) (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) (A6-3-4-2-1(1)) Request - with out 'F3' for the ending of Name-Value; Response - no response prepared or Status Code: 0Ch(invalid_param) (A6-3-4-2-1(1)) Request - with invalid token type of EndData: 0e0h; Response - no response prepared (A6-3-4-2-1) Request - with invalid token type of EndData: 0e0h; Response - no response prepared (A6-3-4-2-1) Request - with invalid token type of StatusCode Start: 0e0h; Response - no response prepared (A6-3-8-1-2) Request - with first Status token = 81h(short); Response - no response prepared (A6-3-8-3-1) Request - with first Status Code! = 0h; Response - no response prepared (A6-3-8-3-3-2) Request - with first Status Code! = 0h; Response - no response prepared	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-3-1(2)) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CttTloken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with invalid token type of StartList: 0e0h; Response - no response prepared (A6-3-4-2-1) Request - with invalid token type of StartList: 0e0h; Response - no response prepared or Status Code: 0Ch(invalid_param) (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) (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) (A6-3-4-2-1(1)) Request - with or properties encoded twice; Response - no response prepared or Status Code: 0Ch(invalid_param) (A6-3-4-2-1) Request - with invalid token type of EndData: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of StatusCode; Response - no response prepared (A6-3-8-1-2) Request - with first Status Code != 0h(found in status Code); Response - no response prepared (A6-3-8-1-2) Request - with first Status Code != 0h; Response - Normal (A6-3-8-3-2) Request - with first Status Code != 0h; Response - Normal (A6-3-8-6-1) Request - with non-uinteger(byte) atom for 1st statusCode; Response - no response prepared (A6-3-8-6-1) Request - with non-uinteger(byte) atom for 1st statusCode;	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-4-2-1) Request - with invalid token type of StartList: 0e0h; Response - no response prepared (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) (A6-3-4-2-1(1)) Request - with binvalid token type of StartList: 0e0h; Response - no response prepared or Status Code: 0Ch(invalid_param) (A6-3-4-2-1(1)) Request - with orabid token type of EndList: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of Status: 0e0h; Response - no response prepared (A6-3-8-1-2) Request - with invalid token type of Status: 0e0h; Response - no response prepared (A6-3-8-1-2) Request - with invalid token type of Status: 0e0h; Response - no response prepared (A6-3-8-1-2) Request - with first Status Code != 0h; Response - no response prepared (A6-3-8-1-2) Request - with first Status Code != 0h; Response - Normal (A6-3-8-1-2) Request - with hird Status Code != 0h; Response - Normal (A6-3-8-6-1) Request - with non-uinteger(integer) atom for 2nd statusCode; Response - no response prepared (A6-3-8-6-1) Request - with non-uinteger	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: integer) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: uinteger) in MethodID; Response - no response prepared (A6-3-4-2-1) Request - with invalid token type of StartList: 0e0h; Response - no response prepared or Status Code: 0Ch(invalid_param) (A6-3-4-2-1(1)) Request - with bive atom for value in Name-Value; Response - no response prepared or Status Code: 0Ch(invalid_param) (A6-3-4-2-1(1)) Request - with out 'F2' for the ending of Name-Value; Response - no response prepared or Status Code: 0Ch(invalid_param) (A6-3-4-2-1(1)) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-4-2-1) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of StatusCode Start: 0e0h; Response - no response prepared (A6-3-8-3-1) Request - with first Status Code != 0h(found in status Code); Response - no response prepared (A6-3-8-3-2) Request - with first Status Code != 0h; Response - Normal (A6-3-8-3-2) Request - with first Status Code != 0h; Response - Normal (A6-3-8-6-1) Request - with non-uinteger(byte) atom for 1st statusCode; Response - no response prepared (A6-3-8-6-1) Request - with non-uinteger(integer) atom for 3rd statusCode; Response - no response prepared (A6-3-8-6-1) Request - with non-uinteger(integer) atom for 3rd statusCode; Response -	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-4-2-1) Request - with invalid token type of StartList: 0e0h; Response - no response prepared (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) (A6-3-4-2-1(1)) Request - with binvalid token type of StartList: 0e0h; Response - no response prepared or Status Code: 0Ch(invalid_param) (A6-3-4-2-1(1)) Request - with orabid token type of EndList: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of Status: 0e0h; Response - no response prepared (A6-3-8-1-2) Request - with invalid token type of Status: 0e0h; Response - no response prepared (A6-3-8-1-2) Request - with invalid token type of Status: 0e0h; Response - no response prepared (A6-3-8-1-2) Request - with first Status Code != 0h; Response - no response prepared (A6-3-8-1-2) Request - with first Status Code != 0h; Response - Normal (A6-3-8-1-2) Request - with hird Status Code != 0h; Response - Normal (A6-3-8-6-1) Request - with non-uinteger(integer) atom for 2nd statusCode; Response - no response prepared (A6-3-8-6-1) Request - with non-uinteger	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F8: cittToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CittToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: integer) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: integer) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with invalid token type of StartList: 0e0h; Response - no response prepared (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) (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) (A6-3-4-2-1(1)) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-4-2-1(2)) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-8-2-1) Request - with first Status token = 81h(short); Response - no response prepared (A6-3-8-2-1) Request - with first Status token = 81h(short); Response - no response prepared (A6-3-8-3-2) Request - with first Status Code != 0h; Response - Normal (A6-3-8-3-2-1) Request - with hird Status Code != 0h; Response - Normal (A6-3-8-3-2-1) Request - with non-uinteger(pyte) atom for 1st statusCode; Response - no response prepared (A6-3-8-6-1) Request - with non-uinteger(integer) atom for 3rd statusCode; Response - no response prepared (A6-3-8-6-1) Request - with non-uinteger(integer) atom for 3rd statusCode; Response - no response prepared (A6-3-8-6-1) Request - with no	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrIToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrIToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(F4: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: integer) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with invalid token type of StartList: 0e0h; Response - no response prepared (A6-3-2-3-1(2)) Request - with invalid token type of StartList: 0e0h; Response - no response prepared (A6-3-4-2-1(1)) Request - with brut 'F2' for the beginning of Name-Value; Response - no response prepared or Status Code: 0Ch(invalid_param) (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) (A6-3-4-2-1(1)) Request - with invalid token for value in Name-Value; Response - no response prepared or Status Code: 0Ch(invalid_param) (A6-3-4-2-1(1)) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-4-2-1(1)) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of StatusCode Start: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of StatusCode Start: 0e0h; Response - no response prepared (A6-3-8-1) Request - with first Status token = 81h(short); Response - pass (A6-3-8-2-1) Request - with first Status token = 81h(short); Response - no response prepared (A6-3-8-3-2) Request - with non-uinteger(integer) atom for 3rd statusCode; Response - no response prepared (A6-3-8-3-1) Request - with non-uinteger(integer) atom	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-3-1(2)) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F1: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: integer) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with invalid token type of StartList: 0e0h; Response - no response prepared (A6-3-4-2-1) Request - without F2' for the beginning of Name-Value; Response - no response prepared or Status Code: 0Ch(invalid_param) (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) (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) (A6-3-4-2-1(1)) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of EndData: 0e0h; Response - no response prepared (A6-3-8-1) Request - with first Status token = 81h(short); Response - pass (A6-3-8-2-1) Request - with first Status Code = 0h(found in status code); Response - no response prepared (A6-3-8-3-2) Request - with first Status Code = 0h; Response - no response prepared (A6-3-8-6-1) Request - with invalid token type of StatusCode Start: 0e0h; Response - no response prepared (A6-3-8-6-1) Request - with invalid token type of StatusCode; Response - no response prepared (A6-3-8-6-1) Request - with invalid token type of StatusCode; Response - no respon	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-3-1(2)) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: uniteger) in InvokingID; Response - no response prepared (A6-3-2-3-1(2)) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F3: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: integer) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - without F2' for the beginning of Name-Value; Response - no response prepared (A6-3-4-2-1(1)) Request - without F2' for the beginning of Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(1)) Request - without F3' for the ending of Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(1)) Request - without F3' for the ending of Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(1)) Request - without F3' for the ending of Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(1)) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-8-2-1) Request - with first Status Code = Balt(short); Response - no response prepared (A6-3-8-2-1) Request - with first Status Code = 10h; Response - no response prepared (A6-3-8-3-2) Request - with first Status Code = 10h; Response - no response prepared (A6-3-8-3-2) Request - with first Status Code = 10h; Response - Normal (A6-3-8-3-1) Request - with first Status Code = 10h; Response - Normal (A6-3-8-6-1) Request - with non-uinteger(integer) atom for 3rd statusCode; Response - no response prep	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-3-1(2)) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F1: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: integer) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with invalid token type of StartList: 0e0h; Response - no response prepared (A6-3-4-2-1) Request - without F2' for the beginning of Name-Value; Response - no response prepared or Status Code: 0Ch(invalid_param) (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) (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) (A6-3-4-2-1(1)) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of EndData: 0e0h; Response - no response prepared (A6-3-8-1) Request - with first Status token = 81h(short); Response - pass (A6-3-8-2-1) Request - with first Status Code = 0h(found in status code); Response - no response prepared (A6-3-8-3-2) Request - with first Status Code = 0h; Response - no response prepared (A6-3-8-6-1) Request - with invalid token type of StatusCode Start: 0e0h; Response - no response prepared (A6-3-8-6-1) Request - with invalid token type of StatusCode; Response - no response prepared (A6-3-8-6-1) Request - with invalid token type of StatusCode; Response - no respon	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-3-1(2)) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: uniteger) in InvokingID; Response - no response prepared (A6-3-2-3-1(2)) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrlToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F3: Reserved) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(98: integer) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - without F2' for the beginning of Name-Value; Response - no response prepared (A6-3-4-2-1(1)) Request - without F2' for the beginning of Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(1)) Request - without F3' for the ending of Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(1)) Request - without F3' for the ending of Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(1)) Request - without F3' for the ending of Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(1)) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-6-2-1) Request - with invalid token type of EndList: 0e0h; Response - no response prepared (A6-3-8-2-1) Request - with first Status Code = Balt(short); Response - no response prepared (A6-3-8-2-1) Request - with first Status Code = 10h; Response - no response prepared (A6-3-8-3-2) Request - with first Status Code = 10h; Response - no response prepared (A6-3-8-3-2) Request - with first Status Code = 10h; Response - Normal (A6-3-8-3-1) Request - with first Status Code = 10h; Response - Normal (A6-3-8-6-1) Request - with non-uinteger(integer) atom for 3rd statusCode; Response - no response prep	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: tinteger) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - with unexpected token(88: uinteger) in InvokingID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F0: CtrIToken) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F6: Response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(F6: Response - no response - no response prepared (A6-3-2-3-1(2)) Request - with unexpected token(98: integer) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - unexpected token(88: uinteger) in MethodID; Response - no response prepared (A6-3-2-3-1(2)) Request - with invalid token type of StartList: Oe0h; Response - no response prepared (A6-3-2-1) Request - with invalid token type of StartList: Oe0h; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(1)) Request - with byte atom for value in Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(2)) Request - with invalid token type of EndList: Oe0h; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(2)) Request - with invalid token type of EndList: Oe0h; Response - no response prepared (A6-3-2-1) Request - with invalid token type of EndList: Oe0h; Response - no response prepared (A6-3-8-1) Request - with invalid token type of StatusCode Start: Oe0h; Response - no response prepared (A6-3-8-1) Request - with invalid token type of StatusCode Start: Oe0h; Response - no response prepared (A6-3-8-1) Request - with invalid token type of StatusCode; Response - no response prepared (A6-3-8-1) Request - with invalid token type of StatusCode; Response - no response prepared (A6-3-8-1) Request - with non-uinteger(integer) atom for 3rd statusCode; Response - no response prepared (A6-3-8-1) Request - with non-uinteger(integer) atom for 3rd	PASS PASS PASS PASS PASS PASS PASS PASS
(A6-3-1-2-1) Request - with invalid InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - unexpected token(98: integer) in InvokingID; Response - no response prepared (A6-3-1-3-1(2)) Request - with nonexistent MethodID; Response - no response prepared (A6-3-2-1) Request - with unexpected token(98: integer) in MethodID; Response - no response prepared (A6-3-2-1(2)) Request - with unexpected token(F0: CttToken) in MethodID; Response - no response prepared (A6-3-2-1(2)) Request - with unexpected token(8: despring in MethodID; Response - no response prepared (A6-3-2-1(2)) Request - with unexpected token(8: integer) in MethodID; Response - no response prepared (A6-3-2-1(2)) Request - with invalid token type of StartList: De0h; Response - no response prepared (A6-3-2-1(1)) Request - with invalid token type of StartList: De0h; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(1)) Request - with byte atom for value in Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(1)) Request - with byte atom for value in Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(1)) Request - with out for value in Name-Value; Response - no response prepared or Status Code: OCh(invalid_param) (A6-3-4-2-1(1)) Request - with invalid token type of EndData: Och; Response - no response prepared (A6-3-2-1) Request - with invalid token type of EndData: Och; Response - no response prepared (A6-3-2-1) Request - with first Status Code = Oth; Response - no response prepared (A6-3-8-2-1) Request - with first Status Code = Oth; Response - pass (A6-3-8-2-1) Request - with first Status Code = Oth; Response - pass (A6-3-8-2-1) Request - with first Status Code = Oth; Response - no response prepared (A6-3-8-2-1) Request - with non-uinteger(integer) atom for 3rd statusCode; Response - no response prepared (A6-3-8-1) Request - with non-uinteger(integer) atom for 3rd statusCode; Response - no response prepared (A6-3-8-1) Request - with n	PASS PASS PASS PASS PASS PASS PASS PASS

(A12-1-1-5-11) DataStore RequiredParams: Get with 'EndRow' encoded prior to 'StartRow'; Get response - Status Code: 0Ch (Invalid_Param)	P
(A12-1-1-5-12) DataStore RequiredParams: Get with the number of 'StartRow' > 'EndRow'; Get response - Status Code: 0Ch (Invalid_Param)	P
(A12-1-1-6-1) DataStore RequiredParams: Get with 'StartColumn'; Get response - Status Code: OCh (Invalid_Param)	P
A12-1-17-1) DataStore RequiredParams: Get with 'EndColumn'; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-0-1-1-1) MBR RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-0-1-1-2) MBR RequiredParams: Get with 'EndRow' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	P
(A12-1-1-4-5(2)) MBR RequiredParams: Get with 'StartRow' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	P
(A12-1-1-4-10) MBR RequiredParams: Get without 'StartRow' component; Get response - Pass	P
A12-1-1-5-6) MBR RequiredParams: Get with 'EndRow' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	P
(A12-1-1-5-10) MBR RequiredParams: Get without 'EndRow' component; Get response - Pass	P
A12-1-1-5-11) MBR RequiredParams: Get with 'EndRow' encoded prior to 'StartRow'; Get response - Status Code: 0Ch (Invalid_Param)	P
(A12-1-1-5-12) MBR Required Params: Get with the number of 'StartRow' > 'EndRow'; Get response - Status Code: 0Ch (Invalid_Param)	P
(A12-1-1-6-1) MBR RequiredParams: Get with 'StartColumn'; Get response - Status Code: 0Ch (Invalid_Param)	P
(A12-1-1-7-1) MBR RequiredParams: Get with 'EndColumn'; Get response - Status Code: 0Ch (Invalid_Param)	Р
12: Get() - Object Table to AdminSP Grammar check	P
(A12-0-1-1-1) Table RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-0-1-1-2) Table RequiredParams: Get with 'StartColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-0-1-1-2) Table RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-3-1-4-1) Table RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-3-1-5-1) Table RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-3-1-6-6) Table RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-3-1-6-10) Table RequiredParams: Get without 'StartCol' component; Get response - Pass	P
A12-3-1-7-6) Table RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-3-1-7-9) Table RequiredParams: Get without 'EndCol' component; Get response - Pass	P
A12-3-1-7-10) Table RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-3-1-7-10(2)) Table RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-0-1-1-1) SPInfo RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-0-1-1-2) SPInfo RequiredParams: Get with 'StartColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-0-1-1-2) SPInfo RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-3-1-4-1) SPInfo RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch (Invalid_Param)	F
A12-3-1-5-1) SPInfo RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch (Invalid_Param)	F
A12-3-1-6-6) SPInfo RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	F
A12-3-1-6-10) SPInfo RequiredParams: Get without 'StartCol' component; Get response - Pass	F
A12-3-1-7-6) SPInfo RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	F
A12-3-1-7-9) SPInfo RequiredParams: Get without 'EndCol' component; Get response - Pass	F
A12-3-1-7-10) SPInfo RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-3-1-7-10(2)) SPInfo RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-0-1-1-1) SPTemplates RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-0-1-1-2) SPTemplates RequiredParams: Get with 'StartColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-0-1-1-2) SPTemplates RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-3-1-4-1) SPTemplates RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-3-1-5-1) SPTemplates RequiredParams: Get with 'EndRow', Get response - Status Code: 0Ch (Invalid_Param)	P
A12-3-1-6-6) SPTemplates RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-3-1-6-10) SPTemplates RequiredParams: Get without 'StartCol' component; Get response - Pass	P
A12-3-1-7-6) SPTemplates RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-3-1-7-9) SPTemplates RequiredParams: Get without 'EndCol' component; Get response - Pass	P
A12-3-1-7-10) SPTemplates RequiredParams: Get with 'EndCol' encoded prior to 'StartCol', Get response - Status Code: 0Ch (Invalid_Param)	Р
A12-3-1-7-10(2)) SPTemplates RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: 0Ch (Invalid Param)	Р
A12-0-1-1-1) MethodID Required Params: Get with 'Table' component; Get response - Status Code: OCh (Invalid Param)	F
A12-0-1-1-2) MethodID RequiredParams: Get with 'StartColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	Р
A12-0-1-1-2) MethodID RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	F
A12-3-1-4-1) MethodID RequiredParams: Get with 'StartRow'; Get response - Status Code: OCh (Invalid_Param)	Р
A12-3-1-5-1) MethodID Required Params: Set with 'EndRow'; Get response - Status Code: OCh (Invalid Param)	P
A12-3-1-6-6) MethodID Required Params: Get with 'StartCol' > maximum; Get response - Status Code: OCh (Invalid_Param)	F
A12-3-1-6-10) MethodID Required Params: Get with out 'StartCol' component; Get response - Status Code: Golf (invalid_Faram)	F
A12-3-1-7-6) MethodID RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param) A12-3-1-7-9) MethodID RequiredParams: Get without 'EndCol' component; Get response - Pass	P P
A12-3-1-7-9) MethodID Required Params: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid Param)	F
A12-3-1-7-10) MethodiD Required Params: Get with EndCot encoded prior to StartCot'; Get response - Status Code: 0Ch (Invalid_Param) A12-3-1-7-10(2)) MethodiD Required Params: Get with the number of 'StartCot' > 'EndCot'; Get response - Status Code: 0Ch (Invalid_Param)	
	F
A12-0-1-1-1) ACE Required Params: Get with 'Table' component; Get response - Status Code: OCh (Invalid_Param)	F
A12-0-1-1-2) ACE RequiredParams: Get with 'StartColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	F
A12-0-1-1-2) ACE RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	F
A12-3-1-4-1) ACE Required Params: Get with 'StartRow'; Get response - Status Code: 0Ch (Invalid_Param)	F
A12-3-1-5-1) ACE Required Params: Get with 'EndRow'; Get response - Status Code: 0Ch (Invalid_Param)	F
A12-3-1-6-6) ACE RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	F
A12-3-1-6-10) ACE Required Params: Get without 'StartCol' component; Get response - Pass	F
A12-3-1-7-6) ACE Required Params: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	F
A12-3-1-7-9) ACE Required Params: Get without 'EndCol' component; Get response - Pass	F
A12-3-1-7-10) ACE RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-3-1-7-10(2)) ACE RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-0-1-1-1) Authority Required Params: Get with 'Table' component; Get response - Status Code: 0 Ch (Invalid_Param)	P
A12-0-1-1-2) Authority RequiredParams: Get with 'StartColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-0-1-1-2) Authority RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-3-1-4-1) Authority RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-3-1-5-1) Authority RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch (Invalid_Param)	P
A12-3-1-6-6) Authority RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	F
A12-3-1-6-10) Authority Required Params: Get without 'StartCol' component; Get response - Pass	P
	P
A12-3-1-7-6) Authority Requireurarans. Get with Endoot 2 maximum, Get response - Status Code. Och (myalid Param)	P
(A12-3-1-7-6) Authority RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param) (A12-3-1-7-9) Authority RequiredParams: Get without 'EndCol' component; Get response - Pass	
	Р

(A12-0-1-1-1) C_PIN RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	
	PASS
(A12-0-1-1-2) C_PIN RequiredParams: Get with 'StartColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) C_PIN RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-4-1) C_PIN RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-5-1) C_PIN RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-6-6) C_PIN RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-6-10) C PIN RequiredParams: Get without 'StartCol' component: Get response - Pass	PASS
(A12-3-1-7-6) C. PIN RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-9) C_PIN RequiredParams: Get without 'EndCol' component; Get response - Pass	PASS
(A12-3-1-7-10) C_PIN RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
	PASS
(A12-3-1-7-10(2)) C_PIN RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: 0Ch (Invalid_Param)	
(A12-0-1-1-1) TPerInfo RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) TPerInfo RequiredParams: Get with 'StartColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) TPerInfo RequiredParams: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-4-1) TPerInfo RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-5-1) TPerInfo RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-6-6) TPerInfo RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-6-10) TPerInfo RequiredParams: Get without 'StartCol' component; Get response - Pass	PASS
(A12-3-1-7-6) TPerInfo RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-9) TPerInfo RequiredParams: Get without 'EndCol' component; Get response - Pass	PASS
(A12-3-1-7-10) TPerInfo RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-10(2)) TPerInfo RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-1) Template Required Params: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) Template Required Params: Get with 'StartColumn' component encoded twice; Get response - Status Code: OCh (Invalid Param)	PASS
(A12-0-1-1-2) Template Required Params: 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 (Invalid_Falam)	
, , , , , , , , , , , , , , , , , , ,	PASS
(A12-3-1-5-1) Template RequiredParams: Get with 'EndRow'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-6-6) Template RequiredParams: Get with 'StartCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-6-10) Template RequiredParams: Get without 'StartCol' component; Get response - Pass	PASS
(A12-3-1-7-6) Template RequiredParams: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-9) Template Required Params: Get without 'EndCol' component; Get response - Pass	PASS
(A12-3-1-7-10) Template Required Params: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-10(2)) Template Required Params: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-1) SP RequiredParams: Get with 'Table' component; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) SP Required Params: Get with 'Start Column' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-0-1-1-2) SP Required Params: Get with 'EndColumn' component encoded twice; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-4-1) SP RequiredParams: Get with 'StartRow'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-5-1) SP RequiredParams: Get with 'EndRow'; Get response - Status Code: OCh (Invalid Param)	PASS
(A12-3-1-6-6) SP Required Params: Get with 'StartCol' > maximum; Get response - Status Code: OCh (Invalid_Param)	PASS
	PASS
(A12-3-1-6-10) SP Required Params: Get without 'StartCol' component; Get response - Pass	
(A12-3-1-7-6) SP Required Params: Get with 'EndCol' > maximum; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-9) SP RequiredParams: Get without 'EndCol' component; Get response - Pass	PASS
(A12-3-1-7-10) SP RequiredParams: Get with 'EndCol' encoded prior to 'StartCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
(A12-3-1-7-10(2)) SP RequiredParams: Get with the number of 'StartCol' > 'EndCol'; Get response - Status Code: 0Ch (Invalid_Param)	PASS
A13: Set() - Byte Table Grammar check	PASS
A13: Set() - Byte Table Grammar check (A13-2-1-2-5) DataStore OptParams-where: Set with the limit of the byte table; Set response - pass	PASS PASS
" ·	
(A13-2-1-2-5) DataStore OptParams-where: Set with the limit of the byte table; Set response - pass	PASS
(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: 0Ch (Invalid_Param)	PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass	PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param)	PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass	PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - pass	PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whithin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whithout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whithin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whithin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) MBR OptParams-value: Set with data whthin limit of the table; Set response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Pass (A13-2-1-3-5) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-6) 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 - Status Code: 0Ch (Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whithin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) MBR OptParams-value: Set with data whthin limit of the table; Set response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-6) 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 - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-6) MBR OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (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 whithin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whithout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) MBR OptParams-value: Set with data whithout limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whithout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) MBR OptParams-value: Set with data whithout limit of the table; Set response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-5) MBR OptParams-where: 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 - Status Code: 0Ch (Invalid_Param) (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 length = 0 for 'Values' parameter; Set response - Pass A13: Set() - Object Table (LockingSP) Grammar check (A13-4-1-2-1) Authority OptParams-where: Set with 'Where' parameter; Set response - Status Code: 0Ch (Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (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 whithin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whithout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) MBR OptParams-value: Set with data whithout limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whithout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) MBR OptParams-value: Set with data whithout limit of the table; Set response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-5) MBR OptParams-where: 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 - Status Code: 0Ch (Invalid_Param) (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 length = 0 for 'Values' parameter; Set response - Pass A13: Set() - Object Table (LockingSP) Grammar check (A13-4-1-2-1) Authority OptParams-where: Set with 'Where' parameter; Set response - Status Code: 0Ch (Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with the limit of the byte table; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with 'Where' > limit of the byte table; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-2-6) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) MBR OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass A13: Set() - Object Table (LockingSP) Grammar check (A13-4-1-2-1) Authority OptParams-where: Set with 'Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-2-1) Authority OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthinut limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-2-6) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-6) 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 - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass A13: Set() - Object Table (LockingSP) Grammar check (A13-4-1-2-1) Authority OptParams-where: Set with Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-2-1) Locking OptParams-where: Set with 'Where' parameter; Set response - Status Code: 0Ch (Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) MBR OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthin limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) MBR OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass A13: Set() - Object Table (LockingSP) Grammar check (A13-4-1-2-1) Authority OptParams-where: Set with Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-2-1) Locking OptParams-where: Set with TolumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-14) Locking OptParams-where: Set with Where' parameter; Set response - Status Code: 0Ch (Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set with out 'Where' parameter; Set response - Pass (A13-2-1-3-5) MBR OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) 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 - Pass (A13-4-1-4-1) Authority OptParams-where: Set with Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Authority OptParams-where: Set with Where' parameter; Set response - Status Code: 0Ch (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: 0Ch (Invalid_Param) (A13-4-1-4-14) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set without 'Where' parameter; 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 - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass A13: Set() - Object Table (LockingSP) Grammar check (A13-4-1-2-1) Authority OptParams-where: Set with Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-2-1) Authority OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-14) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-2-6) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) MBR OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass A13: Set() - Object Table (LockingSP) Grammar check (A13-4-1-2-1) Authority OptParams-where: Set with 'Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Authority OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-14) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-14) MBRControl OptParams-where: Set with ColumnName-Value not encoded in ascending order; Set respon	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Pass (A13-2-1-2-9) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-9) 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 whthin limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass A13: Set() - Object Table (LockingSP) Grammar check (A13-4-1-2-1) Authority OptParams-where: Set with Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Authority OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value not encoded in ascending order; Set response - Pass (A13-4-1-4-15) MBRControl OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) MBRControl OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Pass (A13-4-1-4-15) MBRControl OptParams-where: Se	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Pass (A13-2-1-2-9) MBR OptParams-where: Set without 'Where' parameter; Set response - Pass (A13-2-1-3-6) 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 whthin limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass A13: Set() - Object Table (LockingSP) Grammar check (A13-4-1-2-1) Authority OptParams-where: Set with Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-2-1) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-2-1) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-14) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) MBRControl OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_	PASS PASS PASS PASS PASS PASS PASS PASS
(A13-2-1-2-5) DataStore OptParams-where: Set with Where' > 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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with "Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-2-6) MBR OptParams-where: 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 (A13-2-1-3-6) 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: Set() - Object Table (LockingSP) Grammar check (A13-4-1-2-1) Authority OptParams-where: Set with 'Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Authority OptParams-where: Set with 'Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-14) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell set seponse - Pass (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-14) MBRControl OptParams-where: Set with ColumnName-Value which indicat	PASS PASS PASS PASS PASS PASS PASS PASS
(A13-2-1-2-5) DataStore OptParams-where: Set with 'Where' > 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: 0Ch (Invalid_Param) (A13-2-1-2-9) DataStore OptParams-where: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-9) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with 'Where' > limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-2-6) MBR OptParams-where: Set with ut 'Where' parameter; Set response - Pass (A13-2-1-3-5) MBR OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) 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: Set() - Object Table (LockingSP) Grammar check (A13-4-1-2-1) Authority OptParams-where: Set with 'Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Authority OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Pass (A13-4-1-4-14) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Pass (A13-4-1-4-14) MBRControl OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) MBRControl O	PASS PASS PASS PASS PASS PASS PASS PASS
(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's limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-2-6) DataStore OptParams-where: Set with Under's limit of the table; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-6) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with 'Where' Imit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-2-9) MBR OptParams-where: Set with Uthere' parameter; Set response - Pass (A13-2-1-3-5) MBR OptParams-value: Set with data whthin limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whthin limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) MBR OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass A13: Set() - Object Table (LockingSP) Grammar check (A13-4-1-2-1) Authority OptParams-where: Set with Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Authority OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value not encoded in ascending order; Set response - Pass (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) MBRControl OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A1	PASS PASS PASS PASS PASS PASS PASS PASS
(A13-2-1-2-5) DataStore OptParams-where: Set with Where' > 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: 0Ch (Invalid_Param) (A13-2-1-3-5) DataStore OptParams-value: Set with other with other bable; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthin limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-6) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-3-9) DataStore OptParams-where: Set with the limit of the byte table; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-2-1-3-5) MBR OptParams-where: 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 (A13-2-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-9) MBR OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-3-9) MBR OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-4-1-4-15) Authority OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Pass (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Pass (A13-4-1-4-14) MBRControl OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Pass (A13-4-1-4-14) MBRControl OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (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 whthin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Pass (A13-2-1-3-9) DataStore OptParams-value: Set with the limit of the bable; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with the limit of the byte table; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with the limit of the byte table; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with Where' > 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 whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) MBR OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-3-9) MBR OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass A13 Set() - Object Table (LockingSP) Grammar check (A13-4-1-2-1) Authority OptParams-where: Set with "Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Pass (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value not encoded in ascending order; Set response - Pass (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value not encoded in ascending order; Set response - Pass (A13-4-1-4-15) MBRControl OptParams-where: Set with ColumnName-Value not encoded in ascending order; Set response - Pass (A13-4-1-4-15)	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (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 - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with the limit of the byte bable; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with Where' parameter; Set response - Pass (A13-2-1-2-9) MBR OptParams-where: 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 (A13-2-1-3-6) MBR OptParams-value: Set with data whthout limit of the table; Set response - Pass A13: Set() - Object Table (LockingSP) Grammar check (A13-4-1-2-1) Authority OptParams-where: Set with Where' parameter; Set response - Status Code: OCh (Invalid_Param) (A13-4-1-4-15) Authority OptParams-where: Set with 'Where' parameter; Set response - Status Code: OCh (Invalid_Param) (A13-4-1-4-14) Locking 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 (A13-4-1-4-15) MBRControl OptParams-where: Set with ColumnName-Value ont encoded in ascending order; Set response - Pass (A13-4-1-4-15) MBRControl OptPara	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (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 - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-6) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - Pass (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" > limit of the table; Set response - Pass (A13-2-1-3-5) MBR OptParams-walue: Set with data whithout limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whithout limit of the table; Set response - Pass (A13-2-1-3-9) MBR OptParams-value: Set with data whithout limit of the table; Set response - Pass (A13-2-1-3-1) MBR OptParams-value: Set with Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-2-1) Locking OptParams-where: Set with "Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-14) MBRControl OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-14) MBRControl OptParams-where: Set with ColumnName-Value which indicate the same cell's modifica	PASS PASS PASS PASS PASS PASS PASS PASS
(A13-2-1-2-5) DataStore OptParams-where: Set with Where' > 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: 0Ch (Invalid_Param) (A13-2-1-3-5) DataStore OptParams-where: Set with data whithin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data without limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data without limit of the table; Set response - Pass (A13-2-1-2-5) MBR OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with the limit of the byte table; Set response - Pass (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 otal without limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-walue: Set with data without limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data without limit of the table; Set response - Pass (A13-2-1-3-9) MBR OptParams-value: Set with data without limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) MBR OptParams-value: Set with data without limit of the table; Set response - Pass A13. Set() - Object Table (LockingSP) Grammar check (A13-4-1-2-1) Authority OptParams-where: Set with Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-2-1) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value not encoded in ascending order; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with Verify parameter; Set respo	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (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 - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data whthout limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-6) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-5) MBR OptParams-where: Set with the limit of the byte table; Set response - Pass (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" > limit of the table; Set response - Pass (A13-2-1-3-5) MBR OptParams-walue: Set with data whithout limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data whithout limit of the table; Set response - Pass (A13-2-1-3-9) MBR OptParams-value: Set with data whithout limit of the table; Set response - Pass (A13-2-1-3-1) MBR OptParams-value: Set with Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-2-1) Locking OptParams-where: Set with "Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-14) MBRControl OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-14) MBRControl OptParams-where: Set with ColumnName-Value which indicate the same cell's modifica	PASS PASS PASS PASS PASS PASS PASS PASS
(A13-2-1-2-5) DataStore OptParams-where: Set with Where' > 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: 0Ch (Invalid_Param) (A13-2-1-3-5) DataStore OptParams-where: Set with data whithin limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data without limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data without limit of the table; Set response - Pass (A13-2-1-2-5) MBR OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with the limit of the byte table; Set response - Pass (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 otal without limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-walue: Set with data without limit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data without limit of the table; Set response - Pass (A13-2-1-3-9) MBR OptParams-value: Set with data without limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) MBR OptParams-value: Set with data without limit of the table; Set response - Pass A13. Set() - Object Table (LockingSP) Grammar check (A13-4-1-2-1) Authority OptParams-where: Set with Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-2-1) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value not encoded in ascending order; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-15) Locking OptParams-where: Set with Verify parameter; Set respo	PASS PASS PASS PASS PASS PASS PASS PASS
(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: 0Ch (Invalid_Param) (A13-2-1-3-6) DataStore OptParams-where: Set with Universe parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data within limit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data without limit of the table; Set response - Pass (A13-2-1-3-9) DataStore OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with length = 0 for 'Values' parameter; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with Where' > limit of the table; Set response - Pass (A13-2-1-2-9) MBR OptParams-where: Set with Where' parameter; Set response - Pass (A13-2-1-2-9) MBR OptParams-value: Set with data without limit of the table; Set response - Pass (A13-2-1-3-5) MBR OptParams-value: Set with data without limit of the table; Set response - Pass (A13-2-1-3-9) MBR OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-3-9) MBR OptParams-value: Set with length = 0 for 'Values' parameter; Set response - Pass A13-Set() - Object Table (LockingSP) Grammar check (A13-4-1-2-1) Authority OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-15) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-2-1) MBRControl OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-2-1) MBRControl OptParams-whe	PASS PASS PASS PASS PASS PASS PASS PASS
(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' s limit of the table; Set response - Status Code: 0Ch (Invalid_Param) (A13-2-1-2-6) DataStore OptParams-where: Set with Where' parameter; Set response - Pass (A13-2-1-3-5) DataStore OptParams-value: Set with data within ulimit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data within ulimit of the table; Set response - Pass (A13-2-1-3-6) DataStore OptParams-value: Set with data without limit of the table; Set response - Pass (A13-2-1-2-6) MBR OptParams-value: Set with the limit of the byte table; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with the limit of the byte table; Set response - Pass (A13-2-1-2-6) MBR OptParams-where: Set with the Value of Parameter, Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data within ulimit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data within ulimit of the table; Set response - Pass (A13-2-1-3-6) MBR OptParams-value: Set with data within ulimit of the table; Set response - Pass (A13-2-1-3-9) MBR OptParams-value: Set with data within ulimit of the table; Set response - Pass (A13-4-1-3-1) Authority OptParams-value: Set with Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-1) Authority OptParams-where: Set with Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-14) Locking OptParams-where: Set with Where' parameter; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-14) Locking OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-14) MBRControl OptParams-where: Set with ColumnName-Value which indicate the same cell's modification; Set response - Status Code: 0Ch (Invalid_Param) (A13-4-1-4-14) MBRControl OptParams-where: Set with ColumnName-Value which indicate the same	PASS PASS PASS PASS PASS PASS PASS PASS

(A14-1-3-2-5(2)) MethodID OptParams-where: Next with an exiting UID in the table; Next response - Pass	PASS
(A14-1-3-2-8) MethodID OptParams-where: Next with nonexistent UID; Next response - Status Code: 0Ch (Invalid_Param)	PASS
(A14-13-2-11) 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 - all UIDs	PASS
(A14-1-3-3-6(2)) MethodID OptParams-count: Next with count = 0; Next response - no UID returned	PASS
(A14-1-3-3-10) MethodID OptParams-count: Next with omitted count; Next response - Pass	PASS
(A14-1-3-2-5(2)) ACE OptParams-where: Next with an exiting UID in the table; Next response - Pass	PASS
(A14-1-3-2-8) ACE OptParams-where: Next with nonexistent UID; Next response - Status Code: 0Ch (Invalid_Param)	PASS
(A14-1-3-2-11) ACE OptParams-where: Next with omitted 'Where' parameter; Next response - first UID in the table	PASS
(A14-1-3-3-6) ACE OptParams-count: Next with a larger the number of UIDs; Next response - all UIDs	PASS
(A14-1-3-3-6(2)) ACE OptParams-count: Next with count = 0; Next response - no UID returned	PASS
(A14-1-3-3-10) ACE OptParams-count: Next with omitted count; Next response - Pass	PASS
(A14-1-3-2-5(2)) Authority OptParams-where: Next with an exiting UID in the table; Next response - Pass	PASS
(A14-1-3-2-8) Authority OptParams-where: Next with nonexistent UID; Next response - Status Code: 0Ch (Invalid_Param)	PASS
(A14-1-3-2-11) Authority OptParams-where: Next with omitted 'Where' parameter: Next response - first UID in the table	PASS
(A14-1-3-3-6) Authority OptParams-count: Next with a larger the number of UIDs; Next response - all UIDs	PASS
	PASS
(A14-1-3-3-6(2)) Authority OptParams-count: Next with count = 0; Next response - no UID returned	
(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	PASS
(A14-1-3-2-8) C_PIN OptParams-where: Next with nonexistent UID; Next response - Status Code: 0Ch (Invalid_Param)	PASS
(A14-1-3-2-11) C_PIN OptParams-where: Next with omitted 'Where' parameter; Next response - first UID in the table	PASS
(A14-1-3-3-6) C_PIN OptParams-count: Next with a larger the number of UIDs; Next response - all UIDs	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	PASS
(A14-1-3-2-8) Template OptParams-where: Next with nonexistent UID; Next response - Status Code: 0Ch (Invalid_Param)	PASS
(A14-1-3-2-11) Template OptParams-where: Next with omitted 'Where' parameter; Next response - first UID in the table	PASS
(A14-1-3-3-6) Template OptParams-count: Next with a larger the number of UIDs; Next response - all UIDs	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
	PASS
(A14-1-3-3-10) SP OptParams-count: Next with omitted count; Next response - Pass	PASS
A15: GetACL()-AdminSP Basic Grammar check	PASS
(A15-1-1-0-1) Table Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-2-1-1(2)) Table ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-1-1(2)) Table ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass	PASS
(A15-12-2-1(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 response - Pass	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 response - Pass	PASS
(A15-1-2-2-1(2)) SPInfo ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass	PASS
	PASS
(A15-1-2-2-1(2)) SPInfo ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass	
(A15-1-2-3-1) SPInfo ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-1-0-1) SPTemplates Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-2-1-1(2)) SPTemplates ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-1-1(2)) SPTemplates ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) SPTemplates ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) SPTemplates ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass	PASS
	PASS
(A15-1-2-3-1) SPTemplates ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority)	
(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 medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass	PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass	PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass	PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL 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'; GetACL response - Status Code: 01h (Not_Authority)	PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL 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'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority)	PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL 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'; GetACL response - Status Code: 01h (Not_Authority)	PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL 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'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority)	PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL 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'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL 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; GetACL response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL 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'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-3-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL 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; GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-3-1) ACE ReqParams: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-3-1) ACE ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority)	PASS PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL 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'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) ACE ReqParams-methodID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-3-1) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-3-1) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-3-1) ACE ReqParams-GetACL with nonexistence of 'InvokingID' and 'MethodID; GetACL response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) Authority Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority)	PASS PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL 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; GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) ACE ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-3-1) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-3-1) ACE ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) Authority Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with long atom for MethodID; GetACL 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; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-3-1) MethodID ReqParams: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with nonexistence of 'InvokingID' GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) Authority Condition: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) Authority Condition: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL 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; GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) ACE ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-3-1) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-3-1) ACE ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) Authority Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with long atom for MethodID; GetACL 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; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-3-1) MethodID ReqParams: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with nonexistence of 'InvokingID' GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) Authority Condition: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) Authority Condition: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-3-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) ACE ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-getACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-3-1) ACE ReqParams-invokingID: GetACL with onexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) Authority Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL 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'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) ACE ReqParams-methodID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-1-2(2)) ACE ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-3-1) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-3-1) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-1-1(2)) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) Authority Condition: GetACL with util UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-3-1) Authority ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-3-1) Authority ReqParams-methodID:	PASS PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-methodID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with long atom for MethodID; GetACL 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; GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) ACE ReqParams-methodID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-3-1) ACE ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) Authority Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-3-1) Authority ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-3-1) Authority ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-3-1) Authority ReqParams: GetACL with long atom for MethodID; GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) C_PIN Condition: Get	PASS PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with long atom for MethodID; GetACL 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; GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) ACE ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-getACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) Authority Condition: GetACL with uton UID of access control table; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with nonexistence of 'InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-methodID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) Authority ReqParams-methodID: GetACL with nonexistence of 'InvokingID; GetACL response - Pass (A15-1-2-3-1) Authority ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-3-1) Authority ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-3-1) Authority ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) C_PIN	PASS PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with long atom for MethodID; GetACL 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; GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) Authority Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) Authority ReqParams-methodID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) Authority ReqParams-methodID: GetACL with nedium atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) Authority ReqParams-methodID: GetACL with nedium atom for InvokingID; GetACL response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) C_PIN Condition: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) C_PIN ReqParams-invokingID: GetACL with undi	PASS PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with long atom for MethodID; GetACL 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; GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) ACE ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-getACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) Authority Condition: GetACL with uton UID of access control table; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with nonexistence of 'InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-methodID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) Authority ReqParams-methodID: GetACL with nonexistence of 'InvokingID; GetACL response - Pass (A15-1-2-3-1) Authority ReqParams-methodID: GetACL with nonexistence of 'InvokingID; GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) C_PIN Condition: GetACL with onexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) C_PIN ReqParams-invokingID: GetACL with medium atom for InvokingID; Get	PASS PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with long atom for MethodID; GetACL 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; GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) Authority Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) Authority ReqParams-methodID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) Authority ReqParams-methodID: GetACL with nedium atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) Authority ReqParams-methodID: GetACL with nedium atom for InvokingID; GetACL response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) C_PIN Condition: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) C_PIN ReqParams-invokingID: GetACL with undi	PASS PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-methodID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL 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; GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with nedium atom for MethodID; GetACL response - Pass (A15-1-2-3-1) ACE ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-10-1) Authority Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-methodID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-methodID: GetACL with nonexistence of 'InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) C_PIN ReqParams-methodID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) C_PIN ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2	PASS PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL 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'; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-2-1(2)) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-3-1) ACE ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) Authority Condition: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-methodID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-1) (DetACL without UID of access control table; GetACL response - Pass (A15-1-2-1(2)) Authority ReqParams-methodID: GetACL with long atom for MethodID'; GetACL response - Pass (A15-1-2-1(2)) CetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-1-0-1) CetAC Params-invokingID: GetACL with long atom for InvokingID	PASS PASS PASS PASS PASS PASS PASS PASS
(A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) MethodID ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-2-1(2)) MethodID ReqParams-methodID: GetACL with medium atom for MethodID; GetACL 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; GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) ACE Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority) (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) ACE ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass (A15-1-2-1(2)) ACE ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass (A15-1-2-1(2)) ACE ReqParams-getACL with nonexistence of 'InvokingID' and 'MethodID', GetACL response - Status Code: 01h (Not_Authority) (A15-1-0-1) Authority Condition: GetACL with nonexistence of 'InvokingID' and 'MethodID', GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-methodID: GetACL with nonexistence of 'InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) Authority ReqParams-methodID: GetACL with nonexistence of 'InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) C_PIN ReqParams-methodID: GetACL with nonexistence of 'InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) C_PIN ReqParams-methodID: GetACL with nonexistence of 'InvokingID; GetACL response - Pass (A15-1-2-1-1(2)) C_PIN Re	PASS PASS PASS PASS PASS PASS PASS PASS

(A15-1-2-1-1(2)) TPerInfo RegParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass	
	PASS
(A15-1-2-2-1(2)) TPerInfo RegParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) TPerInfo RegParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass	PASS
(A15-12-3-1) TPerInfo ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority)	PASS
	PASS
(A15-1-1-0-1) Template Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority)	
(A15-1-2-1-1(2)) Template ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-1-1(2)) Template ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) Template ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) Template ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-3-1) Template ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-1-0-1) SP Condition: GetACL without UID of access control table; GetACL response - Status Code: 01h (Not_Authority)	PASS
(A15-1-2-1-1(2)) SP ReqParams-invokingID: GetACL with medium atom for InvokingID; GetACL response - Pass	PASS
	PASS
(A15-1-2-1-1(2)) SP ReqParams-invokingID: GetACL with long atom for InvokingID; GetACL response - Pass	
(A15-1-2-2-1(2)) SP ReqParams-methodID: GetACL with medium atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-2-1(2)) SP ReqParams-methodID: GetACL with long atom for MethodID; GetACL response - Pass	PASS
(A15-1-2-3-1) SP ReqParams: GetACL with nonexistence of 'InvokingID' and 'MethodID'; GetACL response - Status Code: 01h (Not_Authority)	PASS
A19: RevertSP() Grammar check	PASS
(A19-1-3-1-10) KeepGlbRange: RevertSP to LockingSP with the omitted KeepGlobalRangeKey; RevertSP Response - Pass	PASS
Revert LockingSP	PASS
Start Session with HostChallenge - AdminSP	PASS
Sync Session - AdminSP	PASS
	PASS
LockingSP-Revert - Request	
LockingSP.Revert - Response	PASS
End Session - Request	PASS
End Session - Response	PASS
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	PASS
Check the state of LockingSP	PASS
End Session - Request	PASS
End Session - Response	PASS
D1: ACE.Set() Grammar and Effect	PASS
	PASS
(D1-1-1-1-9) ACE. Set Grammar: Request with right params; Set Response - 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 Response - Pass	
	PASS
(D1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (Invalid_Param)	PASS N/A
(D1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (Invalid_Param)	N/A
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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	N/A PASS PASS
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value	N/A PASS PASS PASS
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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	N/A PASS PASS
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (Invalid_Param) (D1-1-2-1-1) ACE.Set 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 value (D1-1-3-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value	N/A PASS PASS PASS PASS
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value (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	N/A PASS PASS PASS PASS
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value (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	N/A PASS PASS PASS PASS PASS
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value (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)	N/A PASS PASS PASS PASS PASS PASS
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value (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) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value (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)	N/A PASS PASS PASS PASS PASS PASS
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value (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) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (Invalid_Param) (D1-1-2-1-1) ACE.Set 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 value (D1-1-3-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value D2: Autthority.Set() testing (D2-1-2-1-1) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (D2-1-2-3-1) Authenticate - User1(Enabled=1); Authenticate Response - Success (AuthStatus = 01h) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-2) The previous successful authentication result with this authority in this session shall not be affected	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value (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) (D2-1-2-3-2) The previous successful authentication result with this authority in this session shall not be affected (D2-1-2-2-1) Start Session - as User1(Enabled=1); Sync Session - Pass	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value (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) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-2) 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=0); Sync Session - Pass (D2-1-2-3-1) Start Session - as User1(Enabled=0); Sync Session - Fail	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value (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) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-1) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-1-2-3-1) Start Session - as User1(Enabled=1); Sync Session - Fail (D2-1-3-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value (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) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-1) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-1-2-3-1) Start Session - as User1(Enabled=1); Sync Session - Fail (D2-1-3-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value (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 (D2-1-2-1) Authenticate - User1(Enabled=1); Authenticate Response - Success (AuthStatus = 01h) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-1) 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) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-1-3-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-2) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value (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) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-2) 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 (D2-1-3-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-1) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-1-2-1-2) Set Request: PIN = Null; Response: Pass	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value (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) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-2) 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) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-1-3-1-1) Authority. Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-2) Authority. Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-1-2-1-2) Set Request: PIN = Null; Response: Pass (D3-1-2-1-2) Set Request: PIN set) Set (PIN se	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value (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) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-2) 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=0); Sync Session - Pass (D2-1-2-3-1) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-1-3-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-2) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-1-2-1-2) Set Request: PIN = Null; Response: Pass (D3-1-2-1-2) Set Request: PIN swith 32 byte; Response: Pass (D3-1-3-1-1) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set value	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value (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) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-2) 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) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-1-3-1-1) Authority. Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-2) Authority. Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-1-2-1-2) Set Request: PIN = Null; Response: Pass (D3-1-2-1-2) Set Request: PIN set) Set (PIN se	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value (D1-1-3-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value (D1-1-3-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value (D2-1-2-1) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (D2-1-2-1) Authenticate - User1(Enabled=1); Authenticate Response - Success (AuthStatus = 01h) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-1) 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) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-1-3-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-2) Authority.Set in a transaction and endTran' status = 0; The new value changes back to the original value (D3-1-2-1-2) Set Request: PIN = Null; Response: Pass (D3-1-2-1-2) Set Request: PIN = Null; Response: Pass (D3-1-3-1-1) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set value (D3-1-3-1-1) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value (D1-1-3-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value (D1-1-3-1-2) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value (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) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-1) 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) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-1-3-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-2) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value (D3-1-2-1-2) Set Request: PIN = Null; Response: Pass (D3-1-2-1-2) Set Request: PIN with 32 byte; Response: Pass (D3-1-3-1-1) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value (D3-1-3-1-2) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (Invalid_Param) (D1-1-2-1-1) ACE.Set 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 value (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) 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) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-2) 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) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-1-3-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-2) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D3-1-3-1-2) Set Request: PIN = Null; Response: Pass (D3-1-2-1-2) Set Request: PIN = Null; Response: Pass (D3-1-2-1-2) Set Request: PIN in a transaction with endTransaction status = 0; The PIN retains the set value (D3-1-3-1-1) 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' (D4-1-2-1-2) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: OCh (Invalid_Param)	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (Invalid_Param) (D1-1-2-1-1) ACE.Set is issued to verify; Data comparison - Matching (D1-1-3-1-1) 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 value (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) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-1) 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) Start Session - as User1(Enabled=0); Sync Session - Pass (D2-1-2-3-1-1) Authority. Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-2) Authority. Set in a transaction and endTran' status = 1; The value changes back to the original value (D3-1-3-1-2) Set Request: PIN = Null; Response: Pass (D3-1-3-1-1) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set value (D3-1-3-1-3-1-2) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value (D3-1-3-1-2) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value (D3-1-3-1-2) Ret new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value (D3-1-3-1-1) RangeStart/Length: overlaps with any other range's LBA; Response - Pass	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (Invalid_Param) (D1-1-2-1-1) ACE.Set 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 value (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) 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) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-2) 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) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-1-3-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-2) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D3-1-3-1-2) Set Request: PIN = Null; Response: Pass (D3-1-2-1-2) Set Request: PIN = Null; Response: Pass (D3-1-2-1-2) Set Request: PIN in a transaction with endTransaction status = 0; The PIN retains the set value (D3-1-3-1-1) 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' (D4-1-2-1-2) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: OCh (Invalid_Param)	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (Invalid_Param) (D1-1-2-1-1) ACE.Set is issued to verify; Data comparison - Matching (D1-1-3-1-1) 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 value (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) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-1) 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) Start Session - as User1(Enabled=0); Sync Session - Pass (D2-1-2-3-1-1) Authority. Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-2) Authority. Set in a transaction and endTran' status = 1; The value changes back to the original value (D3-1-3-1-2) Set Request: PIN = Null; Response: Pass (D3-1-3-1-1) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set value (D3-1-3-1-3-1-2) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value (D3-1-3-1-2) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value (D3-1-3-1-2) Ret new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value (D3-1-3-1-1) RangeStart/Length: overlaps with any other range's LBA; Response - Pass	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-1) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 value (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) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-1) Start Session - as User1(Enabled=1); Sync Session - Pass (D2-1-2-3-1) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-1-3-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-2) Authority. Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-1-2-1-2) Set Request: PIN = Nult; Response: Pass (D3-1-3-1-1) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set value (D3-1-3-1-2) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value D4: Locking.Set() for 'RangeStart'Length: overlaps with any other range's LBA; Response - Pass (D4-1-3-1-1) RangeStart/Length: overlaps with any other range's LBA; Response - Pass (D4-1-3-1-1) RangeStart/Length: overlaps with any other range's LBA; Response - Pass (D4-1-3-1-1) RangeStart/Length: Overlaps with any other range's LBA; Response - Pass (D4-1-3-1-1) RangeStart/Length: Overlaps with any other range's LBA; Response - Pass (D4-1-3-1-1) RangeStart/Length: Overlaps with any other range's LBA; Response - Pass (D4-1-	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-3) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 (D1-1-3-1-2) ACE.Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-1-3-1-2) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set value (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) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-2) 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) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-1-3-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-2) Authority.Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C PIN.Set() (D3-1-2-1-2) Set Request: PIN = Null; Response: Pass (D3-1-3-1-2) Set Request: PIN with 32 byte; Response: Pass (D3-1-3-1-2) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set value (D3-1-3-1-2) 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' (D4-1-3-1-1) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass (D4-1-3-1-1) RangeStart/Len Effect: Get with right Name-Value's values; Response - Pass (D4-1-3-1-1) RangeStart/Len Effect: Get with right Name-Value's values; Response with Get - no LBA covered by t	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-1) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (Invalid_Param) (D1-1-2-1-1) ACE.Set with different UIDs; ACE.Set is issued to verify data - Matching (D1-1-2-1-2) ACE.Set with different UIDs; ACE.Set is issued to verify data - Matching (D1-1-3-1-2) ACE.Set with different UIDs; ACE.Set is issued to verify data - Matching (D1-1-3-1-2) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set value (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) (D2-1-2-3-2) The previous successful authentication result with this authority in this session shall not be affected (D2-1-2-3-2) 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) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-1-3-1-1) Authority. Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-2) Authority. Set in a transaction and endTran' status = 1; The value changes back to the original value (D3-1-3-1-2) Set Request: PIN = Null; Response: Pass (D3-1-2-1-2) Set Request: PIN = Null; Response: Pass (D3-1-3-1-1) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set value (D3-1-3-1-2) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value (D3-1-3-1-2) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value (D3-1-3-1-1) RangeStart/Len Effect: with right Name-Value's values; Response - Pass (D4-1-3-1-1) RangeStart/Len Effect: with right Name-Value's values; Response - Pass (D4-1-3-1-1) Range	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 (D1-1-3-1-2) ACE. Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-1-3-1-2) ACE. Set in a transaction with endTransaction status = 0; The new value retains the set value (D1-1-3-1-2) ACE. Set in a transaction with endTransaction status = 1; The value changes back to the original value (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) (D2-1-2-3-2) 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=0); Sync Session - Pass (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 value (D2-1-3-1-2) Authority. Set in a transaction and endTran' status = 0; The new value retains the set value (D3-1-3-1-2) Authority. Set in a transaction and endTran' status = 1; The value changes back to the original value (D3-1-3-1-2) Set Request: PIN = Null; Response: Pass (D3-1-2-1-2) Set Request: PIN = Null; Response: Pass (D3-1-3-1-2) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set value (D3-1-3-1-2) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value (D4-1-3-1-1) RangeStart' and 'RangeLength' (D4-1-3-1-1) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass (D4-1-3-1-1) RangeStart/Len Effect: Set with right Name-Value's values; Response with Get - no LBA covered by Set() (D4-1-3-3-1) RangeStart/Len Effect: with 'RangeStart'= changed and 'RangeLength'; Ot (Pseponse with Get - t	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: OCh (Invalid_Param) (D1-1-2-1-1) ACE. Set 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 end Transaction status = 0; The new value retains the set value (D1-1-3-1-2) ACE.Set in a transaction with end Transaction 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) (D2-1-2-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-1-2-3-2) 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=0); Sync Session - Pass (D2-1-2-3-1) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-1-3-1-1) Authority. Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-2) Authority. Set in a transaction and endTran' status = 1; The value changes back to the original value D3: C_PIN.Set() (D3-1-2-1-2) Set Request: PIN = Null; Response: Pass (D3-1-2-1-2) Set Request: PIN with 32 byte; Response: Pass (D3-1-3-1-1) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set value (D3-1-3-1-2) 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' (D4-1-3-1-1) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass (D4-1-3-1-1) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass (D4-1-3-1-1) RangeStart/Len Effect: with 'RangeStart'-changed and 'RangeLength'-G0, Response with Get - the values as intended by Set() (D4-1-3-3-1) RangeStart/Len Effect: with 'RangeStart	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (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 (D1-1-3-1-2) ACE. Set with different UIDs; ACE.Get is issued to verify data - Matching (D1-1-3-1-2) ACE. Set in a transaction with endTransaction status = 0; The new value retains the set value (D1-1-3-1-2) ACE. Set in a transaction with endTransaction status = 1; The value changes back to the original value (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) (D2-1-2-3-2) 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=0); Sync Session - Pass (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 value (D2-1-3-1-2) Authority. Set in a transaction and endTran' status = 0; The new value retains the set value (D3-1-3-1-2) Authority. Set in a transaction and endTran' status = 1; The value changes back to the original value (D3-1-3-1-2) Set Request: PIN = Null; Response: Pass (D3-1-2-1-2) Set Request: PIN = Null; Response: Pass (D3-1-3-1-2) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set value (D3-1-3-1-2) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value (D4-1-3-1-1) RangeStart' and 'RangeLength' (D4-1-3-1-1) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass (D4-1-3-1-1) RangeStart/Len Effect: Set with right Name-Value's values; Response with Get - no LBA covered by Set() (D4-1-3-3-1) RangeStart/Len Effect: with 'RangeStart'= changed and 'RangeLength'; Ot (Pseponse with Get - t	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (Invalid_Param) (D1-1-2-1-1) ACE. Get is is sued to verify, Data companison - Matching (D1-12-1-1) ACE. Get is is sued to verify, Data companison - Matching (D1-13-1-1) ACE. Set in a transaction with endTransaction status = 0; The new value retains the set value (D1-13-1-1) ACE. Set in a transaction with endTransaction status = 0; The new value retains the set value (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 (D2-12-21-1) 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) (D2-12-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (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=0); Sync Session - Pass (D2-12-3-1) 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 value (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 with 32 byte; 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 value (D3-13-1-2) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value D4-12-12-12 RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: 0Ch (Invalid_Param) (D4-13-1-1) RangeStart/Length: overlaps with any other range's LBA; Response - Pass (D4-13-4-1) RangeStart/Len Effect: Set with right Name-Value's values; Response Pin Set value (D4-13-3-1) RangeStart/Len Effect: with 'RangeLen	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (Invalid_Param) (D1-1-2-1-1) ACE.Set its issued to verify Data comparison - Matching (D1-13-1-1) ACE.Set in size and to verify Data comparison - Matching (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set value (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set value (D1-13-1-12) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value (D2-12-2-14) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (02-12-2-14) Unthenticate - User1(Enabled'-0); Authenticate Response - Success (AuthStatus = 01h) (D2-12-3-1) Authenticate - User1(Enabled-0); Authenticate Response - Fail (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-0); Sync Session - Pass (D2-12-3-1) Start Session - as User1(Enabled-0); Sync Session - Pail (D2-13-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D2-13-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D3-13-1-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN in a transaction with endTransaction status = 0; The PIN retains the set value (D3-13-1-1) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value (D3-13-1-1) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value (D3-13-1-1) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass (D4-13-1-1) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass (D4-13-1-1) RangeStart/Len Effect: with 'RangeStart'-changed and 'RangeLength'-(); Response with Get - no LBA covered	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (Invalid_Param) (D1-12-1-1) ACE.Set with different UIDs; ACE.Oet 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 value (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 - Success (AuthStatus = 01h) (D2-12-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-12-3-2) The previous successful authentication result with this authority in this session shall not be affected (D2-12-2-3) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-12-3-1) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-13-3-1) Authority. Set in a transaction and endTran' status = 0; The new value retains the set value (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 with 32 byte; Response: Pass (D3-13-11) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set value (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' (D4-13-11) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: OCh (Invalid_Param) (D4-13-11) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: OCh (Invalid_Param) (D4-13-2-1) RangeStart/Length: overlaps with any other range's LBA; Response - Was provided by Set() (D4-13-21) RangeStart/Length: overlaps with any other range's LBA; Response - Was provided by Set() (D4-13-21) RangeStart/Length: overla	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (Invalid_Param) (D1-1-2-1-1) ACE.Set its issued to verify Data comparison - Matching (D1-13-1-1) ACE.Set in size and to verify Data comparison - Matching (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set value (D1-13-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set value (D1-13-1-12) ACE.Set in a transaction with endTransaction status = 1; The value changes back to the original value (D2-12-2-14) Get Request (User1) - 'Enabled' column (05h); Get Response - 1/0 in 'Enabled' column (02-12-2-14) Unthenticate - User1(Enabled'-0); Authenticate Response - Success (AuthStatus = 01h) (D2-12-3-1) Authenticate - User1(Enabled-0); Authenticate Response - Fail (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-0); Sync Session - Pass (D2-12-3-1) Start Session - as User1(Enabled-0); Sync Session - Pail (D2-13-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D2-13-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D3-13-1-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN = Null; Response: Pass (D3-12-12) Set Request: PIN in a transaction with endTransaction status = 0; The PIN retains the set value (D3-13-1-1) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value (D3-13-1-1) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value (D3-13-1-1) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass (D4-13-1-1) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass (D4-13-1-1) RangeStart/Len Effect: with 'RangeStart'-changed and 'RangeLength'-(); Response with Get - no LBA covered	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: 0Ch (Invalid_Param) (D1-12-1-1) ACE.Set with different UIDs; ACE.Oet 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 value (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 - Success (AuthStatus = 01h) (D2-12-3-1) Authenticate - User1(Enabled=0); Authenticate Response - Fail (D2-12-3-2) The previous successful authentication result with this authority in this session shall not be affected (D2-12-2-3) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-12-3-1) Start Session - as User1(Enabled=0); Sync Session - Fail (D2-13-3-1) Authority. Set in a transaction and endTran' status = 0; The new value retains the set value (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 with 32 byte; Response: Pass (D3-13-11) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set value (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' (D4-13-11) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: OCh (Invalid_Param) (D4-13-11) RangeStart/Length: overlaps with any other range's LBA; Response - Status Code: OCh (Invalid_Param) (D4-13-2-1) RangeStart/Length: overlaps with any other range's LBA; Response - Was provided by Set() (D4-13-21) RangeStart/Length: overlaps with any other range's LBA; Response - Was provided by Set() (D4-13-21) RangeStart/Length: overla	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-13) ACE.Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: OCh (Invalid_Param) (D1-1-2-1-1) ACE.Set it is issued to verify; Data comparison - Matching (D1-1-3-1-1) ACE.Set it is issued to verify; Data comparison - Matching (D1-1-3-1-1) ACE.Set in a transaction with endTransaction status = 0; The new value retains the set value (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) Grat 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) (D2-1-2-3-1) Authenticate - User1(Enabled-0); Authenticate Response - Fait (D2-1-2-3-2) The previous successful authentication result with this authority in this session shall not be affected (D2-1-2-2-1) Start Session - as User1(Enabled-0); Sync Session - Pass (D2-1-2-3-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-2-3-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-1) Authority.Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-1) Set Request: PIN = Null; Response: Pass (D3-1-2-1-2) Set Request: PIN with 32 byte; Response: Pass (D3-1-2-1-2) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set value (D3-1-3-1-1) Set new PIN in a transaction with endTransaction status = 1; The PIN changes back to the original value D4-Locking.Set() for 'RangeStart'Len Effect: Set with right Name-Value's values; Response - Status Code: OCh (Invalid_Param) (D4-1-3-1-1) RangeStart/Len Effect: With 'RangeStart' enhanged and 'RangeLength'-10; Response with Get - the values as intended by Set() (D4-1-3-3-1) RangeStart/Len Effect: With 'RangeStart' enhange	N/A PASS PASS PASS PASS PASS PASS PASS PA
(D1-1-1-13) ACE Set Grammar: Request with AC_Element > maximum size; Set response - Status Code: OCh (Invalid_Param) (D1-1-2-1-1) ACE Set this clifferent 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 value (D1-1-3-1-2) ACE. Set in a transaction with endTransaction status = 0; The new value retains the set value (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 (1) 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) (D2-1-2-3-1) Authenticate - User1(Enabled-1); Authenticate Response - Fail (D2-1-2-3-1) 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) Start Session - as User1(Enabled-0); Sync Session - Fail (D2-1-3-1-1) Authority, Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-2) Authority, Set in a transaction and endTran' status = 0; The new value retains the set value (D2-1-3-1-2) Set Request: PIN = Null; Response: Pass (D3-1-2-1-2) Set Request: PIN = Null; Response: Pass (D3-1-1-2) Set Request: PIN in a transaction of the endTran' status = 0; The PIN retains the set value (D3-1-3-1-1) Set new PIN in a transaction with endTransaction status = 0; The PIN retains the set value (D4-1-3-1-1) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass (D4-1-3-1) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass (D4-1-3-1) RangeStart/Len Effect: Set with right Name-Value's values; Response - Pass (D4-1-3-1) RangeStart/Len Effect: With 'RangeStart'-changed and 'RangeLength'-0; Response with Get - the values as intended by Set() (D4-1-3-1) RangeStart/Len Effect: wi	N/A PASS PASS PASS PASS PASS PASS PASS PA

(D4-2-2-2-5) RdLockEnabled/Locked=1: Power-on reset; Response - 'ReadLocked' = 1	PASS
(D4-2-2-3-1) RdLockEnabled/Locked=1/0 w/ inactive MBR shadowing: Read with this locked range; Response - Pass	PASS
(D4-2-2-3-1(2)) RdLockEnabled/Locked=1/0 w/ inactive MBR shadowing: Read with multiple ranges (range2); Response - Abort/Pass(if rangeCrossing=1/0)	PASS
(D4-2-2-3-1(2)) RdLockEnabled/Locked=1/0 w/ inactive MBR shadowing: Read with multiple ranges (globalRange); Response - Abort/Pass(if rangeCrossing=1/0)	PASS
(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	PASS
	PASS
$(D4-2-2-3-3) \ RdLockEnabled/Locked=1/0: Locked \ bit=0/1 \ in \ Level \ 0 \ Discovery \ (unlocked-write \ and \ unlocked-read \ on \ other \ ranges)$	
(D4-2-2-3-5) RdLockEnabled/Locked=1/0: Power-on reset; Response - 'ReadLocked' = 1	PASS
(D4-2-2-4-1) RdLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Read with this range; Response - Pass	PASS
(D4-2-2-4-1(2)) RdLockEnabled/Locked=0/0 w/ inactive MBR shadowing; Read with multiple ranges (globalRange); Response - Abort/Pass(if rangeCrossing=1/0)	PASS
(D4-2-2-4-1(2)) RdLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Read with multiple ranges (range2); Response - Abort/Pass(if rangeCrossing=1/0)	PASS
(D4-2-2-4-2) RdLockEnabled/Locked=0/0 w/ active MBR shadowing: Read with LBA covered by this range and not by MBR; Response - Pass	PASS
(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)	PASS
(D4-2-2-4-1) RdLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Read with this range; Response - Pass	PASS
(D4-2-2-4-1(2)) RdLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Read with multiple ranges (globalRange); Response - Abort/Pass(if rangeCrossing=1/0)	PASS
(D4-2-2-4-1(2)) RdLockEnabled/Locked=0/1 w/ inactive MBR shadowing; Read with multiple ranges (range2); Response - Abort/Pass(if rangeCrossing=1/0)	PASS
(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	PASS
(D4-2-2-4-3) RdLockEnabled/Locked=0/1: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges)	PASS
(D4-2-3-1-1) ReadLock Effect in Trans: Set ReadLockEnabled in a transaction and endTran's status=0; The value retains the set value	PASS
(D4-2-3-1-2) ReadLock Effect in Trans: Set ReadLockEnabled in a transaction and endTran's status=1; The value changes back to the original value	PASS
(D4-2-5-1-2) headcook cheeci iii maiis. Set headcockchabled iii a tiansaction and end hair s status-1, the value changes back to the original value	1 A55
D4: Locking.Set() for 'WriteLockEnabled' and 'WriteLocked'	PASS
(D4-3-2-1-1) WrLockEnabled/Locked: Set WriteLockEnabled with tiny atom; Response - Pass	PASS
(D4-3-2-1-1) WrLockEnabled/Locked: Get the contents of 'WriteLockEnabled' and 'WriteLocked'; Get() retrieves the values indicated by Set()	PASS
(D4-3-2-2-1) WrLockEnabled/Locked=1/1 w/ inactive MBR shadowing: Write with this locked range; Response - Command abort	PASS
(D4-3-2-2-1(2)) WrLockEnabled/Locked=1/1 w/ inactive MBR shadowing: Write with other range; Response - Command abort	PASS
(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 abort	
	PASS
(D4-3-2-2-3) WrLockEnabled/Locked=1/1: Locked bit = 0 in Level 0 Discovery	PASS
(D4-3-2-2-5) WrLockEnabled/Locked=1/1: Power-on reset; Response - 'WriteLocked' = 1	PASS
(D4-3-2-3-1) WrLockEnabled/Locked=1/0 w/ inactive MBR shadowing: Write with this locked range; Response - Pass	PASS
(D4-3-2-3-1(2)) WrLockEnabled/Locked=1/0 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abort/Pass(if rangeCrossing=1/0)	PASS
(D4-3-2-3-1(2)) WrLockEnabled/Locked=1/0 w/ inactive MBR shadowing: Write with multiple ranges (globalrange); Response - Abort/Pass(if rangeCrossing=1/0)	PASS
(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	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)	PASS
(D4-3-2-3-5) WrLockEnabled/Locked=1/0: Power-on reset; Response - 'WriteLocked' = 1	PASS
(D4-3-2-4-1) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with this range; Response - Pass	PASS
(D4-3-2-4-1(2)) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abort/Pass(if rangeCrossing=1/0)	PASS
(D4-3-2-4-1(2)) WrLockEnabled/Locked=0/0 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abort/Pass(if rangeCrossing=1/0)	PASS
(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	PASS
	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 w/ inactive MBR shadowing: Write with this range; Response - Pass	PASS
(D4-3-2-4-1(2)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (range2); Response - Abort/Pass(if rangeCrossing=1/0)	PASS
(D4-3-2-4-1(2)) WrLockEnabled/Locked=0/1 w/ inactive MBR shadowing: Write with multiple ranges (globalRange); Response - Abort/Pass(if rangeCrossing=1/0)	PASS
(D4 0 2 4 1(2)) WESTERN CONTROL OF WITH MICHIGAN CONTROL OF WASTERN CO	
(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	PASS
(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 (D4-3-2-4-3) WrLockEnabled/Locked=0/1: Locked bit = 0/1 in Level O Discovery (unlocked-write and unlocked-read on other ranges)	PASS 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)	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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value	PASS 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)	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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value	PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value	PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect	PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass	PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done value; Get() retrieves the values indicated by Set()	PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass	PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done value; Get() retrieves the values indicated by Set()	PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done value; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0)	PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done value; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/LockEnabled = 0/1)	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Set Enable/Done value; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0/1) (D5-1-2-2-3) Enable/Done=1: Read command: pass (WriteLockEnabled/WriteLockenabled)	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done value; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/LockEnabled = 0/1)	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Set Enable/Done value; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0/1) (D5-1-2-2-3) Enable/Done=1: Read command: pass (WriteLockEnabled/WriteLockenabled)	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done value; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (ReadLockEnabled/ReadLocked = 0/1) (D5-1-2-2-3) Enable/Done=1: Read command: pass (WriteLockEnabled/WriteLocked = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done value; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (ReadLockEnabled/ReadLocked = 0/1) (D5-1-2-2-2) Enable/Done=1: Read command: pass (WriteLockEnabled/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done value; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/LockEnabled/ReadLocked = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done value; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (ReadLockEnabled/ReadLocked = 0/1) (D5-1-2-2-2) Enable/Done=1: Read command: pass (WriteLockEnabled/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done value; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/LockEnabled/ReadLocked = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Donevalue; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled/WriteLocked = 0/1) (D5-1-2-2-2) Enable/Done=1: Read command: pass (WriteLockEnabled/WriteLocked = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read command: abort (WriteLocked = 1)	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Donevalue; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Write command: pass (Read/WriteLockEnabled/ReadLocked = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Donevalue; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled/WriteLocked = 0/1) (D5-1-2-2-2) Enable/Done=1: Read command: pass (WriteLockEnabled/WriteLocked = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read command: abort (WriteLocked = 1)	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Donevalue; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Write command: pass (Read/WriteLockEnabled/ReadLocked = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done value; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (ReadlockEnabled/ReadLocked = 0/1) (D5-1-2-2-2) Enable/Done=1: Read command: pass (WriteLockEnabled/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read command: abort (ReadLocked = 1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write Command:	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Set Enable/Done = 1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled/ReadLocked = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with writeLocked = 1) (D5-1-2-2-2) Enable/Done=1: MBRDone' bit = 1 from Level0_Discovery (D5-1-2-2-2) Enable/Done=1: MBRDone' bit = 1 from Level0_Discovery (D5-1-2-2-2) Enable/Done=1: Read with ReadLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abort (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abor	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled/WriteLocked=0/1) (D5-1-2-2-2) Enable/Done=1: Read command: pass (WriteLockEnabled/WriteLocked=0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (ReadelLocked = 1) (D5-1-2-2-2) Enable/Done=1: Write with writeLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abort (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on mult	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Set Enable/Done = 1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled/ReadLocked = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with writeLocked = 1) (D5-1-2-2-2) Enable/Done=1: MBRDone' bit = 1 from Level0_Discovery (D5-1-2-2-2) Enable/Done=1: MBRDone' bit = 1 from Level0_Discovery (D5-1-2-2-2) Enable/Done=1: Read with ReadLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abort (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abor	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0/1) (D5-1-2-2-3) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with wite with wite locked = 1) (D5-1-2-2-2) Enable/Done=1: Write with wite locked = 1) (D5-1-2-2-2) Enable/Done=1: Write with witeLocked = 1) (D5-1-2-2-3) Enable/Done=1: Write command: abort (WriteLocked = 1) (D5-1-2-2-3) Enable/Done=1: Write Read with ReadLocked = 1/Mixed on multiple ranges (Range2): abort (D5-1-2-2-3) Enable/Done=1: Read with ReadLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-1-2-2-3) Enable/Done=1: Read with ReadLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done = Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0/1) (D5-1-2-2-2) Enable/Done=1: Read command: pass (WriteLockEnabled/WriteLockEnabled = 0/1) (D5-1-2-2-3) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write ommand: abort (WriteLocked = 1) (D5-1-2-2-4(2)) Enable/Done=1: Write ommand: abort (WriteLocked = 1) (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abort (D5-1-2-2-3) Enable/Done=1: Read with ReadLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-1-2-2-3) Enable/Done=1: Write with	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done value; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with writeLockEnabled/ReadLocked = 1) (D5-1-2-2-2) Enable/Done=1: Write command: abort (WriteLocked = 1) (D5-1-2-2-2) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abort (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done = Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0/1) (D5-1-2-2-2) Enable/Done=1: Read command: pass (WriteLockEnabled/WriteLockEnabled = 0/1) (D5-1-2-2-3) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write ommand: abort (WriteLocked = 1) (D5-1-2-2-4(2)) Enable/Done=1: Write ommand: abort (WriteLocked = 1) (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abort (D5-1-2-2-3) Enable/Done=1: Read with ReadLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-1-2-2-3) Enable/Done=1: Write with	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done value; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with writeLockEnabled/ReadLocked = 1) (D5-1-2-2-2) Enable/Done=1: Write command: abort (WriteLocked = 1) (D5-1-2-2-2) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abort (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange	PASS PASS PASS PASS PASS PASS PASS PASS
(D4-3-2-4-3) Wrl.ockEnabled/Locked=0/1: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges) (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 (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Set Enable/Done-1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done-1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done-1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done-1: Write command: pass (WriteLockEnabled/MeadLocked = 0/1) (D5-1-2-2-3) Enable/Done-1: Read command: pass (WriteLockEnabled/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done-1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done-1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done-1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done-1: Write writh multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done-1: Write writh multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done-1: Write writh multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done-1: Write writh multiple ranges (globalRange): abort (D5-1-2-2-3) Enable/Done-1: Write writh writeLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abort (D5-1-2-2-3) Enable/Done-1: Write writh WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-1-2-2-3) Enable/Done-1: Write writh WriteLockE	PASS PASS PASS PASS PASS PASS PASS PASS
(D4-3-2-4-3) WrltockEnabled/Locked=0/1: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges) (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 (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Set Enable/Done value; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/LockEnabled/ReadLocked = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (Read-Locked = 1) (D5-1-2-2-2) Enable/Done=1: Write with WriteLockEnabled/Read-Locked = 1/Mixed on multiple ranges (Range2): abort (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/Read-Locked = 1/Mixed on multiple ranges (globalRange): abort (D5-1-2-2-3) Enable/Done=1: Write with	PASS PASS PASS PASS PASS PASS PASS PASS
(D4-3-2-4-3) WrlockEnabled/Locked=0/1: Locked bit = 0/1 in Level O Discovery (unlocked-write and unlocked-read on other ranges) (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 (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done-1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with writeLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abort (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-1-2-3-3) Enable/Don	PASS PASS PASS PASS PASS PASS PASS PASS
(D4-3-2-4-3) WrltockEnabled/Locked=0/1: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges) (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 (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Set Enable/Done value; Get() retrieves the values indicated by Set() (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Read command: pass (Read/LockEnabled/ReadLocked = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write with multiple ranges (Read-Locked = 1) (D5-1-2-2-2) Enable/Done=1: Write with WriteLockEnabled/Read-Locked = 1/Mixed on multiple ranges (Range2): abort (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/Read-Locked = 1/Mixed on multiple ranges (globalRange): abort (D5-1-2-2-3) Enable/Done=1: Write with	PASS PASS PASS PASS PASS PASS PASS PASS
(D4-3-2-4-3) WritockEnabled/Locked=0/1: Locked bit = 0/1 in Level O Discovery (unlocked-write and unlocked-read on other ranges) (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 (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done = Irue (01h); Response - Pass (D5-1-2-2-1) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0/1) (D5-1-2-2-2) Enable/Done=1: Write write panges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-2) Enable/Done=1: Write write multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write writh multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write writh multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write writh multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write writh multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-4) Enable/Done=1: Write writh multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write writh WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abort (D5-1-2-2-3) Enable/Done=1: Write writh WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-1-2-3-3) Enable/Done=1: Write writh WriteLockEnabled/ReadLocked = 1/Mixed on	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done-1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0/1) (D5-1-2-2-3) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-4) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-4) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-4) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-4) Enable/Done=1: Write with WriteLockEnabled = 1) (D5-1-2-2-4) Enable/Done=1: Write with WriteLockEnabled = 1) (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled = 1) (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-1-2-3-3) Enable/Done=1/0: Write addressing ONLY LBA covere	PASS PASS PASS PASS PASS PASS PASS PASS
(D4-3-2-4-3) WritockEnabled/Locked=0/1: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges) (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 (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-12-1-1) Set Enable/Done = True (01h); Response - Pass (D5-12-1-1) Get Enable/Done value; Get() retrieves the values indicated by Set() (D5-12-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-12-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-12-2-3) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0/1) (D5-12-2-3) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-3) Enable/Done=1: Write with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-3) Enable/Done=1: Write command: abort (WriteLocked = 1) (D5-12-2-4(2)) Enable/Done=1: Write command: abort (WriteLocked = 1) (D5-12-2-3) Enable/Done=1: Write Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abort (D5-12-2-3) Enable/Done=1: Read with ReadLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-12-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-12-3-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-12-3-3) E	PASS PASS PASS PASS PASS PASS PASS 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) (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Get Enable/Done-1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-3) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0/1) (D5-1-2-2-3) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-4) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-4) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-4) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-4) Enable/Done=1: Write with WriteLockEnabled = 1) (D5-1-2-2-4) Enable/Done=1: Write with WriteLockEnabled = 1) (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled = 1) (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-1-2-3-3) Enable/Done=1/0: Write addressing ONLY LBA covere	PASS PASS PASS PASS PASS PASS PASS PASS
(D4-3-2-4-3) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (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 (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl. Set() Grammar and Effect (D5-12-1-1) Set Enable/Done = True (D1h); Response - Pass (D5-12-1-1) Get Enable/Done value; Get() retrieves the values indicated by Set() (D5-12-2-2) Enable/Done-1: Read command: pass (Read/WriteLockEnabled = 0) (D5-12-2-2) Enable/Done-1: Write command: pass (Read/WriteLockEnabled = 0) (D5-12-2-2) Enable/Done-1: Write command: pass (Read/WriteLockEnabled = 0/1) (D5-12-2-2) Enable/Done-1: Write write multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done-1: Write writh multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done-1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done-1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done-1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done-1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done-1: Write with writeLockEnabled (Dscovery (D5-12-2-2) Enable/Done-1: Write with writeLockEnabled (Dscovery (D5-12-2-2) Enable/Done-1: Write with WriteLockEnabled/ReadLocked = 1) (D5-12-2-2) Enable/Done-1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-12-2-3) Enable/Done-1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-12-3-3) Enable/Done-1: Read with ReadLoc	PASS PASS PASS PASS PASS PASS PASS PASS
(D4-3-2-4-3) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl. Set() Grammar and Effect (D5-12-1-1) Set Enable/Done = True (D1h); Response - Pass (D5-12-11) Get Enable/Done value; Get() retrieves the values indicated by Set() (D5-12-2-2) Enable/Done-1: Read command: pass (Read/WriteLockEnabled = 0) (D5-12-2-2) Enable/Done-1: Write command: pass (Read/WriteLockEnabled = 0) (D5-12-2-2) Enable/Done-1: Write command: pass (Read/WriteLockEnabled = 0-1) (D5-12-2-2) Enable/Done-1: Read original pass (Read/WriteLockEnabled = 0-1) (D5-12-2-2) Enable/Done-1: Read with multiple ranges (galopale): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done-1: Read with multiple ranges (galopale): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done-1: Read with multiple ranges (galopalenage): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done-1: Read command: abort (Read/Locked = 1) (D5-12-2-2) Enable/Done-1: Write with write University enables (BobalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done-1: Write command: abort (WriteLocked = 1) (D5-12-2-2) Enable/Done-1: Write command: abort (WriteLocked = 1) (D5-12-2-2) Enable/Done-1: Write write WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abort (D5-12-2-2) Enable/Done-1: Write writh WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-12-2-3) Enable/Done-1: Write writh WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-12-3-3) Enable/Done-1/0: Write addressing DNLY LBA covered by MBR table; MBR data returned (D5-12-3	PASS PASS PASS PASS PASS PASS PASS PASS
(D4-3-2-4.3) Wrt.ockEnabled/Locked=0/1: Locked bit = 0/1 in Levet 0 Discovery (unlocked-write and unlocked-read on other ranges) (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 (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-12-1-1) Set Enable/Done = True (01h); Response - Pass (D5-12-1-1) Set Enable/Done value; Cett) retrieves the values indicated by Set() (D5-12-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-12-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-12-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0/1) (D5-12-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled April (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with writeLockEnabled of Pass of Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abort (D5-12-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-12-2-3) Enable/Do	PASS PASS PASS PASS PASS PASS PASS PASS
(D4-3-2-4-3) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-1) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl. Set() Grammar and Effect (D5-12-1-1) Set Enable/Done = True (D1h); Response - Pass (D5-12-11) Get Enable/Done value; Get() retrieves the values indicated by Set() (D5-12-2-2) Enable/Done-1: Read command: pass (Read/WriteLockEnabled = 0) (D5-12-2-2) Enable/Done-1: Write command: pass (Read/WriteLockEnabled = 0) (D5-12-2-2) Enable/Done-1: Write command: pass (Read/WriteLockEnabled = 0-1) (D5-12-2-2) Enable/Done-1: Read original pass (Read/WriteLockEnabled = 0-1) (D5-12-2-2) Enable/Done-1: Read with multiple ranges (galopale): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done-1: Read with multiple ranges (galopale): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done-1: Read with multiple ranges (galopalenage): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done-1: Read command: abort (Read/Locked = 1) (D5-12-2-2) Enable/Done-1: Write with write University enables (BobalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done-1: Write command: abort (WriteLocked = 1) (D5-12-2-2) Enable/Done-1: Write command: abort (WriteLocked = 1) (D5-12-2-2) Enable/Done-1: Write write WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abort (D5-12-2-2) Enable/Done-1: Write writh WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-12-2-3) Enable/Done-1: Write writh WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-12-3-3) Enable/Done-1/0: Write addressing DNLY LBA covered by MBR table; MBR data returned (D5-12-3	PASS PASS PASS PASS PASS PASS PASS PASS
(D4-3-2-4.3) Wrt.ockEnabled/Locked=0/1: Locked bit = 0/1 in Levet 0 Discovery (unlocked-write and unlocked-read on other ranges) (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 (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-12-1-1) Set Enable/Done = True (01h); Response - Pass (D5-12-1-1) Set Enable/Done value; Cett) retrieves the values indicated by Set() (D5-12-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-12-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-12-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0/1) (D5-12-2-2) Enable/Done=1: Read command: pass (Read/WriteLockEnabled April (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with writeLockEnabled of Pass of Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abort (D5-12-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-12-2-3) Enable/Do	PASS PASS PASS PASS PASS PASS PASS PASS
(D4-3-2-4-3) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and entTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and entTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and entTran's status=1; The value changes back to the original value (D5-12-1-1) Set Enable/Done = True (01h); Response - Pass (D5-12-1-1) Set Enable/Done = True (01h); Response - Pass (D5-12-1-1) Set Enable/Done = True (01h); Response - Pass (D5-12-1-1) Set Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-12-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-12-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-12-2-2) Enable/Done=1: Write command: pass (MetaOckEnabled/ReadLocked = 0/1) (D5-12-2-2) Enable/Done=1: Read command: pass (WriteLockEnabled/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-2) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1) (D5-12-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1) (D5-12-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1) (Mixed on multiple ranges (globalRange): abort (D5-12-3-2) Enable/Done=1: Write	PASS PASS PASS PASS PASS PASS PASS PASS
(D4-3-4-3) Wht LockErfect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (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 (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value (D5-12-1-1) Set Enable/Done = True (01h); Response - Pass (D5-12-1-1) Set Enable/Done = True (01h); Response - Pass (D5-12-1-1) Set Enable/Done = True (01h); Response - Pass (D5-12-1-1) Set Enable/Done = True (01h); Response - Pass (D5-12-2-1) Enable/Done = True (01h); Response - Pass (Read/WriteLockEnabled = 0) (D5-12-2-3) Enable/Done = True (01h); Response - Pass (Read/WriteLockEnabled = 0) (D5-12-2-3) Enable/Done = True (01h); Response - Pass (Read/WriteLockEnabled = 0) (D5-12-2-3) Enable/Done = True Read command: pass (WriteLockEnabled Area (D6-12-2-3) Enable/Done = True and thin multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-3) Enable/Done = True write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-3) Enable/Done = True Read command: abort (Read/Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-3) Enable/Done = True Read command: abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-12-2-4) Enable/Done = True Read command: abort (WriteLocked = 1) (D5-12-2-4) Enable/Done = True Read command: abort (WriteLocked = 1) (D5-12-2-2) Enable/Done = True Read command: abort (WriteLocked = 1) (D5-12-2-2) Enable/Done = True Read write Read/Read/Read/Read/Read/Read/Read/Read/	PASS PASS PASS PASS PASS PASS PASS PASS
(D4-3-2-4.3) WrtockEnabled/Locked=0/1: Locked bit = 0/1 in Level 0 Discovery (unlocked-write and unlocked-read on other ranges) (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=0; The value retains the set value (D4-3-3-1-2) WriteLock Effect in Trans: Set WriteLockEnabled in a transaction and endTran's status=1; The value changes back to the original value D5: MBRControl.Set() Grammar and Effect (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Set Enable/Done = True (01h); Response - Pass (D5-1-2-1-1) Set Enable/Done=1: Read command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled = 0) (D5-1-2-2-2) Enable/Done=1: Write command: pass (Read/WriteLockEnabled Pol) (D5-1-2-2-2) Enable/Done=1: Write command: pass (WriteLockEnabled Averaged LockEnabled = 0/1) (D5-1-2-2-2) Enable/Done=1: Read with multiple ranges (grange2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (grange2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1) (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1) (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1) (D5-1-2-2-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort (D5-1-2-3-3) Enable/Done=1: Write with WriteLockEnabled/ReadLocked	PASS PASS PASS PASS PASS PASS PASS PASS

(D5-1-2-4-1) Enable/Done=0/0: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1) (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 (ReadLocked = 1) (D5-1-2-4-2) Enable/Done=0/0: Write command: fail (WriteLocked = 1) (D5-1-2-4-3) Enable/Done=0/0: MBRDone* bit = 0 from Level0, Discovery	PASS
(D5-1-2-4-1) Enable/Done=0/0: Read command: fail (ReadLocked = 1) (D5-1-2-4-2) Enable/Done=0/0: Write command: fail (WriteLocked = 1)	
(D5-1-2-4-1) Enable/Done=0/0: Read command: fail (ReadLocked = 1) (D5-1-2-4-2) Enable/Done=0/0: Write command: fail (WriteLocked = 1)	PASS
(D5-1-2-4-2) Enable/Done=0/0: Write command: fail (WriteLocked = 1)	PASS
·	
(D5-1-2-4-3) Enable/Done=0/0: 'MBRDone' bit = 0 from Level0 Discovery	PASS
	PASS
(D5-1-2-4-3(2)) Enable/Done=0/0: 'MBREnable' bit = 0 from Level0_Discovery	PASS
Enable/Done=0/0: Read with ReadLockEnabled/ReadLocked= 1/Mixed on multiple ranges (Range2): abort	PASS
Enable/Done=0/0: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abort	PASS
Enable/Done=0/0: Read with ReadLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort	PASS
Enable/Done=0/0: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort	PASS
(D5-1-2-4-1) Enable/Done=0/1: Read command: pass (Read/WriteLockEnabled = 0)	PASS
(D5-1-2-4-2) Enable/Done=0/1: Write command: pass (Read/WriteLockEnabled = 0)	PASS
(D5-1-2-4-1) Enable/Done=0/1: Read with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	PASS
(D5-1-2-4-2) Enable/Done=0/1: Write with multiple ranges (range2): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	PASS
(D5-1-2-4-1) Enable/Done=0/1: Read with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	PASS
(D5-1-2-4-2) Enable/Done=0/1: Write with multiple ranges (globalRange): pass/abort (Read/WriteLockEnabled = 0: rangeCross = 0/1)	PASS
(D5-1-2-4-1) Enable/Done=0/1: Read command: fail (ReadLocked = 1)	PASS
(D5-1-2-4-2) Enable/Done=0/1: Write command: fail (WriteLocked = 1)	PASS
(D5-1-2-4-3) Enable/Done=0/1: 'MBRDone' bit = 0 from Level0_Discovery	PASS
(D5-1-2-4-3(2)) Enable/Done=0/1: 'MBREnable' bit = 0 from Level0_Discovery	PASS
Enable/Done=0/1: Read with ReadLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abort	PASS
Enable/Done=0/1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (Range2): abort	PASS
Enable/Done=0/1: Read with ReadLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort	PASS
Enable/Done=0/1: Write with WriteLockEnabled/ReadLocked = 1/Mixed on multiple ranges (globalRange): abort	PASS
(D5-1-3-1-1) Set 'Enable' = 1 in a transaction and endTransaction status = 0; The value retains the set value	PASS
(D5-1-3-1-2) Set 'Enable' = 0 in a transaction and endTransaction status = 1; The value changes back to the original value	PASS
D6: MBR.Set() Grammar and Effect	PASS
"	
(D6-1-1-1-1) Set data into MBR table; Response - Pass	PASS
(D6-1-1-1-1) Get data from MBR table; Compare data - Matching	PASS
(D6-1-1-1-1(2)) Read commands will retrieve MBR data - Pass	PASS
(D6-1-2-1-1) Set data to MBR table in a transaction with endTransaction status = 0; The data retains the set value	PASS
(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	PASS
D7: DataStore.Set() -Basic Grammar and Effect	PASS
(D7-1-1-1-1) 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 value	PASS
(D7-1-2-1-2) Datastore. Set in a transaction with endTransaction status = 1; The data changes back to the original value	PASS
(D7-1-2-1-2) Datastore. Set in a transaction with enumansaction status - 1, The data changes back to the original value	1 A33
D8: GenKey() Effect check	PASS
(D8-1-1-1-1) GenKey Grammar: Request with rigth parameter; Response - pass	PASS
(D8-1-2-1-1) GenKey Effect: The media encryption key used to encrypt/decrypt user data changes	PASS
(D8-1-3-1-1) GenKey Effect in a transaction with endTransaction status = 0; The range's media encryption key changes	PASS
(D8-1-3-1-2) GenKey Effect in a transaction with endTransaction status = 1; The range's media encryption key backs to the value before	PASS
200 Anti-man () Effect a hard	B400
	PASS
	PASS N/A
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail)	
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass	N/A N/A
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockignSP.Activate() Conditon: Activate to LockingSP; Response - Pass	N/A N/A PASS
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockignSP.Activate() Conditon: Activate to LockingSP; Response - Pass	N/A N/A
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockignSP.Activate() Conditon: Activate to LockingSP; Response - 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 N/A PASS
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockignSP.Activate() Conditon: Activate to LockingSP; Response - 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 (D9-1-3-1-2) LockignSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery	N/A N/A PASS N/A PASS
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1) LockignSP-Activate() Condition: Activate to LockingSP; Response - 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 (D9-1-3-1-2) LockignSP-Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (D9-1-3-1-3) LockignSP-Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table	N/A N/A PASS N/A PASS PASS
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockignSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockignSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 83 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockignSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (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	N/A N/A PASS N/A PASS PASS PASS
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockignSP.Activate() Condition: Activate to LockingSP; Response - 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 (D9-1-3-1-2) LockignSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (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	N/A N/A PASS N/A PASS PASS
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockignSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockignSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 83 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockignSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (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 and compare data - matching	N/A N/A PASS N/A PASS PASS PASS PASS
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockingSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockingSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 85 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP.Activate() Effect: LockingEnabled bit = 1 from Levelo_Discovery (D9-1-3-1-3) LockingSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockingSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockingSP.Activate() Effect: Read and compare data - matching (D9-1-3-2-1) LockingSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN	N/A N/A PASS N/A PASS PASS PASS PASS
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockingSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockingSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 85 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP.Activate() Effect: LockingEnabled bit = 1 from Levelo_Discovery (D9-1-3-1-3) LockingSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockingSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockingSP.Activate() Effect: Read and compare data - matching (D9-1-3-2-1) LockingSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN	N/A N/A PASS N/A PASS PASS PASS PASS
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1) LockingSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockingSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 85 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP.Activate() Effect: LockingEnabled bit = 1 from Level0. Discovery (D9-1-3-1-3) LockingSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockingSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockingSP.Activate() Effect: Read and compare data - matching (D9-1-3-2-1) LockingSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockingSP.Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change	N/A N/A PASS N/A PASS PASS PASS PASS PASS
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockignSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockignSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 83 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockignSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (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 and compare data - matching (D9-1-3-2-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change	N/A N/A PASS N/A PASS PASS PASS PASS
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockignSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockignSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 83 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockignSP.Activate() Effect: LockingEnabled bit = 1 from Levelo_Discovery (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 and compare data - matching (D9-1-3-2-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change	N/A N/A PASS N/A PASS PASS PASS PASS PASS
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockingSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockingSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 83 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (D9-1-3-1-3) LockingSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockingSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockingSP.Activate() Effect: Read and compare data - matching (D9-1-3-2-1) LockingSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockingSP.Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockingSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockingSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 83 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (D9-1-3-1-3) LockingSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockingSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockingSP.Activate() Effect: Read and compare data - matching (D9-1-3-1-5) LockingSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockingSP.Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change 2010: AdminSP.Revert() Effect check (D10-1-1-1-1) 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 aborted	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockingSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockingSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 83 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (D9-1-3-1-3) LockingSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockingSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockingSP.Activate() Effect: Read and compare data - matching (D9-1-3-1-5) LockingSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockingSP.Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change 2010: AdminSP.Revert() Effect check (D10-1-1-1-1) 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 aborted	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockingSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockingSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 83 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP.Activate() Effect: LockingEnabled bit = 1 from Levelo_Discovery (D9-1-3-1-3) LockingSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockingSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockingSP.Activate() Effect: Read and compare data - matching (D9-1-3-2-1) LockingSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockingSP.Activate() Effect: LockingSP in mfg state - 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 aborted (D10-1-2-1-2) AdminSP.Revert Effect: for ATA devices: check bit1 of word 82; bit1 of word 85; word 89; 90; 128	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1) LockignSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockignSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 83 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockignSP.Activate() Effect: LockingEnabled bit = 1 from LevelO_Discovery (D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StatSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Read and compare data - matching (D9-1-3-2-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change 100: 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 aborted (D10-1-2-1-2) AdminSP.Revert Effect: LockingEnabled bit = 0 from LevelO_Discovery	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockignSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockignSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 83 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockignSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (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 and compare data - matching (D9-1-3-2-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change D10: AdminSP.Revert() Effect check (D10-1-1-1-1) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be aborted (D10-1-2-1-1) AdminSP.Revert Effect: Tor ATA devices: check bit1 of word 82; bit1 of word 85; word 89; 90; 128 (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured/Inactivate)	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockingSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockingSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 83 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (D9-1-3-1-3) LockingSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockingSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockingSP.Activate() Effect: Read and compare data - matching (D9-1-3-2-1) LockingSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockingSP.Activate() Effect: LockingSP in mfg state - 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 (D10-1-2-1-1) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be aborted (D10-1-2-1-2) AdminSP.Revert Effect: LockingEnabled bit = 0 from Level0_Discovery (D10-1-2-1-3) AdminSP.Revert Effect: LockingSP is in OFS(Manufactured/Manufactured-Inactivate) (D10-1-2-1-4) AdminSP.Revert Effect: The state of LockingSP; SyncSession - Status Code: != 0 or no data returned	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockignSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockignSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 83 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockignSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (D9-1-3-1-2) 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 and compare data - matching (D9-1-3-2-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change D10: AdminSP.Revert() Effect check (D10-1-1-1-1) 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 aborted (D10-1-2-1-2) AdminSP.Revert Effect: LockingEnabled bit = 0 from Level0_Discovery (D10-1-2-1-3) AdminSP.Revert Effect: LockingEnabled bit = 0 from Level0_Discovery (D10-1-2-1-4) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured-Inactivate) (D10-1-2-1-5) AdminSP.Revert Effect: The state of LockingSP; SyncSession - Status Code: != 0 or no data returned	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockingSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockingSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 83 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP.Activate() Effect: LockingEnabled bit = 1 from Levelo_Discovery (D9-1-3-1-3) LockingSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockingSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockingSP.Activate() Effect: Read and compare data - matching (D9-1-3-1-5) LockingSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockingSP.Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change D10: AdminSP.Revert() Effect check (D10-1-2-1-1) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be aborted (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-Inactivate) (D10-1-2-1-4) 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: != 0 or no data returned (D10-1-2-1-6) AdminSP.Revert Effect: StartSession on AdminSP with MSID's PIN; SyncSession - pass	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1) LockignSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockignSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 83 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockignSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (D9-1-3-1-3) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP.Activate() Effect: StattSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP.Activate() Effect: Read and compare data - matching (D9-1-3-2-1) LockignSP.Activate() Effect: LockingSP in mfg.inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change D10: AdminSP.Revert() Effect check (D10-1-1-1-1) 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 aborted (D10-1-2-1-2) AdminSP.Revert Effect: Tor ATA devices: check bit1 of word 82; bit1 of word 85; word 89; 90; 128 (D10-1-2-1-3) AdminSP.Revert Effect: LockingEnabled bit = 0 from Level0_Discovery (D10-1-2-1-4) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured-Inactivate) (D10-1-2-1-5) AdminSP.Revert Effect: StartSession on AdminSP with MSID's PIN; SyncSession - pass (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: StartSession on AdminSP with MSID's PIN; SyncSession - pass (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: StartSession on AdminSP with MSID's PIN; SyncSession - pass	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1) LockingSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockingSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 83 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (D9-1-3-1-3) LockingSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockingSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockingSP.Activate() Effect: Read and compare data - matching (D9-1-3-1-5) LockingSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-2-1) LockingSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change 100: AdminSP.Revert() Effect check (D10-1-3-1-1) 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 aborted (D10-1-2-1-2) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured-Inactivate) (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured-Inactivate) (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: StartSession on AdminSP with MSID's PIN; SyncSession - pass (D10-1-2-3-1) AdminSP.Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in nactive: Read and compare data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in nactive: Read and compare data - matching	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockignSP.Activate() Condition: Activate to LockingSP; Response - 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 (D9-1-3-1-2) LockignSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (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 and compare data - matching (D9-1-3-1-5) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (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 within the AdminSP.Revert() was issued shall be aborted (D10-1-2-1-1) AdminSP.Revert Effect: Tor ATA devices: check bit1 of word 82; bit1 of word 85; word 89; 90; 128 (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured-Inactivate) (D10-1-2-1-4) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-3) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-3) AdminSP.Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-2-2) AdminSP.Revert Effect: LockingSP in active: Read and compare bata - matching (D10-1-2-2-2) AdminSP.Revert Effect: LockingSP in active: Read data - Pass with data mismatching/Fail (D10-1-2-2-2) AdminSP.Revert Effect: LockingSP in active: Read bata - bata bata bata be	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockignSP.Activate() Condition: Activate to LockingSP; Response - 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 (D9-1-3-1-2) LockignSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (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 and compare data - matching (D9-1-3-1-5) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (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 within the AdminSP.Revert() was issued shall be aborted (D10-1-2-1-1) AdminSP.Revert Effect: Tor ATA devices: check bit1 of word 82; bit1 of word 85; word 89; 90; 128 (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured-Inactivate) (D10-1-2-1-4) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-3) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-3) AdminSP.Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-2-2) AdminSP.Revert Effect: LockingSP in active: Read and compare bata - matching (D10-1-2-2-2) AdminSP.Revert Effect: LockingSP in active: Read data - Pass with data mismatching/Fail (D10-1-2-2-2) AdminSP.Revert Effect: LockingSP in active: Read bata - bata bata bata be	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockingSP.Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockingSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 85 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (D9-1-3-1-3) LockingSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockingSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockingSP.Activate() Effect: Read and compare data - matching (D9-1-3-1-5) LockingSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockingSP.Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change 100: AdminSP.Revert() Effect check (D10-1-1-1-1) AdminSP.Revert Effect: The session within the AdminSP, Revert () was issued shall be aborted (D10-1-2-1-1) AdminSP.Revert Effect: Tor ATA devices: check bit1 of word 82; bit1 of word 85; word 89; 90; 128 (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured-Inactivate) (D10-1-2-1-4) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-6) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-3) AdminSP.Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-2-2) AdminSP.Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-2-2) AdminSP.Revert Effect: LockingSP in active: Read and compare bata - matching (D10-1-2-2-2) AdminSP.Revert Effect: LockingSP in active: Read data - Pass with data mismatching/Fail (D10-1-2-2-2) AdminSP.Revert Effect: LockingSP in active: Read bata - Bata Swith data m	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockignSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 85 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-1) LockignSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (D9-1-3-1-2) LockignSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (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: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-2-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect check (D10-1-2-1-1) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be aborted (D10-1-2-1-1) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be aborted (D10-1-2-1-2) AdminSP.Revert Effect: LockingEnabled bit = 0 from Level0_Discovery (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured-Inactivate) (D10-1-2-1-4) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-3) AdminSP.Revert Effect: StartSession on AdminSP with MSID's PIN; SyncSession - pass (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Read and compare data - matching (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Bead and compare data - matching (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value in OFS	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP. Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1) LockingSP. Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-2) LockingSP. Activate() Effect: Check bit 1 of word 82; bit 1 of word 85 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP. Activate() Effect: LockingEnabled bit = 1 from Level0. Discovery (D9-1-3-1-3) LockingSP. Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockingSP. Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockingSP. Activate() Effect: Read and compare data - matching (D9-1-3-2-1) LockingSP. Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockingSP. Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change D10: AdminSP. Revert() Effect check (D10-1-1-1-1) AdminSP. Revert Effect: The session within the AdminSP. Revert() was issued shall be aborted (D10-1-2-1-2) AdminSP. Revert Effect: The session within the AdminSP. Revert() was issued shall be aborted (D10-1-2-1-3) AdminSP. Revert Effect: LockingEnabled bit = 0 from Level0_Discovery (D10-1-2-1-4) 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: != 0 or no data returned (D10-1-2-1-6) AdminSP. Revert Effect: StartSession on AdminSP with MSID's PIN; SyncSession - pass (D10-1-2-2-1) AdminSP. Revert Effect: LockingSP in active: Read and compare data - matching (D10-1-2-2-2) AdminSP. Revert Effect: LockingSP in active: Read and compare data - matching (D10-1-2-2-3) AdminSP. Revert Effect: LockingSP in active: Data in DataStore table shall be the value in OFS (D10-1-2-2-3) AdminSP. Revert Effect: LockingSP in active: Data in MBR table shall be the value in OF	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP. Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1) LockingSP. Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-2) LockingSP. Activate() Effect: Check bit 1 of word 82; bit 1 of word 85 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP. Activate() Effect: LockingEnabled bit = 1 from Level0. Discovery (D9-1-3-1-3) LockingSP. Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockingSP. Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockingSP. Activate() Effect: Read and compare data - matching (D9-1-3-2-1) LockingSP. Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockingSP. Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change D10: AdminSP. Revert() Effect check (D10-1-1-1-1) AdminSP. Revert Effect: The session within the AdminSP. Revert() was issued shall be aborted (D10-1-2-1-2) AdminSP. Revert Effect: The session within the AdminSP. Revert() was issued shall be aborted (D10-1-2-1-3) AdminSP. Revert Effect: LockingEnabled bit = 0 from Level0_Discovery (D10-1-2-1-4) 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: != 0 or no data returned (D10-1-2-1-6) AdminSP. Revert Effect: StartSession on AdminSP with MSID's PIN; SyncSession - pass (D10-1-2-2-1) AdminSP. Revert Effect: LockingSP in active: Read and compare data - matching (D10-1-2-2-2) AdminSP. Revert Effect: LockingSP in active: Read and compare data - matching (D10-1-2-2-3) AdminSP. Revert Effect: LockingSP in active: Data in DataStore table shall be the value in OFS (D10-1-2-2-3) AdminSP. Revert Effect: LockingSP in active: Data in MBR table shall be the value in OF	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP. Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1) LockignSP. Activate() Condition: Activate to LockingSP; Response - 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 (D9-1-3-1-2) LockignSP. Activate() Effect: LockingEnabled bit = 1 from Level0. Discovery (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 and compare data - matching (D9-1-3-2-1) LockignSP. Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockignSP. Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change 100: AdminSP. Revert() Effect check (D10-1-1-1-1) AdminSP. Revert Effect: The session within the AdminSP. Revert() was issued shall be aborted (D10-1-2-1-2) AdminSP. Revert Effect: The session within the AdminSP. Revert() was issued shall be aborted (D10-1-2-1-3) AdminSP. Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured-Inactivate) (D10-1-2-1-4) AdminSP. Revert Effect: The state of LockingSP; sin OFS(Manufactured/Manufactured-Inactivate) (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 and compare data - matching (D10-1-2-2-1) AdminSP. Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-2-1) AdminSP. Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-2-1) AdminSP. Revert Effect: LockingSP in inactive: Bead and compare data - matching (D10-1-2-2-1) AdminSP. Revert Effect: LockingSP in inactive: Data in DataStore table shall be the value in OFS	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockignSP.Activate() Condition: Activate to LockingSP; Response - 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 (D9-1-3-1-2) LockignSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (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 and compare data - matching (D9-1-3-1-5) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (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 within the AdminSP.Revert() was issued shall be aborted (D10-1-2-1-2) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be aborted (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured-Inactivate) (D10-1-2-1-4) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-3) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-3) AdminSP.Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value in OFS (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value in OFS (D10-1-2-1-1) LockingSP.Revert Effect: The session to LockingSP; Revert response - Pass (D10-2-2-1-1-1) LockingSP.Revert Effect: The session remains open after iss	N/A N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP. Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - 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 (D9-1-3-1-2) LockignSP. Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (D9-1-3-1-3) LockignSP. Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-3) LockignSP. Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-4) LockignSP. Activate() Effect: Read and compare data - matching (D9-1-3-1-5) LockignSP. Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockignSP. Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP. Revert() Effect check (D10-1-2-1-1) AdminSP. Revert Effect: The session within the AdminSP. Revert() was issued shall be aborted (D10-1-2-1-1) AdminSP. Revert Effect: Tor ATA devices: check bit1 of word 82; bit1 of word 85; word 89; 90; 128 (D10-1-2-1-3) AdminSP. Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured-Inactivate) (D10-1-2-1-4) AdminSP. Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-3) 1, AdminSP. Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned (D10-1-2-3-3) 1, AdminSP. Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-2-3) AdminSP. Revert Effect: LockingSP in active: Read and compare data - matching (D10-1-2-2-3) AdminSP. Revert Effect: LockingSP in active: Read and compare batale be the value in OFS (D10-1-2-2-3) AdminSP. Revert Effect: LockingSP in active: Data in DataStore table shall be the value in OFS (D10-2-2-1-1) LockingSP. Revert Effect: The session remains open after issuing Locking. Revert() (D10-2-2-1-1) LockingSP. Revert Ef	N/A N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockingSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 85 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (D9-1-3-1-3) LockingSP.Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-3) LockingSP.Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-4) LockingSP.Activate() Effect: Read and compare data - matching (D9-1-3-1-5) LockingSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockingSP.Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change 100: AdminSP.Revert() Effect check (D10-1-1-1-1) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be aborted (D10-1-2-1-1) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be aborted (D10-1-2-1-3) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured-Inactivate) (D10-1-2-1-4) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-3-1) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-3-1) AdminSP.Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Read and compare data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Read and compare batale hall be the value in OFS (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value in OFS (D10-1-2-2-1) LockingSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in OFS	N/A N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP, Activate() Condition: Activate to LockingSP; Asceurity is disabled; Response - Pass (D9-1-1-1) LockingSP, Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-2) LockingSP, Activate() Effect: Check bit 1 of word 82; bit 1 of word 85 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP, Activate() Effect: LockingEnabled bit = 1 from LevelQ_Discovery (D9-1-3-1-3) LockingSP, Activate() Effect: LockingEnabled bit = 1 from LevelQ_Discovery (D9-1-3-1-3) LockingSP, Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-4) LockingSP, Activate() Effect: Read and compare data - matching (D9-1-3-2-1) LockingSP, Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockingSP, Activate() Effect: LockingSP in mfg-inative - 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 aborted (D10-1-2-1-2) AdminSP, Revert Effect: LockingEnabled bit = 0 from LevelQ_Discovery (D10-1-2-1-3) AdminSP, Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-6) AdminSP, Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-6) AdminSP, Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-1) AdminSP, Revert Effect: LockingSP in in active: Read and compare data - matching (D10-1-2-3-1) AdminSP, Revert Effect: LockingSP in active: Data in DataStore table shall be the value in OFS (D10-1-2-3) AdminSP, Revert Effect: LockingSP in active: Data in DataStore table shall be the value in OFS (D10-1-2-2-3) AdminSP, Revert Effect: LockingSP in active: Data in MBR table shall be	N/A N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP Activate() Condition: Activate to LockingSP; Ascurity is disabled; Response - Pass (D9-1-1-1-1) LockingSP Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-2) LockingSP Activate() Effect: Check bit 1 of word 82; bit 1 of word 83 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP Activate() Effect: Check bit 1 of word 82; bit 1 of word 83 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (D9-1-3-1-3) LockingSP Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockingSP Activate() Effect: Read and compare data - matching (D9-1-3-1-5) LockingSP Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockingSP Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change 100: AdminSP.Revert() Effect check (D10-1-2-1-1) 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 aborted (D10-1-2-1-2) AdminSP.Revert Effect: CockingEnabled bit = 0 from Level0_Discovery (D10-1-2-1-4) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured-Inactivate) (D10-1-2-1-4) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-6) AdminSP.Revert Effect: StartSession on AdminSP with MSID's PIN; SyncSession - pass (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Read data - Pass with data mismatching/Fail (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Read data - Pass with data mismatching/Fail (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in OFS (D10-2-2-1-1	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP-Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockingSP-Activate() Effect: Check bit 1 of word 82; bit 1 of word 85 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP-Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (D9-1-3-1-3) LockignSP-Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockignSP-Activate() Effect: Stat/Session on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockignSP-Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-1-5) LockignSP-Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (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 within the AdminSP, Revert () was issued shall be aborted (D10-1-2-1-2) AdminSP-Revert Effect: LockingEnabled bit = 0 from Level0_Discovery (D10-1-2-1-4) AdminSP-Revert Effect: LockingEnabled bit = 0 from Level0_Discovery (D10-1-2-1-4) AdminSP-Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-3) AdminSP-Revert Effect: StartSession on AdminSP with MSID's PIN; SyncSession - pass (D10-1-2-3-1) AdminSP-Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-3-1) AdminSP-Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-3-1) AdminSP-Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-3-1) AdminSP-Revert Effect: LockingSP in active: Data in DataStore table shall be the value in OFS (D10-1-2-3-1) AdminSP-Revert Effect: LockingSP in active: Data in MBR table shall be the value in OFS (D10-1-2-3-1) LockingSP-Revert Grammar: Revert Session remains open after issuing Locking Rever	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1) LockingSP Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-2) LockingSP Activate() Effect: Check bit 1 of word 82; bit 1 of word 83 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockingSP Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (D9-1-3-1-3) LockingSP Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-4) LockingSP Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-5) LockingSP Activate() Effect: Read and compare data - matching (D9-1-3-1-2) LockingSP Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockingSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP.Revert() Effect check (D10-1-2-1-1) AdminSP.Revert Grammar: Revert Session to AdminSP; Revert response - Pass (D10-1-2-1-1) AdminSP.Revert Effect: for ATA devices: check bit1 of word 82; bit1 of word 85; word 89; 90; 128 (D10-1-2-1-3) AdminSP.Revert Effect: The session within the AdminSP.Revert() was issued shall be aborted (D10-1-2-1-3) AdminSP.Revert Effect: StartSession on LockingSP is in OFS(Manufactured/Manufactured-Inactivate) (D10-1-2-1-4) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: != 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 active: Read and compare data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value in OFS D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value in OFS D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in MBR table shall be the	N/A N/A N/A N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP: Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-3-1-1) LockingSP: Activate() Condition: Activate to LockingSP; Response - Pass (D9-1-3-1-1) LockingSP: Activate() Effect: Check bit 1 of word 82; bit 1 of word 85 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockignSP: Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (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 and compare data - matching (D9-1-3-1-1) LockignSP: Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockignSP: Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change D10: AdminSP: Revert() Effect check (D10-1-2-1-1) AdminSP: Revert Effect: The session within the AdminSP; Revert response - Pass (D10-1-2-1-1) AdminSP: Revert Effect: Tor ATA devices: check bit1 of word 82; bit1 of word 85; word 89; 90; 128 (D10-1-2-1-3) AdminSP: Revert Effect: CockingEnabled bit = 0 from Level0_Discovery (D10-1-2-1-4) AdminSP: Revert Effect: StartSession on LockingSP: SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-6) AdminSP: Revert Effect: StartSession on LockingSP: SyncSession - Status Code: != 0 or no data returned (D10-1-2-1-6) AdminSP: Revert Effect: LockingSP in active: Read and compare data - matching (D10-1-2-2-1) AdminSP: Revert Effect: LockingSP in active: Read and compare data - matching (D10-1-2-2-1) AdminSP: Revert Effect: LockingSP in active: Data in DataStore table shall be the value in OFS (D10-1-2-2-3) AdminSP: Revert Effect: LockingSP in active: Data in DataStore table shall be the value in OFS (D10-2-2-1-1) LockingSP: Revert Effect: LockingSP in active: Data in MBR table shall be	N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-2) Activate to LockingSP if ATA security is enabled; Response - Status Code: 3Fh (Fail) (D9-1-2-1-1) LockingSP Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - 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 (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 (D9-1-3-1-2) LockignSP Activate() Effect: LifeCycleState = 09h of LockingSP in the SP table (D9-1-3-1-3) LockignSP Activate() Effect: StartSession on LockingSP with SID's PIN; SyncSession - pass (D9-1-3-1-4) LockignSP Activate() Effect: Read and compare data - matching (D9-1-3-1-4) LockignSP Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockignSP Activate() Effect: LockingSP in mfg state - PIN for Admin1 does not change D10: AdminSP.Revert() Effect check (D10-1-1-1-1) 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 If was issued shall be aborted (D10-1-2-1-1) AdminSP.Revert Effect: The state of LockingSP is in OFS(Manufactured/Manufactured-Inactivate) (D10-1-2-1-3) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: 1= 0 or no data returned (D10-1-2-1-6) AdminSP.Revert Effect: StartSession on LockingSP; SyncSession - Status Code: 1= 0 or no data returned (D10-1-2-1-1) AdminSP.Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-1-1) AdminSP.Revert Effect: LockingSP in active: Read and compare data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Read and compare data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Read and compare data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Read and compare data - matching (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore	N/A N/A N/A N/A PASS N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA
(D9-1-2-1-1) LockingSP.Activate() Condition: Activate to LockingSP if ATA security is disabled; Response - Pass (D9-1-1-1-1) LockignSP.Activate() Effect: Check bit 1 of word 82; bit 1 of word 85 and all bits of word 89; 90; 92; 128 = 0 (D9-1-3-1-2) LockignSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (D9-1-3-1-3) LockignSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (D9-1-3-1-3) LockignSP.Activate() Effect: LockingEnabled bit = 1 from Level0_Discovery (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 and compare data - matching (D9-1-3-1-5) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 is the same as the SID's PIN (D9-1-3-3-1) LockignSP.Activate() Effect: LockingSP in mfg-inative - PIN for Admin1 does not change (D10-1-2-1-1) AdminSP.Revert () Effect LockingSP in mfg state - PIN for Admin1 does not change (D10-1-2-1-1) 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 aborted (D10-1-2-1-2) 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 LockingSP; SyncSession - Status Code: I= 0 or no data returned (D10-1-2-3-1) AdminSP.Revert Effect: LockingSP in inactive: Read and compare data - matching (D10-1-2-2-1) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value in OFS (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value in OFS (D10-1-2-2-3) AdminSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value in OFS (D10-2-2-1-1) LockingSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in OFS (D10-2-2-	N/A N/A N/A N/A PASS N/A PASS PASS PASS PASS PASS PASS PASS PA

(D10-2-2-2-2) LockingSP.Revert Effect: LockingSP in active: Data in DataStore table shall be the value in OFS	PASS
(D10-2-2-2-3) LockingSP.Revert Effect: LockingSP in active: Data in MBR table shall be the value in OFS	PASS
D40 D 10005% 1 1 1 1	D400
D10: RevertSP() Effect check	PASS
(D10-3-1-1-1) LockingSP.RevertSP Grammar: RevertSP without parameters; RevertSP Response - Pass	PASS
(D10-3-2-1-1) LockingSP. RevertSP Condition: RevertSP if 'KeepGlobalRangeKey' = 1 and read-unlocked/write-unlocked for the Locking GlobalRange; RevertSP Res	
(D10-3-2-1-1) LockingSP.RevertSP Condition: RevertSP if "KeepGlobalRangeKey" = 1 and read-unlocked for the Locking GlobalRange; RevertSP Response - Pass	PASS
(D10-3-2-1-1) LockingSP.RevertSP Condition: RevertSP if "KeepGlobalRangeKey" = 1 and write-unlocked for the Locking GlobalRange; RevertSP Response - Pass	PASS
(D10-3-2-1-2) LockingSP.RevertSP Condition: RevertSP if 'KeepGlobalRangeKey' = 1 and read and write-locked for the Locking GlobalRange; RevertSP Response - If	
(D10-3-3-1-1) LockingSP.RevertSP Effect: The session shall be aborted: Get_Rqs for LifeCycleState after RevertSP() is successful; Get_Rsp - no data returned	PASS
(D10-3-3-1-2) LockingSP-RevertSP Effect: for ATA devices: check bit1 of word 82; bit1 of word 85; word 89; 90; 128	N/A
(D10-3-3-1-3) LockingSP.RevertSP Effect: LockingEnabled bit = 0 from Level0_Discovery	PASS
(D10-3-3-1-4) LockingSP.RevertSP Effect: LifeCycleState = 08h (Manufactured-Inactivate)	PASS
(D10-3-3-1-5) LockingSP.RevertSP Effect: StartSession on LockingSP; SyncSession - Status Code: != 0 or no data returned	PASS
(D10-3-3-2-1) LockingSP-RevertSP Effect: LockingSP in active: KeepGKey=1 and data covered by GlobalRange; Data shall not change	PASS
(D10-3-3-2-3) LockingSP.RevertSP Effect: LockingSP in active: KeepGKey=0; Data shall change	PASS
(D10-3-3-2-4) LockingSP-RevertSP Effect: LockingSP in active: Data in DataStore table shall be the value in OFS	PASS
(D10-3-3-2-5) LockingSP.RevertSP Effect: LockingSP in active: Data in MBR table shall be the value in OFS	PASS
DO DAO Astrona and Dougla ATA command should be Doubleted Occurred and by	NI/A
D9-D10 Activate and Revert: ATA command check in RestrictedCommands table (D9-1-3-1-6) RestrictedCmds: ATA command check after LockingSP.Activate	N/A N/A
,	
(D10-3-3-1-6) RestrictedCmds: ATA command check after LockingSP. RevertSP	N/A N/A
(D10-2-2-1-6) Restricted Cmds: ATA command check after Locking SP. Revert	
(D10-1-2-1-7) RestrictedCmds: ATA command check after AdminSP.Revert	N/A
D11: Power Cycle	PASS
(D11.1-1.1) C_PIN: after power cycle 1.if Persistence=1 Tries=no change; 2.if Persistence=0 Tries=0	PASS
(D11-1-1-1) RestrictedCmds: Next()/Get() to get the cell contents	N/A
(D11-2-1-1-1) Restricted Cmds: Allowed column check after power cycle	N/A
(D12 1 1 1) restricted on the state of the s	1477
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
** Opal V2.0 - Generic **	
Protocol 2 Command Test	PASS
Check Get_ComID command	PASS
Check Verify_ComID_Valid command	PASS
Check Get_ComID_Rsp command	PASS
	D400
Check SSC information Identify SSC type from the TPerInfo table	PASS PASS
Check SSC Feature Descriptor from Level 0 Discovery	PASS
Glick Stori Catalic Descriptor Holin Leveto Discovery	1 700
Verify Geometry information	PASS
Geometry Reporting Feature returned from Level0_Discovery	PASS
Contents of column 07-0Ah returned from the LockingInfo table	PASS
Verify Geometry Info between LockingInfo table and Level0_Discovery	PASS
TPer Reset Command Test	PASS
Check the support of TPer_Reset command	PASS
If TPer_Reset is disabled; Issue TPer_Reset - aborted	PASS
Enable TPer_Reset command: set ProgrammaticResetEnable=1 in the TPerInfo table	PASS
All open session SHALL be aborted on all ComID	PASS
All uncommitted transactions SHAII be aborted on all ComID	PASS
The synchronous protocol stack for all ComID SHAll be reset to its initial state	PASS
All related method processing occurring on all ComIDs SHALL be aborted	PASS
Host's communications capabilities SHAll be reset to the initial minimum assumptions	PASS
Read/WriteLocked do not change for all Locking objects if Programmatic enumeration is not in LockOnReset	PASS
Read/WriteLocked = True for all Locking objects if the LockOnReset = Programmatic enumeration value	PASS
Done does not change in MBRControl table if Programmatic enumeration is not in DoneOnReset	PASS
Done = False in MBRControl table if the DoneOnReset = Programmatic enumeration value	PASS
Check Read/WriteLocked for all Locking objects before and after TPer_Reset is disabled	PASS
Check Done in the MBRControl table before and after TPer_Reset is disabled	PASS
Stack Reset Test	PASS
Check the support of Stack_Reset command	PASS
The data returned from Stack_Reset rsponse - Get_ComID_Rsp	PASS
All open session for that ComID SHALL be aborted	PASS
All open session for that Comid SHALL be aborted All uncommitted transactions SHALL be aborted	PASS
All related method on that ComID SHALL be aborted	PASS
The protocol stack for all ComIDs SHALL be reset to its initial state	PASS
All communications properties SHALL be reset to their default values	PASS
No Response Available if no Handle_ComID_Request command preceded the Get_ComID_Response Check 'Readl ocked' and 'Writel ocked' values in Locking table.	PASS PASS
Check 'ReadLocked' and 'WriteLocked' values in Locking table Check 'Done' value in MBRControl table	PASS
Stack_Reset with non-zero reserved byte; It shall be ignored by both host and device	PASS
Casal_neess man non-zero reserved byte, it shak be ignored by both nost and device	1 100

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
Authoritania de de COD	D.100
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	PASS
Check the state of LockingSP	PASS
End Session - Request	PASS
·	
End Session - Response	PASS
Check Authenticate method	PASS
Check the support of Authenticate method - AdminSP	PASS
Authenticate - SID; Authenticate Response - Success(AuthStatus = 01h)	PASS
Authenticate - Admin1(non-authorized UID); Authenticate Response - Fail(AuthStatus = 0h)	PASS
Check the support of Authenticate method - LockingSP	PASS
Authenticate - Admin1; Authenticate Response - Success(AuthStatus = 01h)	PASS
Authenticate - User1 (authority UID); Authenticate Response - Success(AuthStatus = 01h)	PASS
Authenticate - User2 (non-authority UID); Authenticate Response - Fail(AuthStatus = 0h)	PASS
Number of authenticate attempts > MaxAuthentications; Authenticate Response - Fail(AuthStatus = 0h)	PASS
Authenticate - UserX (invalid-authority UID); Authenticate Response - StatusCode = 0Ch(Invalid Param)	PASS
Authenticate - User1 with incorrect optional param; Authenticate Response - StatusCode = 0Ch(Invalid Param)	PASS
Authorited Continual mediceroptional paramy retrieval energy of the continual continua	17100
Check Random method	PASS
Check the support of Random method - AdminSP	PASS
Random Request with count < 20h in AdminSP; Random Response - Success	PASS
	PASS
Random Request with count = 20h in AdminSP; Random Response - Success	
Random Request with count > 20h in AdminSP; Random Response - Success or StatusCode=0Ch(Invalid_Param)	PASS
Check the support of Random method - LockingSP	PASS
Random Request with count < 20h in LockingSP; Random Response - Success	PASS
Random Request with count = 20h in LockingSP; Random Response - Success	PASS
Random Request with count > 20h in LockingSP; Random Response - Success or StatusCode=0Ch(Invalid_Param)	PASS
Alignment LBA Test	PASS
RangeStart/Length: Aligned; Response - Pass	PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams)	PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams)	PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams)	PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams)	PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart!=0 and startAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength!=0 and LengthAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart!=0; RangeLength!=0 and LengthAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams)	PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) Data Alignment Restriction on Byte Table - DataStore	PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart!=0 and startAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength!=0 and LengthAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart!=0; RangeLength!=0 and LengthAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) Data Alignment Restriction on Byte Table - DataStore Get MandatoryWriteGranularity and RecommendedAccessGranularity of DataStore from Table table	PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart!=0 and startAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength!=0 and LengthAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart!=0; RangeLength!=0 and LengthAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) 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 PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart!=0 and startAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength!=0 and LengthAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart!=0; RangeLength!=0 and LengthAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) Data Alignment Restriction on Byte Table - DataStore Get MandatoryWriteGranularity and RecommendedAccessGranularity of DataStore from Table table	PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pramm)	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pramm) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pramm)	PASS PASS PASS PASS PASS PASS PASS N/A N/A
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into DataStore table; Response - Pass	PASS PASS PASS PASS PASS PASS PASS N/A N/A PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pramm) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pramm)	PASS PASS PASS PASS PASS PASS PASS N/A N/A
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into DataStore table; Response - Pass	PASS PASS PASS PASS PASS PASS PASS N/A N/A PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into DataStore table; Response - Pass	PASS PASS PASS PASS PASS PASS PASS N/A N/A PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(engthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(engthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsettMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsettMWriteGran=0 and lengthMWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) Data Alignment Restriction on Byte Table - DataStore Get MandatoryWriteGranularity and RecommendedAccessGranularity of DataStore from Table table MandatoryWriteGranularity of DataStore StALL be less than or equal to 8192 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(lengthMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0) and lengthMWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(lengthMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(lengthMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS N/A N/A PASS PASS PASS PASS NAS PASS PASS NAS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(lengthMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into MBR table; Response - Pass	PASS PASS PASS PASS PASS PASS PASS N/A N/A PASS PASS PASS PASS PASS PASS PASS PA
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0) and lengthMWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(lengthMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(lengthMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS N/A N/A PASS PASS PASS PASS NAS PASS PASS NAS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(engthMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(lengthMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into MBR table; Response - Pass	PASS PASS PASS PASS PASS PASS PASS N/A N/A PASS PASS PASS PASS PASS PASS PASS PA
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(IngthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0 and lengthMwriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(lengthMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into MBR table; Response - Pass Get and Compare data from MBR table - Matching AdminSP.Revert() Effect check	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) Data Alignment Restriction on Byte Table - DataStore Get MandatoryWriteGranularity and RecommendedAccessGranularity of DataStore from Table table MandatoryWriteGranularity of DataStore StALL be less than or equal to 8192 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0) and lengthMWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(lengthMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into MBR table; Response - Pass Get and Compare data from MBR table - Matching AdminSP.Revert() Effect check AdminSP.Revert with 'Behavior of C_PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass	PASS PASS PASS PASS PASS PASS PASS N/A N/A PASS PASS PASS PASS PASS PASS PASS PA
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(IngthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0 and lengthMwriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(lengthMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into MBR table; Response - Pass Get and Compare data from MBR table - Matching AdminSP.Revert() Effect check	PASS PASS PASS PASS PASS PASS PASS N/A N/A PASS PASS PASS PASS N/A N/A PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran!=0) and lengthMWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(lengthMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0 and length/WriteGran=0) into MBR table; Response - Pass Get and Compare data from MBR table - Matching AdminSP.Revert() Effect check AdminSP.Revert with 'Behavior of C_PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN = C_PIN_MSID and 'Initial C_PIN_SID'=0	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(lengthMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0) into MBR table; Res	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran!=0) and lengthMWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(lengthMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0 and length/WriteGran=0) into MBR table; Response - Pass Get and Compare data from MBR table - Matching AdminSP.Revert() Effect check AdminSP.Revert with 'Behavior of C_PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN = C_PIN_MSID and 'Initial C_PIN_SID'=0	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pramm) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pramm) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(lengthMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Pramm) Set data(offsetMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Pramm) Set data(offsetMWriteGran!=0) into MBR table; Response - Status Code: 0Ch(Invalid_Pramm) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into MBR table; Response - Pass Get and Compare data from MBR table - Matching AdminSP.Revert() Effect check AdminSP.Revert with 'Behavior of C_PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN = C_PIN_MSID and 'Initial C_PIN_SID =0 Data Removal Mechanism Opal 2.02: Cverwrite Data Erase or Block Erase MAY be supported	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) Data Alignment Restriction on Byte Table - DataStore Get MandatoryWriteGranularity of DataStore shalLL be less than or equal to 8192 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(lengthMWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(lengthMWriteGran=1=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=1=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into MBR table; Response - Pass Get and Compare data from MBR table - Matching AdminSP.Revert() Effect check AdminSP.Revert with "Behavior of C_PIN_SID PIN on TPer Revert"=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Beh	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran!=0) and lengthMWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(lengthMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0) and lengthMWriteGran=0) into MBR table; Response - Pass Get and Compare data from MBR table - Matching AdminSP.Revert() Effect check AdminSP.Revert() Effect check AdminSP.Revert with 'Behavior of C_PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN = C_PIN_MSID and 'Initial C_PIN_SID'=0 Data Removal Mechanism Opal 2.02: Overwrite Data Erase or Block Erase MAY be supported Opal 2.02: Overwrite Data Erase or Block Erase MAY be supported Opal 2.02: Bit3/4 and Bit6/7(Reserved) in Byte6/7 are zero	PASS PASS PASS PASS PASS PASS PASS N/A N/A PASS PASS PASS PASS PASS PASS PASS PA
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) Data Alignment Restriction on Byte Table - DataStore Get MandatoryWriteGranularity of DataStore shalLL be less than or equal to 8192 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(lengthMWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(lengthMWriteGran=1=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=1=0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0 and lengthMWriteGran=0) into MBR table; Response - Pass Get and Compare data from MBR table - Matching AdminSP.Revert() Effect check AdminSP.Revert with "Behavior of C_PIN_SID PIN on TPer Revert"=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Beh	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart !=0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran!=0) and lengthMWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(lengthMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0) into MBR table; Response - Status Code: 0Ch(Invalid_Pram) Set data(offsetMWriteGran=0) and lengthMWriteGran=0) into MBR table; Response - Pass Get and Compare data from MBR table - Matching AdminSP.Revert() Effect check AdminSP.Revert() Effect check AdminSP.Revert with 'Behavior of C_PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN = C_PIN_MSID and 'Initial C_PIN_SID'=0 Data Removal Mechanism Opal 2.02: Overwrite Data Erase or Block Erase MAY be supported Opal 2.02: Overwrite Data Erase or Block Erase MAY be supported Opal 2.02: Bit3/4 and Bit6/7(Reserved) in Byte6/7 are zero	PASS PASS PASS PASS PASS PASS PASS N/A N/A PASS PASS PASS PASS PASS PASS PASS PA
RangeStart/Length: Aligned; Response - Pass RangeStart: EnangeStart: =0 and startAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart: =0; RangeLength!=0 and LengthAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart!=0; RangeLength!=0 and LengthAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGrani-0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGrani-0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGrani-0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGrani-0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGrani-0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGrani-0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGrani-0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGrani-0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGrani-0) into MBR table; Response - Pass Get and Compare data from MBR table - Matching AdminSP.Revert() Effect check AdminSP.Revert with 'Behavior of C_PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN = C_PIN_MSID and 'Initial C_PIN_SID'=0 Data Removal Mechanism Opal 2.02: Overwrite Data Erase or Block Erase MAY be supported Opal 2.02: Crypto Erase bit SHALL be supported Opal 2.02: Crypto Erase bit SHALL be supported Opal 2.02: EtiS/A and Bit6/7(Reserved) in Byte6/7 are zero Check the support of DataRemovalMechanism table Get Request on ActiveDataRemovalMechanism of the DRM table; Get Response: Pass	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart: 1=0 and startAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart: 1=0; RangeLength: 1=0 and LengthAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart: 1=0; RangeLength: 1=0 and LengthAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthHWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Prarm) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0 and lengthHWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data[engthHWriteGran=0] into MBR stable; Response - Status Code: 0Ch(Invalid_Param) Set data[offsetMWriteGran=0] into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data[offsetMWriteGran=0] into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data[offsetMWriteGran=0] into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data[offsetMWriteGran=0] into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data[offsetMWriteGran=0] into MBR table - Maching AdminSP.Revert() Effect check AdminSP.Revert() Effect check AdminSP.Revert with 'Behavior of C_PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN = C_PIN_MSID and 'Initial C_PIN_SID'=0 Data Removal Mechanism Opal 2.02: Overwrite Data Erase or Block Erase MAY be supported Opal 2.02: Bit3/4 and Bit6/7(Reserved) in Byte6/7 are zero Check the support of DataRemovalMechanism table Get Request on ActiveDataAremovalMechanism table Get Request on ActiveDataAremovalMechanism	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: EnangeStart: =0 and startAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart: =0; RangeLength!=0 and LengthAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart!=0; RangeLength!=0 and LengthAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthMWriteGrani-0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGrani-0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGrani-0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGrani-0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGrani-0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGrani-0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGrani-0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGrani-0) into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGrani-0) into MBR table; Response - Pass Get and Compare data from MBR table - Matching AdminSP.Revert() Effect check AdminSP.Revert with 'Behavior of C_PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN = C_PIN_MSID and 'Initial C_PIN_SID'=0 Data Removal Mechanism Opal 2.02: Overwrite Data Erase or Block Erase MAY be supported Opal 2.02: Crypto Erase bit SHALL be supported Opal 2.02: Crypto Erase bit SHALL be supported Opal 2.02: EtiS/A and Bit6/7(Reserved) in Byte6/7 are zero Check the support of DataRemovalMechanism table Get Request on ActiveDataRemovalMechanism of the DRM table; Get Response: Pass	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart != 0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) Data Alignment Restriction on Byte Table - DataStore Get MandatoryWriteGranularity and RecommendedAccessGranularity of DataStore from Table table MandatoryWriteGranularity of DataStore StriALL be less than or equal to 8192 Set data[engthMWriteGranil-0] into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data[offsetMWriteGrani-0] into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data[offsetMWriteGran-0] and lengthMWriteGran-0] into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data[engthMWriteGrani-0] into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data[engthMWriteGran=0] into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data[offsetMWriteGran=0] into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data[offsetMWriteGran=0 and lengthMWriteGran=0) into MBR table; Response - Pass Get and Compare data from MBR table - Matching AdminSP.Revert() Effect check AdminSP.Revert() Effect check AdminSP.Revert with 'Behavior of C_PIN_SID PIN on TPer Revert(=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN = C_PIN_MSID and 'Initial C_PIN_SID'=0 Data Removal Mechanism Opal 2.02: Cverwrite Data Erase or Block Erase MAY be supported Opal 2.02: Cverwrite Data Erase or Block Erase MAY be supported Opal 2.02: Cverwrite Data Erase or Block Erase MAY be supported Opal	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart: 1=0 and startAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart: 1=0; RangeLength: 1=0 and LengthAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart: 1=0; RangeLength: 1=0 and LengthAlignment!=0; Response - Status Code: 0Ch(Invalid_Prams) 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 Set data(lengthHWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Prarm) Set data(offsetMWriteGran!=0) into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data(offsetMWriteGran=0 and lengthHWriteGran=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data[engthHWriteGran=0] into MBR stable; Response - Status Code: 0Ch(Invalid_Param) Set data[offsetMWriteGran=0] into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data[offsetMWriteGran=0] into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data[offsetMWriteGran=0] into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data[offsetMWriteGran=0] into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data[offsetMWriteGran=0] into MBR table - Maching AdminSP.Revert() Effect check AdminSP.Revert() Effect check AdminSP.Revert with 'Behavior of C_PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN = C_PIN_MSID and 'Initial C_PIN_SID'=0 Data Removal Mechanism Opal 2.02: Overwrite Data Erase or Block Erase MAY be supported Opal 2.02: Bit3/4 and Bit6/7(Reserved) in Byte6/7 are zero Check the support of DataRemovalMechanism table Get Request on ActiveDataAremovalMechanism table Get Request on ActiveDataAremovalMechanism	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart != 0 and startAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart =0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) RangeLength: RangeStart !=0; RangeLength !=0 and LengthAlignment !=0; Response - Status Code: 0Ch(Invalid_Prams) Data Alignment Restriction on Byte Table - DataStore Get MandatoryWriteGranularity and RecommendedAccessGranularity of DataStore from Table table MandatoryWriteGranularity of DataStore StriALL be less than or equal to 8192 Set data[engthMWriteGranil-0] into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data[offsetMWriteGrani-0] into DataStore table; Response - Status Code: 0Ch(Invalid_Param) Set data[offsetMWriteGran-0] and lengthMWriteGran-0] into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data[engthMWriteGrani-0] into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data[engthMWriteGran=0] into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data[offsetMWriteGran=0] into MBR table; Response - Status Code: 0Ch(Invalid_Param) Set data[offsetMWriteGran=0 and lengthMWriteGran=0) into MBR table; Response - Pass Get and Compare data from MBR table - Matching AdminSP.Revert() Effect check AdminSP.Revert() Effect check AdminSP.Revert with 'Behavior of C_PIN_SID PIN on TPer Revert(=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN'=0: PIN = C_PIN_MSID and 'Initial C_PIN_SID'=0 Data Removal Mechanism Opal 2.02: Cverwrite Data Erase or Block Erase MAY be supported Opal 2.02: Cverwrite Data Erase or Block Erase MAY be supported Opal 2.02: Cverwrite Data Erase or Block Erase MAY be supported Opal	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart 1-0 and startAlignment 1=0; Response - Status Code: OCh(Invalid_Prams) RangeLength: RangeStart 1-0; RangeLength 1-0 and LengthAlignment 1=0; Response - Status Code: OCh(Invalid_Prams) RangeLength: RangeStart 1-0; RangeLength 1=0 and LengthAlignment 1=0; Response - Status Code: OCh(Invalid_Prams) 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 Set data(lengthMWriteGran1=0) into DataStore table; Response - Status Code: OCh(Invalid_Pram) Set data(offsetMWriteGran1=0) into DataStore table; Response - Status Code: OCh(Invalid_Pram) Set data(offsetMWriteGran1=0) into DataStore table; Response - Status Code: OCh(Invalid_Pram) Set data(offsetMWriteGran1=0) into DataStore table; Response - Pass Get and Compare data from DataStore - Matching Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table MandatoryWriteGranularity and RecommendedAccessGranularity of MBR from Table table MandatoryWriteGran1=0) into MBR table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1=0) into MBR table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1=0) into MBR table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1=0) and lengthMWriteGran=0) into MBR table; Response - Pass Get and Compare data from MBR table - Matching AdminSP.Revert() Effect check AdminSP.Revert() Effect check AdminSP.Revert() Effect check AdminSP.Revert with 'Behavior of C_PIN_SID PIN on TPer Revert'=0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN=0: PIN = C_PIN_MSID and 'Initiat C_PIN_SID=0 Data Removal Mechanism Opal 2.02: Overwrite Data Erase or Block Erase MAY be supported Opal 2.02: Overwrite Data Erase or Block Erase MAY be supported Opal 2.02: Overwrite Data Erase or Blo	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart 1-0; and start/lignment 1-0; Response - Status Code: OCh(Invalid_Prams) RangeLength: RangeStart 1-0; RangeLength 1-0 and LengthAlignment 1-0; Response - Status Code: OCh(Invalid_Prams) RangeLength: RangeStart 1-0; RangeLength 1-0 and LengthAlignment 1-0; Response - Status Code: OCh(Invalid_Prams) 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 Set data(engthMWriteGran1-0) into DataStore stable; Response - Status Code: OCh(Invalid_Param) Set data(engthMWriteGran1-0) into DataStore table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into DataStore table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into DataStore table; Response - Status Code: OCh(Invalid_Param) Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(offsetMWriteGran1-0) into MBR table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into MBR table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into MBR table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into MBR table; Response - Status Code: OCh(Invalid_Param) AdminSP.Revert() Effect check AdminSP.Revert() Effect check AdminSP.Revert() Effect check AdminSP.Revert with 'Behavior of C_PIN_SID PIN on TPer Revert*0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN*0: PIN = C_PIN_MSID and 'Initial C_PIN_SID*0 Data Removal Mechanism Opal 2.02: Overwrite Data Erase or Block Erase MAY be supported Opal 2.02: Crypto Erase bit SHALL be supported Opal 2.02: Crypto Erase bit SHALL be supported Opal 2.02: Otypto Erase bit SHALL be supported Opal 2.02: Otypto Erase bit SHALL be supported Opal 2.02: Otypto Erase bit SHALL be SHALL be SHALL be SHALL be SHALL S	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart 1-0; and startAlignment 1-0; Response - Status Code: OCh(Invalid_Prams) RangeLength: RangeStart 1-0; RangeLength 1-0 and LengthAlignment 1-0; Response - Status Code: OCh(Invalid_Prams) RangeLength: RangeStart 1-0; RangeLength 1-0 and LengthAlignment 1-0; Response - Status Code: OCh(Invalid_Prams) 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 Set data(engthMWriteGran1-0) into DataStore stable; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into DataStore table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into DataStore table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into DataStore table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into DataStore table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into DataStore table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into MBR table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into MBR table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into MBR table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into MBR table; Response - Pass Get and Compare data from MBR table - Matching AdminSP. Revert () Effect check AdminSP. Revert (b) Effect check AdminSP. Revert with 'Behavior of C_PIN_SID PIN on TPer Revert -0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN -0: PIN = C_PIN_MSID and 'Initial C_PIN_SID'-0 Data Removal Mechanism Opal 2.02: Overwrite Data Erase or Block Erase MAY be supported Opal 2.02: Oserwite Data Erase or Block Erase MAY be supported Opal 2.02: DistA/4 and BitG/T/Reserved) in ByteG/7 are zero Check the supported ActiveDRM of the DRM table; Get Response: Statu	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart 1-0; and start/lignment 1-0; Response - Status Code: OCh(Invalid_Prams) RangeLength: RangeStart 1-0; RangeLength 1-0 and LengthAlignment 1-0; Response - Status Code: OCh(Invalid_Prams) RangeLength: RangeStart 1-0; RangeLength 1-0 and LengthAlignment 1-0; Response - Status Code: OCh(Invalid_Prams) 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 Set data(engthMWriteGran1-0) into DataStore stable; Response - Status Code: OCh(Invalid_Param) Set data(engthMWriteGran1-0) into DataStore table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into DataStore table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into DataStore table; Response - Status Code: OCh(Invalid_Param) Data Alignment Restriction on Byte Table - MBR Get MandatoryWriteGranularity of MBR SHALL be less than or equal to 8192 Set data(offsetMWriteGran1-0) into MBR table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into MBR table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into MBR table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into MBR table; Response - Status Code: OCh(Invalid_Param) AdminSP.Revert() Effect check AdminSP.Revert() Effect check AdminSP.Revert() Effect check AdminSP.Revert with 'Behavior of C_PIN_SID PIN on TPer Revert*0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN*0: PIN = C_PIN_MSID and 'Initial C_PIN_SID*0 Data Removal Mechanism Opal 2.02: Overwrite Data Erase or Block Erase MAY be supported Opal 2.02: Crypto Erase bit SHALL be supported Opal 2.02: Crypto Erase bit SHALL be supported Opal 2.02: Otypto Erase bit SHALL be supported Opal 2.02: Otypto Erase bit SHALL be supported Opal 2.02: Otypto Erase bit SHALL be SHALL be SHALL be SHALL be SHALL S	PASS PASS PASS PASS PASS PASS PASS PASS
RangeStart/Length: Aligned; Response - Pass RangeStart: RangeStart 1-0; and startAlignment 1-0; Response - Status Code: OCh(Invalid_Prams) RangeLength: RangeStart 1-0; RangeLength 1-0 and LengthAlignment 1-0; Response - Status Code: OCh(Invalid_Prams) RangeLength: RangeStart 1-0; RangeLength 1-0 and LengthAlignment 1-0; Response - Status Code: OCh(Invalid_Prams) 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 Set data(engthMWriteGran1-0) into DataStore stable; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into DataStore table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into DataStore table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into DataStore table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into DataStore table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into DataStore table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into MBR table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into MBR table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into MBR table; Response - Status Code: OCh(Invalid_Param) Set data(offsetMWriteGran1-0) into MBR table; Response - Pass Get and Compare data from MBR table - Matching AdminSP. Revert () Effect check AdminSP. Revert (b) Effect check AdminSP. Revert with 'Behavior of C_PIN_SID PIN on TPer Revert -0 or 1: Revert Response - Pass 'Behavior of C_PIN_SID PIN -0: PIN = C_PIN_MSID and 'Initial C_PIN_SID'-0 Data Removal Mechanism Opal 2.02: Overwrite Data Erase or Block Erase MAY be supported Opal 2.02: Oserwite Data Erase or Block Erase MAY be supported Opal 2.02: DistA/4 and BitG/T/Reserved) in ByteG/7 are zero Check the supported ActiveDRM of the DRM table; Get Response: Statu	PASS PASS PASS PASS PASS PASS PASS PASS

StartSession - SessionTimeout: zero with MaxSessionTimeout=0/non-zero; SyncSession - Pass/Status Code=0Ch(Invalid Param)	N/A
Session Timeout: Start/Sync Session after a session aborted due to the session timeout during traffic - Pass	PASS
Hardware Reset tests on LockOnReset/DoneOnReset	PASS
Locking table: Set Hardware Reset to 'LockOnReset' column	PASS
Locking table: Set Read/WriteLockEnabled to True and Read/WriteLocked to False	PASS
Locking table: Issue Hardware Reset	PASS
Locking table: Verify Read/WriteLocked = True after Hardware Reset	PASS
Locking table w/o Hardware Reset: Check Hardware Reset not in 'LockOnReset' column	PASS
Locking table w/o Hardware Reset: Set Read/WriteLockEnabled to True and Read/WriteLocked to False	PASS PASS
Locking table w/o Hardware Reset: Issue Hardware Reset Locking table w/o Hardware Reset: Verify Read/WriteLocked remain the same after Hardware Reset	PASS
MBRControl table: Set Hardware Reset to 'DoneOnReset' column	PASS
MBRControl table: Set Enable/Done to True/True	PASS
MBRControl table: Issue Hardware Reset	PASS
MBRControl table: Verify Done = False after Hardware Reset	PASS
MBRControl table w/o Hardware Reset: Check Hardware Reset not in 'DoneOnReset' column	PASS
MBRControl table w/o Hardware Reset: Set Enable/Done to True/True	PASS
MBRControl table w/o Hardware Reset: Issue Hardware Reset	PASS
MBRControl table w/o Hardware Reset: Verify Done remain the same after Hardware Reset	PASS
Verify 'RangeStart' and 'RangeLength' after Revert/RevertSP	PASS
Set Request on RangeStart/Length; Set Response - Pass	PASS
Revert Session to AdminSP; Revert Response - Pass	PASS
Verify RangeStart/Length on Range1/2=zero after AdminSP.Revert()	PASS
Revert Session to LockingSP; Revert Response - Pass	PASS
Verify RangeStart/Length on Range1/2-zero after LokingSP.Revert()	PASS
RevertSP to LockingSP without KeepGlobalRangeKey; RevertSP Response - Pass Verify RangeStart/Length on Range1/2=zero after RevertSP() without the parameter	PASS PASS
RevertSP to LockingSP with KeepGlobalRangeKey=True; RevertSP Response - Pass	PASS
Neering accounting with the experimental and the experimental accounting to the experimental accounting the experi	PASS
RevertSP to LockingSP with KeepGlobalRangeKey=False; RevertSP Response - Pass	PASS
Verify RangeStart/Length on Range1/2=zero after RevertSP() with KeepGlobalRangeKey=False	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
** Opal V2.0 - Table Contents **	
C1: Level 0 Discovery Contents	PASS
(C1) Display the contents from Level0_Discovery	PASS
(C1) Check TPer Feature	PASS
(C1) Check Locking Feature	PASS
(C1) Check SSC Feature	PASS
	D400
C2: Properties Contents	PASS
(C2) Properties Parameter and Host Properties Parameter	PASS PASS
(C2) Check TPer properties (C2(1)) Check Host properties	PASS
(02/1) Oncor host properties	17100
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 Chockthe restrict for the start of the chiefer	PASS PASS
Check the state of LockingSP	PASS
End Session - Request End Session - Response	PASS
C3: Get() Byte Table: DataStore and MBR Contents check	PASS
(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	PASS
(C3-24) Get data from MBR table	PASS
(C3-24) Data Comparison from MBR table	PASS
C3: Get() Object Table Contents to AdminSP	PASS
(C3-1) Table: Next() method for table (AdminSP)	PASS
(C3-1) Table: Get the entries from table (AdminSP)	PASS
(C3-1) Table: Verify the table contents (AdminSP)	PASS
(C3-2) SPInfo: Get the entries from table (AdminSP)	PASS
(C3-2) SPInfo: Verify the table contents (AdminSP)	PASS
(C3-3) SPTemplates: Next() method for table (AdminSP)	PASS
(C3-3) SPTemplates: Get the entries from table (AdminSP)	
(C3-3) SPTemplates: Verify the table contents (AdminSP)	PASS PASS

(C3-4) MethodID: Next() method for table (AdminSP)	
	PASS
(C3-4) MethodID: Get the entries from table (AdminSP)	PASS
(C3-4) MethodID: Verify the table contents (AdminSP)	PASS
(C3-6) ACE: Next() method for table (AdminSP)	PASS
(C3-6) ACE: Get the entries from table (AdminSP)	PASS
(C3-6) ACE: Verify the table contents (AdminSP)	PASS
(C3-7) Authority: Next() method for table (AdminSP)	PASS
(C3-7) Authority: Get the entries from table (AdminSP)	PASS
(C3-7) Authority: Verify the table contents (AdminSP)	PASS
(C3-8) C_PIN: Next() method for table (AdminSP)	PASS
(C3-8) C_PIN: Get the entries from table (AdminSP)	PASS
(C3-8) C_PIN: Verify the table contents (AdminSP)	PASS
(C3-9) TPerInfo: Get the entries from table (AdminSP)	PASS
(C3-9) TPerInfo: Verify the table contents (AdminSP)	PASS
	PASS
(C3-10) Template: Next() method for table (AdminSP)	
(C3-10) Template: Get the entries from table (AdminSP)	PASS
(C3-10) Template: Verify the table contents (AdminSP)	PASS
(C3-11) SP: Next() method for table (AdminSP)	PASS
(C3-11) SP: Get the entries from table (AdminSP)	PASS
(C3-11) SP: Verify the table contents (AdminSP)	PASS
C2: Cat/) Object Table Contents to Locking SD	PASS
C3: Get() Object Table Contents to LockingSP	
(C3-12) Table: Next() method for table (LockingSP)	PASS
(C3-12) Table: Get the entries from table (LockingSP)	PASS
(C3-12) Table: Verify the table contents (LockingSP)	PASS
(C3-13) SPInfo: Get the entries from table (LockingSP)	PASS
(C3-13) SPInfo: Verify the table contents (LockingSP)	PASS
(C3-14) SPTemplates: Next() method for table (LockingSP)	PASS
(C3-14) SPTemplates: Get the entries from table (LockingSP)	PASS
(C3-14) SPTemplates: Verify the table contents (LockingSP)	PASS
(C3-16) MethodID: Next() method for table (LockingSP)	PASS
(C3-16) MethodID: Get the entries from table (LockingSP)	PASS
(C3-16) MethodID: Verify the table contents (LockingSP)	PASS
(C3-18) ACE: Next() method for table (LockingSP)	PASS
(C3-18) ACE: Get the entries from table (LockingSP)	PASS
(C3-18) ACE: Verify the table contents (LockingSP)	PASS
(C3-19) Authority: Next() method for table (LockingSP)	PASS
(C3-19) Authority: Get the entries from table (LockingSP)	PASS
(C3-19) Authority: Verify the table contents (LockingSP)	PASS
	PASS
(C3-20) C_PIN: Next() method for table (LockingSP)	
(C3-20) C_PIN: Get the entries from table (LockingSP)	PASS
(C3-20) C_PIN: Verify the table contents (LockingSP)	PASS
(C3-21) LockingInfo: Get the entries from table (LockingSP)	PASS
(GG-21) EUCKINGIIIO. GET THE ENTITIES HOTH TABLE (EUCKINGGI)	1 700
(C3-21) LockingInfo: Verify the table contents (LockingSP)	PASS
	PASS PASS
(C3-22) Locking: Next() method for table (LockingSP)	PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP)	PASS PASS
(C3-22) Locking: Next() method for table (LockingSP)	PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP)	PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP)	PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP)	PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP)	PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP)	PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP)	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP)	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP)	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP)	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP)	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Get the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP)	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP)	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Get the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP)	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP)	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Oet the entries from table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP)	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Get the table contents (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Get the entries from table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP)	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Oet the entries from table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP)	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-) SecretProtect: Verify the table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP)	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Get the entries from table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C4-1) Next() - Table Table (C4-1) Verify UIDs for Table Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-) SecretProtect: Verify the table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP)	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Get the entries from table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C4-1) Next() - Table Table (C4-1) Verify UIDs for Table Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-3) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Get the entries from table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C4-1) Next() - Table Table (C4-1) Verify UIDs for Table Table (C4-1) Verify UIDs for Table Table (C4-3) Next() - SPTemplates Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Wext() method for table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) C4: Next() Table Contents (AdminSP) (C4-1) Next() - Table Table (C4-3) Next() - SPTemplates Table (C4-3) Next() - WethodID Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-3) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Get the entries from table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C4-1) Next() - Table Table (C4-1) Verify UIDs for Table Table (C4-1) Verify UIDs for Table Table (C4-3) Next() - SPTemplates Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Wext() method for table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) C4: Next() Table Contents (AdminSP) (C4-1) Next() - Table Table (C4-3) Next() - SPTemplates Table (C4-3) Next() - WethodID Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Set the entries from table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C4-1) Next() - Table Table (C4-1) Next() - Table Table (C4-3) Nerify UIDs for Table Table (C4-4) Next() - MethodID Table (C4-4) Next() - MethodID Table (C4-4) Next() - Authority Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-3) SecretProtect: Next() method for table (LockingSP) (C3-1) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Get the entries from table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C4-1) Next() - Table Table (C4-1) Verify UIDs for Table Table (C4-3) Next() - SPTemplates Table (C4-4) Next() - MethodID Table (C4-4) Next() - MethodID Table (C4-4) Verify UIDs for MethodID Table (C4-6) Verify UIDs for MethodID Table (C4-6) Verify UIDs for Authority Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) C4: Next() Table Contents (AdminSP) (C4-1) Next() - Table Table (C4-1) Next() - SPTemplates Table (C4-3) Next() - SPTemplates Table (C4-4) Next() - MethodID Table (C4-4) Next() - MethodID Table (C4-6) Next() - MethodID Table (C4-6) Next() - Authority Table (C4-6) Next() - Authority Table (C4-7) Next() - ACE Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-3) SecretProtect: Next() method for table (LockingSP) (C3-1) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Get the entries from table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C4-1) Next() - Table Table (C4-1) Verify UIDs for Table Table (C4-3) Next() - SPTemplates Table (C4-4) Next() - MethodID Table (C4-4) Next() - MethodID Table (C4-4) Verify UIDs for MethodID Table (C4-6) Verify UIDs for MethodID Table (C4-6) Verify UIDs for Authority Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) C4: Next() Table Contents (AdminSP) (C4-1) Next() - Table Table (C4-1) Next() - Table Table Table (C4-3) Next() - SPTemplates Table (C4-4) Next() - MethodID Table (C4-4) Next() - MethodID Table (C4-4) Next() - Authority Table (C4-6) Next() - Authority Table (C4-7) Next() - ACE Table (C4-7) Next() - ACE Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-) SecretProtect: Verify the table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C4-1) Next() - Table Table (C4-1) Verify UIDs for Table Table (C4-3) Next() - SPTemplates Table (C4-4) Next() - MethodID Table (C4-4) Next() - MethodID Table (C4-6) Next() - Authority Table (C4-6) Verify UIDs for Acte Table (C4-7) Next() - Authority Table (C4-7) Next() - Authority Table (C4-7) Next() - Acte Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) C4: Next() Table Contents (AdminSP) (C4-1) Next() - Table Table (C4-1) Next() - Table Table Table (C4-3) Next() - SPTemplates Table (C4-4) Next() - MethodID Table (C4-4) Next() - MethodID Table (C4-4) Next() - Authority Table (C4-6) Next() - Authority Table (C4-7) Next() - ACE Table (C4-7) Next() - ACE Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-) SecretProtect: Verify the table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C4-1) Next() - Table Table (C4-1) Verify UIDs for Table Table (C4-3) Next() - SPTemplates Table (C4-4) Next() - MethodID Table (C4-4) Next() - MethodID Table (C4-6) Next() - Authority Table (C4-6) Verify UIDs for Acte Table (C4-7) Next() - Authority Table (C4-7) Next() - Authority Table (C4-7) Next() - Acte Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) C4: Next() Table Contents (AdminSP) (C4-1) Next() - Table Table (C4-1) Next() - Table Table (C4-3) Next() - SPTemplates Table (C4-4) Next() - SPTemplates Table (C4-4) Next() - MethodID Table (C4-4) Next() - MethodID Table (C4-6) Next() - Authority Table (C4-6) Next() - Authority Table (C4-7) Next() - ACE Table (C4-7) Next() - ACE Table (C4-8) Next() - C_PIN Table (C4-8) Next() - C_PIN Table (C4-8) Next() - C_PIN Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-3) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Get the entries from table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) C4: Next() Table Contents (AdminSP) (C4-1) Next() - Table Table (C4-1) Verify UIDs for Table Table (C4-3) Next() - SPTemplates Table (C4-4) Next() - MethodID Table (C4-4) Next() - MethodID Table (C4-4) Next() - Authority Table (C4-6) Next() - Authority Table (C4-7) Next() - ACE Table (C4-7) Next() - CE Inble (C4-8) Next() - CE Inble (C4-9) Next() - CE In Table (C4-10) Next() - Template Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) C4: Next() Table Contents (AdminSP) (C4-1) Next() - Table Table (C4-1) Next() - Table Table (C4-3) Next() - SPTemplates Table (C4-4) Next() - SPTemplates Table (C4-4) Next() - MethodID Table (C4-4) Next() - MethodID Table (C4-6) Next() - Authority Table (C4-6) Next() - Authority Table (C4-7) Next() - ACE Table (C4-7) Next() - ACE Table (C4-8) Next() - C_PIN Table (C4-8) Next() - C_PIN Table (C4-8) Next() - C_PIN Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-3) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Get the entries from table (LockingSP) (C3-27) RestrictedCmds: Get the entries from table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) C4: Next() Table Contents (AdminSP) (C4-1) Next() Table Table (C4-1) Verify UIDs for Table Table (C4-3) Next() - SPTemplates Table (C4-4) Next() - MethodID Table (C4-4) Next() - Authority Table (C4-4) Next() - Authority Table (C4-6) Next() - Authority Table (C4-7) Next() - ACE Table (C4-8) Next() - CEIN Table (C4-9) Next() - CEIN Table (C4-10) Next() - CEIN Table (C4-10) Next() - CEIN Table (C4-10) Next() - Template Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-33) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C4-1) Next() - Table Table (C4-1) Verify UIDs for Table Table (C4-1) Verify UIDs for SPTemplates Table (C4-3) Verify UIDs for SPTemplates Table (C4-4) Verify UIDs for MethodID Table (C4-4) Verify UIDs for Authority Table (C4-6) Verify UIDs for Authority Table (C4-7) Next() - ACE Table (C4-7) Next() - CE Table (C4-7) Next() - CPIN Table (C4-8) Next() - CPIN Table (C4-10) Next() - Template Table (C4-11) Next() - SP Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) C4: Next() Table Contents (AdminSP) (C4-1) Next() - Table Table (C4-1) Next() - Table Table Table (C4-3) Next() - SPTemplates Table (C4-4) Next() - MethodID Table (C4-4) Next() - MethodID Table (C4-4) Next() - Authority Table (C4-6) Next() - Authority Table (C4-7) Next() - ACE Table (C4-7) Next() - ACE Table (C4-8) Next() - C PIN Table (C4-8) Next() - C PIN Table (C4-8) Next() - D PIN Table (C4-9) Next() - Template Table (C4-10) Next() - Template Table (C4-10) Next() - Template Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Set the entries from table (LockingSP) (C3-3) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C4-1) Next() - Table Table (C4-1) Verify UIDs for Table Table (C4-3) Next() - SPTemplates Table (C4-4) Next() - MethodID Table (C4-4) Next() - MethodID Table (C4-4) Next() - Authority Table (C4-6) Next() - Authority Table (C4-7) Next() - ACE Table (C4-7) Next() - ACE Table (C4-7) Next() - MethodID Table (C4-8) Next() - SPTemplate Table (C4-8) Next() - C_PIN Table (C4-10) Next() - Template Table (C4-11) Next() - SP Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Get the entries from table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) C4: Next() Table Contents (AdminSP) (C4-1) Next() - Table Table (C4-1) Next() - Table Table Table (C4-3) Next() - SPTemplates Table (C4-4) Next() - WethodID Table (C4-4) Next() - MethodID Table (C4-4) Next() - Authority Table (C4-6) Next() - Authority Table (C4-7) Next() - ACE Table (C4-8) Next() - CPIN Table (C4-8) Next() - CPIN Table (C4-10) Next() - Template Table (C4-10) Next() - Template Table (C4-11) Next() - SP Table (C4-11) Next() - Table Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C4-1) Next() - Table Table (C4-1) Verify UIDs for Table Table (C4-1) Verify UIDs for Table Table (C4-3) Next() - SPTemplates Table (C4-4) Next() - MethodID Table (C4-4) Next() - MethodID Table (C4-6) Next() - Authority Table (C4-6) Next() - Authority Table (C4-7) Next() - ACE Table (C4-7) Next() - Table Table (C4-8) Next() - C_PIN Table (C4-8) Next() - C_PIN Table (C4-10) Next() - Template Table (C4-10) Next() - Template Table (C4-11) Next() - SP Table (C4-12) Next() - Table Table (C4-12) Next() - Table Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-3) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-) SecretProtect: Verify the table contents (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) C4: Next() Table Contents (AdminSP) (C4-1) Next() - Table Table (C4-1) Next() - SPTemplates Table (C4-3) Verify UIDs for Table Table (C4-4) Next() - MethodID Table (C4-4) Next() - MethodID Table (C4-4) Next() - MethodID Table (C4-6) Verify UIDs for MethodID Table (C4-7) Verify UIDs for Act Table (C4-8) Next() - Authority Table (C4-9) Next() - Act Table (C4-10) Next() - Template Table (C4-10) Next() - Template Table (C4-10) Next() - Template Table (C4-11) Next() - SP Table (C4-11) Next() - SP Table (C4-11) Next() - Table Table (C4-12) Next() - Table Table (C4-12) Next() - Table Table (C4-12) Next() - Table Table	PASS PASS PASS PASS PASS PASS PASS PASS
(C3-22) Locking: Next() method for table (LockingSP) (C3-22) Locking: Get the entries from table (LockingSP) (C3-22) Locking: Verify the table contents (LockingSP) (C3-23) MBRControl: Get the entries from table (LockingSP) (C3-23) MBRControl: Verify the table contents (LockingSP) (C3-) SecretProtect: Next() method for table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-) SecretProtect: Get the entries from table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Next() method for table (LockingSP) (C3-25) K_AES_256: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Next() method for table (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C3-27) RestrictedCmds: Verify the table contents (LockingSP) (C4-1) Next() - Table Table (C4-1) Verify UIDs for Table Table (C4-1) Verify UIDs for Table Table (C4-3) Next() - SPTemplates Table (C4-4) Next() - MethodID Table (C4-4) Next() - MethodID Table (C4-6) Next() - Authority Table (C4-6) Next() - Authority Table (C4-7) Next() - ACE Table (C4-7) Next() - Table Table (C4-8) Next() - C_PIN Table (C4-8) Next() - C_PIN Table (C4-10) Next() - Template Table (C4-10) Next() - Template Table (C4-11) Next() - SP Table (C4-12) Next() - Table Table (C4-12) Next() - Table Table	PASS PASS PASS PASS PASS PASS PASS 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
	PASS
(C4-19) Next() - Authority Table	
(C4-19) Verify UIDs for Authority Table	PASS
(C4-20) Next() - C_PIN Table	PASS
(C4-20) Verify UIDs for C_PIN Table	PASS
(C4-22) Next() - Locking Table	PASS
(C4-22) Verify UIDs for Locking Table	PASS
(C4-23) Next() - RestrictedCmds Table	N/A
C5: GetACL() Table Contents (AdminSP)	PASS
(C5-1) Next() - Table Table	PASS
(C5-1) GetACL() - Table Table	PASS
(C5-1) Verify ACL values for Table Table	PASS
(C5-2) GetACL() - SPInfo Table	PASS
(C5-2) Verify ACL values for SPInfo Table	PASS
(C5-3) Next() - SPTemplates Table	PASS
(C5-3) GetACL() - SPTemplates Table	PASS
(C5-3) Verify ACL values for SPTemplates Table	PASS
(C5-4) Next() - MethodID Table	PASS
(C5-4) GetACL() - MethodID Table	PASS
(C5-4) Verify ACL values for MethodID Table	PASS
(C5-5) Next() - ACE Table	PASS
(C5-5) GetACL() - ACE Table	PASS
(C5-5) Verify ACL values for ACE Table	PASS
• • •	
(C5-6) Next() - Authority Table	PASS
(C5-6) GetACL() - Authority Table	PASS
(C5-6) Verify ACL values for Authority Table	PASS
(C5-7) Next() - C_PIN Table	PASS
(C5-7) GetACL() - C_PIN Table	PASS
(C5-7) Verify ACL values for C_PIN Table	PASS
(C5-8) GetACL() - TPerInfo Table	PASS
(C5-8) Verify ACL values for TPerInfo Table	PASS
(C5-9) Next() - Template Table	PASS
	PASS
(C5-9) GetACL() - Template Table	
(C5-9) Verify ACL values for Template Table	PASS
(C5-10) Next() - SP Table	PASS
(C5-10) GetACL() - SP Table	PASS
(C5-10) Verify ACL values for SP Table	PASS
(C5-10) Verify ACL values for SP Table	PASS
(C5-10) Verify ACL values for SP Table C5: GetACL() Table Contents (LockingSP)	PASS PASS
C5: GetACL() Table Contents (LockingSP)	PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table	PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table	PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) Verify ACL values for Table Table	PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) Verify ACL values for Table Table (C5-12) GetACL() - SPInfo Table	PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) Verify ACL values for Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPinfo Table	PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) Verify ACL values for Table Table (C5-12) GetACL() - SPInfo Table	PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) Verify ACL values for Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPinfo Table	PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) Verify ACL values for Table Table (C5-12) GetACL() - SPInto Table (C5-12) Verify ACL values for SPinto Table (C5-13) Next() - SPTemplates Table	PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) Verify ACL values for Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPinfo Table (C5-13) Next() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) Verify ACL values for Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPinfo Table (C5-13) Next() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) Yerify ACL values for Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPinfo Table (C5-13) Next() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) Yerify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) GetACL() - MethodID Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) Verify ACL values for Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPinfo Table (C5-13) Next() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) GetACL() - MethodID Table (C5-15) Verify ACL values for MethodID Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) Verify ACL values for Table Table (C5-12) Verify ACL values for SPinfo Table (C5-12) Verify ACL values for SPinfo Table (C5-13) Next() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) GetACL() - MethodID Table (C5-16) Next() - ACE Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) Verify ACL values for Table Table (C5-12) Verify ACL values for Spinfo Table (C5-12) Verify ACL values for Spinfo Table (C5-13) Next() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) Next() - MethodID Table (C5-15) GetACL() - MethodID Table (C5-16) Next() - ACE Table (C5-16) GetACL() - ACE Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) GetACL() - SPInfo Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPInfo Table (C5-13) Next() - SPTemplates Table (C5-13) Next() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) GetACL() - MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-16) Next() - ACE Table (C5-16) Next() - ACE Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) Verify ACL values for Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPinfo Table (C5-13) Next() - SPTemplates Table (C5-13) Next() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) Next() - MethodID Table (C5-15) GetACL() - MethodID Table (C5-16) Next() - ACE Table (C5-16) GetACL() - ACE Table (C5-16) Verify ACL values for ACE Table (C5-16) Verify ACL values for ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) GetACL() - SPInto Table Table (C5-12) GetACL() - SPInto Table (C5-12) Verify ACL values for SPinto Table (C5-13) Next() - SPTemplates Table (C5-13) Next() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) GetACL() - MethodID Table (C5-16) Next() - ACE Table (C5-16) GetACL() - ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) GetACL() - Authority Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) Verify ACL values for Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPinfo Table (C5-13) Next() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) GetACL() - MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-16) Next() - ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) GetACL() - Authority Table (C5-17) GetACL() - Authority Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) GetACL() - SPInto Table Table (C5-12) GetACL() - SPInto Table (C5-12) Verify ACL values for SPinto Table (C5-13) Next() - SPTemplates Table (C5-13) Next() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) GetACL() - MethodID Table (C5-16) Next() - ACE Table (C5-16) GetACL() - ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) GetACL() - Authority Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) Verify ACL values for Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPinfo Table (C5-13) Next() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) GetACL() - MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-16) Next() - ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) GetACL() - Authority Table (C5-17) GetACL() - Authority Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) Verify ACL values for Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPInfo Table (C5-13) Next() - SPTemplates Table (C5-13) Next() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-16) Next() - ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) GetACL() - Authority Table (C5-17) GetACL() - Authority Table (C5-17) Verify ACL values for Authority Table (C5-17) Verify ACL values for Authority Table (C5-17) Next() - C_PIN Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) GetACL() - SPInfo Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPInfo Table (C5-13) Next() - SPTemplates Table (C5-13) Mext() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) Next() - MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-16) Next() - ACE Table (C5-16) Next() - ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) Next() - Authority Table (C5-17) Next() - Authority Table (C5-17) Next() - CPIN Table (C5-18) Next() - CPIN Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) Mext() - Table Table (C5-11) GetACL() - Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPinfo Table (C5-13) Next() - SPTemplates Table (C5-13) Next() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) GetACL() - MethodID Table (C5-16) Next() - ACE Table (C5-16) Next() - ACE Table (C5-16) GetACL() - ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) Werify ACL values for Authority Table (C5-17) Verify ACL values for Authority Table (C5-18) Next() - C_PIN Table (C5-18) Next() - C_PIN Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) Verify ACL values for Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPinfo Table (C5-13) Next() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) GetACL() - MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-16) Next() - ACE Table (C5-16) GetACL() - ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) Verify ACL values for Authority Table (C5-18) Next() - Authority Table (C5-18) Next() - C_PIN Table (C5-18) GetACL() - C_PIN Table (C5-18) Verify ACL values for C_PIN Table (C5-19) Verify ACL values for C_PIN Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) Mext() - Table Table (C5-11) GetACL() - Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPInfo Table (C5-12) Verify ACL values for SPInfo Table (C5-13) Next() - SPTemplates Table (C5-13) Next() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) SetACL() - MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-16) Next() - ACE Table (C5-16) Next() - ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) GetACL() - Authority Table (C5-17) GetACL() - Authority Table (C5-18) Next() - C_PIN Table (C5-18) Next() - C_PIN Table (C5-18) Verify ACL values for C_PIN Table (C5-18) Verify ACL values for C_PIN Table (C5-19) GetACL() - LockingInfo Table (C5-19) Verify ACL values for LockingInfo Table (C5-19) Verify ACL values for LockingInfo Table (C5-19) Verify ACL values for LockingInfo Table (C5-20) Next() - Locking Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) Mext() - Table Table (C5-11) GetACL() - Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPInfo Table (C5-12) Verify ACL values for SPInfo Table (C5-13) Next() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) Next() - MethodID Table (C5-15) SetACL() - MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-16) Next() - ACE Table (C5-16) Next() - ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) Next() - Authority Table (C5-17) GetACL() - Authority Table (C5-18) Next() - C_PIN Table (C5-18) Next() - C_PIN Table (C5-18) Verify ACL values for C_PIN Table (C5-18) Verify ACL values for C_PIN Table (C5-19) GetACL() - LockingInfo Table (C5-19) Verify ACL Values for LockingInfo Table (C5-20) Next() - Locking Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) GetACL() - SPInfo Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPInfo Table (C5-13) Next() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) Mext() - MethodID Table (C5-16) Verify ACL values for MethodID Table (C5-16) Next() - ACE Table (C5-16) Next() - ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) Next() - Authority Table (C5-17) Verify ACL values for Authority Table (C5-18) Next() - C_PIN Table (C5-18) Next() - C_PIN Table (C5-18) Verify ACL values for C_PIN Table (C5-19) Verify ACL values for C_PIN Table (C5-19) Verify ACL values for LockingInfo Table (C5-20) Next() - Locking Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPinfo Table (C5-13) Next() - SPTemplates Table (C5-13) Mext() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) Next() - MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-16) Next() - ACE Table (C5-16) Next() - ACE Table (C5-16) GetACL() - ACE Table (C5-17) Next() - Authority Table (C5-17) Next() - Authority Table (C5-17) Verify ACL values for Authority Table (C5-17) Verify ACL values for Authority Table (C5-18) Next() - C_PIN Table (C5-18) Next() - C_PIN Table (C5-18) Verify ACL values for C_PIN Table (C5-19) GetACL() - LockingInfo Table (C5-19) Verify ACL values for Table (C5-19) Verify ACL values for LockingInfo Table (C5-20) Next() - Locking Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) GetACL() - SPInto Table (C5-12) GetACL() - SPInto Table (C5-12) Verify ACL values for SPinto Table (C5-13) Next() - SPTemplates Table (C5-13) Next() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) GetACL() - MethodID Table (C5-16) Next() - ACE Table (C5-16) Next() - ACE Table (C5-16) GetACL() - ACE Table (C5-16) GetACL() - ACE Table (C5-17) Next() - Authority Table (C5-17) Next() - Authority Table (C5-17) Verify ACL values for Authority Table (C5-18) Next() - C_PIN Table (C5-18) GetACL() - C_PIN Table (C5-18) GetACL() - C_PIN Table (C5-19) GetACL() - LockingInfo Table (C5-19) Verify ACL values for LockingInfo Table (C5-20) Next() - Locking Table (C5-20) Next() - Locking Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPinfo Table (C5-13) Next() - SPTemplates Table (C5-13) Mext() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) Next() - MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-16) Next() - ACE Table (C5-16) Next() - ACE Table (C5-16) GetACL() - ACE Table (C5-17) Next() - Authority Table (C5-17) Next() - Authority Table (C5-17) Verify ACL values for Authority Table (C5-17) Verify ACL values for Authority Table (C5-18) Next() - C_PIN Table (C5-18) Next() - C_PIN Table (C5-18) Verify ACL values for C_PIN Table (C5-19) GetACL() - LockingInfo Table (C5-19) Verify ACL values for Table (C5-19) Verify ACL values for LockingInfo Table (C5-20) Next() - Locking Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) GetACL() - SPInto Table (C5-12) GetACL() - SPInto Table (C5-12) Verify ACL values for SPinto Table (C5-13) Next() - SPTemplates Table (C5-13) Next() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) GetACL() - MethodID Table (C5-16) Next() - ACE Table (C5-16) Next() - ACE Table (C5-16) GetACL() - ACE Table (C5-16) GetACL() - ACE Table (C5-17) Next() - Authority Table (C5-17) Next() - Authority Table (C5-17) Verify ACL values for Authority Table (C5-18) Next() - C_PIN Table (C5-18) GetACL() - C_PIN Table (C5-18) GetACL() - C_PIN Table (C5-19) GetACL() - LockingInfo Table (C5-19) Verify ACL values for LockingInfo Table (C5-20) Next() - Locking Table (C5-20) Next() - Locking Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) GetACL() - SPInfo Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPInfo Table (C5-13) Next() - SPTemplates Table (C5-13) Next() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Mext() - MethodID Table (C5-15) SetACL() - MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-16) Next() - ACE Table (C5-16) Next() - ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) Next() - Authority Table (C5-17) GetACL() - Authority Table (C5-18) Next() - C_PIN Table (C5-18) Next() - C_PIN Table (C5-18) SetACL() - C_PIN Table (C5-18) Verify ACL values for C_PIN Table (C5-19) GetACL() - LockingInfo Table (C5-19) GetACL() - LockingInfo Table (C5-20) Next() - Locking Table (C5-20) Verify ACL values for LockingInfo Table (C5-21) GetACL() - Locking Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) GetACL() - SPInfo Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPInfo Table (C5-13) Next() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) Mext() - MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-16) Next() - ACE Table (C5-16) Next() - ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) Next() - Authority Table (C5-17) Verify ACL values for Authority Table (C5-18) Next() - C_PIN Table (C5-18) Next() - C_PIN Table (C5-18) Verify ACL values for C_PIN Table (C5-19) Verify ACL values for LockingInfo Table (C5-20) Next() - Locking Table (C5-20) Next() - Locking Table (C5-21) Verify ACL values for Locking Table (C5-22) Verify ACL values for MBRControl Table (C5-22) Verify ACL values for MBRControl Table (C5-22) Verify ACL values for MBR Table	PASS PASS PASS PASS PASS PASS PASS PASS
CS: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPinfo Table (C5-13) Next() - SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) Next() - MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-16) Next() - ACE Table (C5-16) Next() - ACE Table (C5-16) Next() - ACE Table (C5-16) GetACL() - ACE Table (C5-17) Next() - Authority Table (C5-17) Next() - Authority Table (C5-17) Verify ACL values for Authority Table (C5-18) Next() - C_PIN Table (C5-18) Next() - C_PIN Table (C5-18) Verify ACL values for C_PIN Table (C5-18) GetACL() - Locking Info Table (C5-19) GetACL() - Locking Info Table (C5-19) GetACL() - Locking Table (C5-19) GetACL() - Locking Table (C5-20) Next() - Locking Table (C5-21) GetACL() - MBRControl Table (C5-22) Verify ACL values for MBRControl Table (C5-22) Verify ACL values for MBRControl Table (C5-22) Verify ACL values for MBRControl Table (C5-23) Verify ACL values for MBRControl Table (C5-23) Verify ACL values for MBRControl Table	PASS PASS PASS PASS PASS PASS PASS PASS
CS: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) GetACL() - SPInto Table (C5-12) GetACL() - SPInto Table (C5-12) Verify ACL values for SPinto Table (C5-13) Next() - SPTemplates Table (C5-13) Next() - SPTemplates Table (C5-13) Verify ACL values for SPEmplates Table (C5-13) Verify ACL values for SPEmplates Table (C5-13) Verify ACL values for SPEmplates Table (C5-15) Next() - MethodID Table (C5-15) Next() - MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-16) Next() - ACE Table (C5-16) Next() - ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) Verify ACL values for Authority Table (C5-17) Verify ACL values for Authority Table (C5-18) Next() - C_PIN Table (C5-18) Next() - C_PIN Table (C5-18) Verify ACL values for C_PIN Table (C5-19) GetACL() - LockingInfo Table (C5-19) Verify ACL values for LockingInfo Table (C5-20) Next() - Locking Table (C5-20) Next() - Locking Table (C5-21) Verify ACL values for Locking Table (C5-22) GetACL() - MBRControl Table (C5-23) Verify ACL values for MBRControl Table (C5-23) GetACL() - MBR Table	PASS PASS PASS PASS PASS PASS PASS PASS
CS: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) GetACL() - SPInfo Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPInfo Table (C5-13) Next() - SPTemplates Table (C5-13) Next() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) Next() - MethodID Table (C5-15) SetACL() - MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-16) Next() - ACE Table (C5-16) Next() - ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) Next() - Authority Table (C5-17) GetACL() - Authority Table (C5-17) GetACL() - Authority Table (C5-18) Next() - C_PIN Table (C5-18) Mext() - C_PIN Table (C5-18) Verify ACL values for C_PIN Table (C5-19) GetACL() - LockingInfo Table (C5-19) GetACL() - LockingInfo Table (C5-19) GetACL() - Locking Table (C5-20) Next() - Locking Table (C5-20) Verify ACL values for LockingInfo Table (C5-21) GetACL() - MBRControl Table (C5-22) Verify ACL values for MBR Table (C5-22) Verify ACL values for MBR Table (C5-23) Verify ACL values for MBR Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) GetACL() - SPInfo Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPInfo Table (C5-13) Next() - SPTemplates Table (C5-13) Next() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) Next() - MethodID Table (C5-15) Next() - MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-16) Next() - ACE Table (C5-16) Next() - ACE Table (C5-16) Next() - ACE Table (C5-16) Verify ACL values for ACETable (C5-17) Next() - Authority Table (C5-17) Next() - Authority Table (C5-17) Verify ACL values for Authority Table (C5-18) Next() - C_PIN Table (C5-18) Next() - C_PIN Table (C5-18) Verify ACL values for C_PIN Table (C5-19) Yerify ACL values for C_PIN Table (C5-19) Yerify ACL values for LockingInfo Table (C5-19) Verify ACL values for LockingInfo Table (C5-20) Next() - Locking Table (C5-20) Next() - Locking Table (C5-21) Verify ACL values for MBRControl Table (C5-22) Verify ACL values for MBRCOntrol Table (C5-22) Verify ACL values for MBRCOntrol Table (C5-23) Verify ACL values for MBR Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) Next() - Table Table (C5-11) Verify ACL values for Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPInfo Table (C5-13) Next() - SPTemplates Table (C5-13) Next() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) Next() - MethodID Table (C5-15) Next() - MethodID Table (C5-15) Next() - ACE Table (C5-16) Verify ACL values for MethodID Table (C5-16) Verify ACL values for ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) Next() - Authority Table (C5-17) Verify ACL values for Authority Table (C5-18) Next() - C_PIN Table (C5-18) Next() - C_PIN Table (C5-18) GetACL() - C_PIN Table (C5-19) GetACL() - LockingInfo Table (C5-19) Verify ACL values for LockingInfo Table (C5-19) Verify ACL values for LockingInfo Table (C5-20) Next() - Locking Table (C5-20) Next() - Locking Table (C5-20) GetACL() - MBR Table (C5-21) Verify ACL values for MBRControl Table (C5-22) Verify ACL values for MBRControl Table (C5-23) GetACL() - MBR Table (C5-24) Verify ACL values for MBRControl Table (C5-24) Verify ACL values for MBRControl Table (C5-23) GetACL() - Locking Table (C5-24) Verify ACL values for MBRControl Table (C5-23) GetACL() - Locking Table	PASS PASS PASS PASS PASS PASS PASS PASS
CS: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) GetACL() - Table Table (C5-12) GetACL() - SPInfo Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPinfo Table (C5-13) Next() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Mext() - MethodID Table (C5-15) Next() - MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-16) Verify ACL values for MethodID Table (C5-16) Verify ACL values for MethodID Table (C5-16) GetACL() - ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) Next() - Authority Table (C5-17) Verify ACL values for Authority Table (C5-18) Mext() - C_PIN Table (C5-18) SetACL() - LockingInfo Table (C5-18) Verify ACL values for C_PIN Table (C5-19) Verify ACL values for LockingInfo Table (C5-19) Verify ACL values for LockingInfo Table (C5-20) Next() - Locking Table (C5-20) Next() - Locking Table (C5-21) GetACL() - MBR Table (C5-22) GetACL() - MBR Table (C5-22) Verify ACL values for MBRControl Table (C5-23) Verify ACL values for MBR Table (C5-24) Verify ACL values for MBR Table (C5-25) OetACL() - MBR Table	PASS PASS PASS PASS PASS PASS PASS PASS
C5: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) Next() - Table Table (C5-11) Verify ACL values for Table Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPInfo Table (C5-13) Next() - SPTemplates Table (C5-13) Next() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Next() - MethodID Table (C5-15) Next() - MethodID Table (C5-15) Next() - MethodID Table (C5-15) Next() - ACE Table (C5-16) Verify ACL values for MethodID Table (C5-16) Verify ACL values for ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) Next() - Authority Table (C5-17) Verify ACL values for Authority Table (C5-18) Next() - C_PIN Table (C5-18) Next() - C_PIN Table (C5-18) GetACL() - C_PIN Table (C5-19) GetACL() - LockingInfo Table (C5-19) Verify ACL values for LockingInfo Table (C5-19) Verify ACL values for LockingInfo Table (C5-20) Next() - Locking Table (C5-20) Next() - Locking Table (C5-20) GetACL() - MBR Table (C5-21) Verify ACL values for MBRControl Table (C5-22) Verify ACL values for MBRControl Table (C5-23) GetACL() - MBR Table (C5-24) Verify ACL values for MBRControl Table (C5-24) Verify ACL values for MBRControl Table (C5-23) GetACL() - Locking Table (C5-24) Verify ACL values for MBRControl Table (C5-23) GetACL() - Locking Table	PASS PASS PASS PASS PASS PASS PASS PASS
CS: GetACL() Table Contents (LockingSP) (C5-11) Next() - Table Table (C5-11) GetACL() - Table Table (C5-11) GetACL() - Table Table (C5-12) GetACL() - SPInfo Table (C5-12) GetACL() - SPInfo Table (C5-12) Verify ACL values for SPinfo Table (C5-13) Next() - SPTemplates Table (C5-13) GetACL() - SPTemplates Table (C5-13) Verify ACL values for SPTemplates Table (C5-15) Mext() - MethodID Table (C5-15) Next() - MethodID Table (C5-15) Verify ACL values for MethodID Table (C5-16) Verify ACL values for MethodID Table (C5-16) Verify ACL values for MethodID Table (C5-16) GetACL() - ACE Table (C5-16) Verify ACL values for ACE Table (C5-17) Next() - Authority Table (C5-17) Next() - Authority Table (C5-17) Verify ACL values for Authority Table (C5-18) Mext() - C_PIN Table (C5-18) SetACL() - LockingInfo Table (C5-18) Verify ACL values for C_PIN Table (C5-19) Verify ACL values for LockingInfo Table (C5-19) Verify ACL values for LockingInfo Table (C5-20) Next() - Locking Table (C5-20) Next() - Locking Table (C5-21) GetACL() - MBR Table (C5-22) GetACL() - MBR Table (C5-22) Verify ACL values for MBRControl Table (C5-23) Verify ACL values for MBR Table (C5-24) Verify ACL values for MBR Table (C5-25) OetACL() - MBR Table	PASS PASS PASS PASS PASS PASS PASS PASS
CS: GetACL() Table Table (CS-11) Next() - Table Table (CS-11) GetACL() - Table Table (CS-11) GetACL() - SPInfo Table (CS-12) GetACL() - SPInfo Table (CS-12) Verify ACL values for Spinfo Table (CS-13) Next() - SPTemplates Table (CS-13) Next() - SPTemplates Table (CS-13) Next() - SPTemplates Table (CS-13) Verify ACL values for SPTemplates Table (CS-15) Next() - MethodID Table (CS-15) Next() - MethodID Table (CS-15) Next() - MethodID Table (CS-15) Next() - ACE Table (CS-16) GetACL() - ACE Table (CS-16) Verify ACL values for ACE Table (CS-16) Verify ACL values for ACE Table (CS-17) Next() - Authority Table (CS-17) Verify ACL values for Authority Table (CS-18) Next() - C_PIN Table (CS-18) Next() - C_PIN Table (CS-18) Verify ACL values for C_PIN Table (CS-19) GetACL() - LockingInfo Table (CS-19) Verify ACL values for LockingInfo Table (CS-19) Verify ACL values for LockingInfo Table (CS-20) Verify ACL values for Locking Table (CS-20) Verify ACL values for Locking Table (CS-20) Verify ACL values for MBRControl Table (CS-21) GetACL() - Locking Table (CS-22) Verify ACL values for MBRControl Table (CS-23) GetACL() - K_AES_128 Table (CS-23) GetACL() - K_AES_128 Table (CS-23) GetACL() - K_AES_128 Table (CS-23) Verify ACL values for K_AES_128 Table (CS-24) GetACL() - DataStore Table (CS-25) Verify ACL values for DataStore Table	PASS PASS PASS PASS PASS PASS PASS PASS

(C5-) Verify ACL values for SecretProtect Table	PASS
(C5-26) Next() - RestrictedCmds Table	N/A
(C5-26) GetACL() - RestrictedCmds Table	N/A
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
Activating the Locking SD	PASS
Activating the Locking SP	
Start Session with HostChallenge - AdminSP	PASS
Sync Session - AdminSP	PASS
Activate_LockingSP	PASS
Activate_LockingSP - Response	PASS
Get - LifeCycle(Locking SP) - Request	PASS
Get - LifeCycle(Locking SP) - Response	PASS
Check the state of LockingSP	PASS
	PASS
End Session - Request	
End Session - Response	PASS
** Opal V2.0 - Feature Set **	
Opal SSC Feature Set: PSID	PASS
Check the support of PSID Authority	PASS
Verify the contents of C_PIN_PSID in C_PIN table	PASS
Verify the contents of ACE_C_PIN_Get_PSID_NoPIN in ACE table	PASS
Verify the contents of ACE SP PSID in ACE table	PASS
•	
Verify ACE_C_PIN_Get_PSID_NoPIN in AccessControl Table	PASS
Start a session to AdminSP as PSID with PSID's PIN	PASS
Revert AdminSP - Pass	PASS
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	PASS
Check the state of LockingSP	PASS
Check the state of LockingSP End Session - Request	PASS PASS
End Session - Request	PASS
End Session - Request End Session - Response	PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables	PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery	PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass	PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery	PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table	PASS PASS PASS PASS PASS PASS PASS PAS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery	PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table	PASS PASS PASS PASS PASS PASS PASS PAS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass	PASS PASS PASS PASS PASS PASS PASS PAS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space)	PASS PASS PASS PASS PASS PASS PASS PAS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with non-align DataStore; Response - StatusCode=0Ch(Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS PAS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with non-align DataStore; Response - StatusCode=0Ch(Invalid_Param) Activate() method without dataStoreList; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with non-align DataStore; Response - StatusCode=0Ch(Invalid_Param) Activate() method without dataStoreList; Response - Pass Activate():The size of dataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (> maxDSSize); Response - Pass Activate() method with non-align DataStore; Response - StatusCode=09h(Insufficient_Space) Activate() method with non-align DataStore; Response - StatusCode=0Ch(Invalid_Param) Activate() method without dataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with all DataStore table; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with non-align DataStore; Response - StatusCode=0Ch(Invalid_Param) Activate() method without dataStoreList; Response - Pass Activate():The size of dataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (> maxDSSize); Response - Pass Activate() method with non-align DataStore; Response - StatusCode=09h(Insufficient_Space) Activate() method with non-align DataStore; Response - StatusCode=0Ch(Invalid_Param) Activate() method without dataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with all DataStore table; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with non-align DataStore; Response - Pass Activate() method without dataStore list qual to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with all DataStore table; Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate(); method with non-align DataStore; Response - Pass Activate(); The size of dataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - StatusCode=09h(Insufficient_Space)	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore is exponse - StatusCode=00h(Invalid_Param) Activate() method with out dataStoreList; Response - Pass Activate():The size of dataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - StatusCode=09h(Invalid_Param) ReActivate() method with DataStore size (<= maxDSSize); Response - StatusCode=09h(Invalid_Param) ReActivate() method with non-align DataStore; Response - Pass Reactivate() method with non-align DataStore size (>= maxDSSize); Response - StatusCode=09h(Invalid_Param) ReActivate() method with out dataStoreList; Response - Pass	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with non-align DataStore; Response - Pass Activate() method without dataStore lable; Response - Pass Activate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (<= maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with non-align DataStore; Response - StatusCode=00Ch(Invalid_Param)	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with non-align DataStore; Response - Pass Activate() method with adtaStore size (> maxDssize); Response - Pass Activate() method with all DataStore table; Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (<= maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (> maxDsSize); Response - StatusCode=00ch(Invalid_Param) Reactivate() method with on-align DataStore; Response - StatusCode=00ch(Invalid_Param) Reactivate() method with on-align DataStore; Response - StatusCode=00ch(Invalid_Param) Reactivate() method without dataStoreList; Response - Pass Reactivate() method without dataStoreList; Response - StatusCode=00ch(Invalid_Param) Reactivate() method without dataStoreList; Response - StatusCode=00ch(Invalid_Param) Reactivate() method without dataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with DataStore in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with non-align DataStore; Response - Pass Activate() method with dataStore size (spanse - Pass Activate():The size of dataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=00h(Insufficient_Space) Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=00h(Insufficient_Space) Reactivate() method with DataStore size (> maxDsSize); Response - StatusCode=00h(Insufficient_Space) Reactivate() method with DataStore size (> maxDsSize); Response - StatusCode=00h(Insufficient_Space) Reactivate() method with DataStore size (> maxDsSize); Response - StatusCode=00h(Insufficient_Space) Reactivate() method with DataStore size (> maxDsSize); Response - StatusCode=00h(Insufficient_Space) Reactivate() method with DataStore size (> maxDsSize); Response - StatusCode=00h(Insufficient_Space) Reactivate() method with DataStore size (> maxDsSize); Response - StatusCode=00h(Insufficient_Space) Reactivate() method with DataStore size (> maxDsSi	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with non-align DataStore; Response - Pass Activate() method without dataStoreList; Response - Pass Activate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (>= maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with non-align DataStore; Response - StatusCode=0Ch(Invalid_Param) ReActivate() method without dataStoreList; Response - Pass Reactivate() method without dataStoreList; Response - Pass ReActivate() method without dataStoreList; Response - Pass ReActivate() The size of dataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Opal SSC Feature Set: Single User Mode Check the feature support of Single User Mode	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (< maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with out dataStoreList; Response - Pass Activate():The size of dataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with DataStore size (< maxDSSize); Response - Pass Reactivate() method with DataStore size (< maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (> maxDSSize); Response - Pass Reactivate() method with out-align DataStore; Response - Pass Reactivate() method with out-align DataStore; Response - Pass Reactivate() method with out-align DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Opal SSC Feature Set: Single User Mode Check the feature support of Single User Mode Check the support of ReActivate and Erase methods in the MethodID table	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (> maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with on-align DataStore; Response - Pass Activate() method without dataStore list; Response - Pass Activate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with DataStore table; Response - Pass Reactivate() method with DataStore size (< maxDSSize); Response - Pass Reactivate() method with DataStore size (< maxDSSize); Response - Pass Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=0Ch(Invalid_Param) ReActivate() method with on-align DataStore; Response - Pass ReActivate() method with ond dataStore size (> maxDSSize); Response - Pass ReActivate() method with ond dataStore size (> maxDSSize); Response - StatusCode=0Ch(Invalid_Param) ReActivate() method with ond AdataStore size (> maxDSSize); Response - StatusCode=0Ch(Invalid_Param) ReActivate() method with ond AdataStore size (> maxDSSize); Response - StatusCode=0Ch(Invalid_Param) ReActivate() method with ond AdataStore size (> maxDSSize); Response - StatusCode=0Ch(Invalid_Param	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (< maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with out dataStoreList; Response - Pass Activate():The size of dataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with DataStore size (< maxDSSize); Response - Pass Reactivate() method with DataStore size (< maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (> maxDSSize); Response - Pass Reactivate() method with out-align DataStore; Response - Pass Reactivate() method with out-align DataStore; Response - Pass Reactivate() method with out-align DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Opal SSC Feature Set: Single User Mode Check the feature support of Single User Mode Check the support of ReActivate and Erase methods in the MethodID table	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (> maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with on-align DataStore; Response - Pass Activate() method without dataStore list; Response - Pass Activate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with DataStore table; Response - Pass Reactivate() method with DataStore size (< maxDSSize); Response - Pass Reactivate() method with DataStore size (< maxDSSize); Response - Pass Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=0Ch(Invalid_Param) ReActivate() method with on-align DataStore; Response - Pass ReActivate() method with ond dataStore size (> maxDSSize); Response - Pass ReActivate() method with ond dataStore size (> maxDSSize); Response - StatusCode=0Ch(Invalid_Param) ReActivate() method with ond AdataStore size (> maxDSSize); Response - StatusCode=0Ch(Invalid_Param) ReActivate() method with ond AdataStore size (> maxDSSize); Response - StatusCode=0Ch(Invalid_Param) ReActivate() method with ond AdataStore size (> maxDSSize); Response - StatusCode=0Ch(Invalid_Param	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with AcE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (> maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with non-align DataStore; Response - StatusCode=0Ch(Invalid_Param) Activate() method with out dataStoreList; Response - Pass Activate() method with DataStore size (> maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - Pass Reactivate() method with DataStore size (> maxDSSize); Response - Pass Reactivate() method with DataStore size (> maxDSSize); Response - Pass Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (> maxDSSize); Response - Pass Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=00h(Invalid_Param) ReActivate() method with out dataStoreList; Response - Pass ReActivate() of dataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Opal SSC Feature Set: Single User Mode Check the support of ReActivate and Erase methods in the MethodID table Get the valu	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (>= maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with out adaStore! sepanse - Pass Activate() method with out adaStore! sequal to the "Maximum total size of DataStore' from Level0_Discovery ReActivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (>= maxDSSize); Response - Pass Reactivate() method with DataStore size (>= maxDSSize); Response - Pass Reactivate() method with out adaStore; Response - StatusCode=0Ch(Invalid_Param) ReActivate() method with out dataStore; Response - Pass Reactivate() method with out dataStore; Response - Pass ReActivate() method with out dataStore; Response - Pass ReActivate() method with out dataStore is equal to the "Maximum total size of DataStore' from Level0_Discovery Opal SSC Feature Set: Single User Mode Check the support of Single User Mode Check the support of ReActivate and Erase methods in the MethodID table Get the values of 'SingleUserModeRanges' and 'RangeStartLengthPolicy' from the LockingInfo table Activate() method with LockingObject not included in Locking Template; Response - StatusCode=OCh(Invalid_Param) Activate() method	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (<= maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with non-align DataStore; Response - StatusCode=00h(Invalid_Param) Activate() method with out dataStore lable; Response - Pass Activate() The size of dataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - StatusCode=00h(Invalid_Param) ReActivate() method with DataStore; Response - StatusCode=00h(Invalid_Param) ReActivate() method with out dataStoreList; Response - Pass ReActivate() method with out dataStoreList; Response - Pass ReActivate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Opal SSC Feature Set: Single User Mode from Level0_Discovery Check the feature support of Single User Mode from Level0_Discovery Check the support of ReActivate and Erase methods in the MethodID table Get the values of 'SingleUserModeRanges' and 'RangeStartLengthPolicy' from the LockingInfo table Activate() method with LockingObject not included in Locking Template; Response - StatusCode=	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with AEI in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (>= maxDSSize); Response - Pass Activate() method with DataStore size (>= maxDSsize); Response - StatusCode=09h(Insufficient_Space) Activate() method with on-align DataStore; Response - StatusCode=0Ch(Invalid_Param) Activate() method with out dataStorelist; Response - Pass Activate() method with DataStore size (>= maxDSsize); Response - Pass Reactivate() method with DataStore size (>= maxDSsize); Response - Pass Reactivate() method with DataStore size (>= maxDSsize); Response - Pass Reactivate() method with DataStore size (>= maxDSsize); Response - Pass Reactivate() method with DataStore size (>= maxDSsize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (>= maxDssize); Response - StatusCode=00h(Invalid_Param) ReActivate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Opal SSC Feature Set: Single User Mode Check the feature support of Single User Mode Check the support of ReActivate and Erase methods in the MethodID table Get the values of 'SingleUserModeRanges' and 'RangeStattLengthPolicy' from the LockingInfo table Activate() method with SP not included in Locking Template; Response - StatusCode=0Ch(Invalid_Param) Activate() method with RangeN(N=LockingInfo.MaxRanges/2); SyncSessi	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries added to the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - Pass Activate() method without dataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with DataStore table; Response - Pass Reactivate() method with DataStore table; Response - Pass Reactivate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Reactivate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Reactivate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Opal SSC Feature Set: Single User Mode Check the feature support of Single User Mode from Level0_Discovery Check the support of ReActivate and Erase methods in the MethodID table Get the values of 'SingleUserModeRanges' and 'RangeStartLengthPolicy' from the LockingInfo table Activate() method with SP not included in Locking Template; Response - StatusCode=0Ch(Invalid_Param) Activate() method with RangeN(N-LockingInfo.MaxRanges/2); SyncSession - Pass Verify: StartSession to Locking SP as User(N-I)(N-MaxRanges/2); SyncSession - Pass Activate() method with RangeN(N-LockingInfo.MaxRanges/2); SyncSession - Pass Activate() method with RangeN(PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries in the ACE table Verify the values of ACL associated with DataStore in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with oun-align DataStore; Response - StatusCode=O9h(Insufficient_Space) Activate() method with all DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - StatusCode=O9h(Insufficient_Space) Reactivate() method with DataStore size (<= maxDSsize); Response - StatusCode=O9h(Insufficient_Space) Reactivate() method with DataStore size (>= maxDSsize); Response - StatusCode=O9h(Insufficient_Space) Reactivate() method with DataStore size (>= maxDSsize); Response - StatusCode=O0h(Invalid_Param) ReActivate() method with onn-align DataStore; Response - Pass ReActivate() method with Opace and DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Opal SSC Feature Set: Single User Mode Check the feature support of Single User Mode from Level0_Discovery Check the support of ReActivate and Erase methods in the MethodID table Get the values of 'SingleUserModeRanges' and 'Range	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries added to the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (<= maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with adaStore size (>= maxDSSize); Response - StatusCode=00h(Invalid_Param) Activate() method with out dataStore: Response - Pass Reactivate() method with all DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Reactivate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Reactivate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Reactivate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Opal SSC Feature Set: Single User Mode Check the feature support of Single User Mode Check the feature support of Single User Mode Check the support of Reactivate and Erase methods in the MethodID table Get the values of 'SingleUser ModeRanges' and 'RangeStartLengthPolicy' from the LockingInfo table Activate() method with LockingObject not included in Locking table; Response - StatusCode=OCh(Invalid_Param) Activate() method with RangeN(N-LockingInfo.MaxRanges/	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries in the ACE table Verify the values of ACL associated with DataStore in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with oun-align DataStore; Response - StatusCode=O9h(Insufficient_Space) Activate() method with all DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - StatusCode=O9h(Insufficient_Space) Reactivate() method with DataStore size (<= maxDSsize); Response - StatusCode=O9h(Insufficient_Space) Reactivate() method with DataStore size (>= maxDSsize); Response - StatusCode=O9h(Insufficient_Space) Reactivate() method with DataStore size (>= maxDSsize); Response - StatusCode=O0h(Invalid_Param) ReActivate() method with onn-align DataStore; Response - Pass ReActivate() method with Opace and DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Opal SSC Feature Set: Single User Mode Check the feature support of Single User Mode from Level0_Discovery Check the support of ReActivate and Erase methods in the MethodID table Get the values of 'SingleUserModeRanges' and 'Range	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries added to the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (<= maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with adaStore size (>= maxDSSize); Response - StatusCode=00h(Invalid_Param) Activate() method with out dataStore: Response - Pass Reactivate() method with all DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Reactivate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Reactivate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Reactivate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Opal SSC Feature Set: Single User Mode Check the feature support of Single User Mode Check the feature support of Single User Mode Check the support of Reactivate and Erase methods in the MethodID table Get the values of 'SingleUser ModeRanges' and 'RangeStartLengthPolicy' from the LockingInfo table Activate() method with LockingObject not included in Locking table; Response - StatusCode=OCh(Invalid_Param) Activate() method with RangeN(N-LockingInfo.MaxRanges/	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with ACE in the AccessControl table Verify the values of ACL associated with DataStore in the ACEssion to table Verify the values of ACL associated with DataStore in the ACEssion to table Activate() method with DataStore size (<= maxDSSize); Response - Pass Activate() method with DataStore size (<= maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with non-align DataStore; Response - Pass Activate() method with adaStore size (<= maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with all DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with DataStore size (<= maxDSSize); Response - Pass Reactivate() method with non-align DataStore; Response - StatusCode=09h(Insufficient_Space) Reactivate() method with non-align DataStore; Response - Pass Reactivate() method with out dataStore list; Response - Pass ReActivate() method with out dataStore list; Response - Pass ReActivate() method with DataStore size (<= maxDSSize); Response - Pass ReActivate() method with DataStore size (<= maxDSSize); Response - Pass ReActivate() method with DataStore size (<= maxDSSize); Response - Pass ReActivate() method with DataStore size (<= maxDSSize); Response - Pass ReActivate() method with RangeN(N=LockingInfo, MaxRanges/2); SyncSession - StatusCode=0Ch(Invalid	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the values of ACL associated with ACE table Verify the values of ACL associated with DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries in the ACE table Verify the values of ACL associated with DataStore in the AccessControl table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (> maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - StatusCode=00h(Invalid_Param) Activate() method with non-align DataStore; Response - StatusCode=0Ch(Invalid_Param) Activate():The size of dataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate():The size of dataStore size (> maxDSSize); Response - Pass Reactivate() method with DataStore size (> maxDSSize); Response - Pass Reactivate() method with DataStore size (> maxDSSize); Response - Pass Reactivate() method with DataStore size (> maxDSSize); Response - Pass Reactivate() method with DataStore size (> maxDSSize); Response - Pass Reactivate() method with DataStore size (> maxDSSize); Response - Pass Reactivate() method with DataStore size (> maxDSSize); Response - Pass ReActivate():The size of dataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Opal SSC Feature Set: Single User Mode Check the feature support of Ractivate and Erase methods in the MethodID Table Get the values of 'SingleUser ModeRanges' and 'RangeStartLengthPolicy' from the LockingInfo table Activate() method with SP not included in Locking Template; Respon	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore Teleature Set from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries in the ACE table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (- maxDSSize); Response - Pass Activate() method with DataStore size (- maxDSSize); Response - Pass Activate() method with non-align DataStore; Response - StatusCode=00h(insufficient_Space) Activate() method with non-align DataStore; Response - Pass Activate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with DataStore size (- maxDSSize); Response - Pass Reactivate() method with DataStore size (- maxDSSize); Response - Pass Reactivate() method with DataStore size (- maxDSSize); Response - Pass Reactivate() method with non-align DataStore; Response - StatusCode=00h(insufficient_Space) Reactivate() method with non-align DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Opal SSC Feature Set: Single User Mode Check the feature support of Single User Mode Check the feature support of Single User Mode Check the feature support of Single User M	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore table; Response - Pass Compare the number of Additional DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries added to the ACE table Verify the values of ACL associated with ACE table Verify the values of ACL associated with ACE table Verify the values of ACL associated with ACE table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (< maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Activate() method with DataStore size (> maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - Pass Activate() method with DataStore size (> maxDSSize); Response - Pass Reactivate() method with DataStore size (< maxDSSize); Response - Pass Reactivate() method with DataStore size (> maxDSSize); Response - Pass Reactivate() method with DataStore size (> maxDSSize); Response - Pass Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=09h(Insufficient_Space) Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=00h(Invalid_Param) Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=00h(Invalid_Param) Reactivate() method with out adaStore size (> maxDSSize); Response - StatusCode=00h(Invalid_Param) Reactivate() method with DataStore size (> maxDSSize); Response - StatusCode=00h(Invalid_Param) Activate() method with ParaMaramages and RangeStartLengthPolicy from the LockingInfo table Get the values of 'SingleUser ModeRanges' and 'RangeStartLengthPolicy' from the LockingInfo table Activate() method with RangeN(N=LockingInfo MaxRanges/2); SyncSes	PASS PASS PASS PASS PASS PASS PASS PASS
End Session - Request End Session - Response Opal SSC Feature Set: Additional DataStore Tables Verify the support of Additional DataStore Feature Set from Level0_Discovery Activate() method with all DataStore Teleature Set from Level0_Discovery Check the number of new entries added to the ACE table Verify the contents of new entries in the ACE table Verify the contents of new entries in the ACE table Verify the values of ACL associated with DataStore in the Table table and the maximum number of DataStore tables from Level0_Discovery Check the number of new entries in the ACE table Verify the values of ACL associated with DataStore in the AccessControl table Activate() method with DataStore size (- maxDSSize); Response - Pass Activate() method with DataStore size (- maxDSSize); Response - Pass Activate() method with non-align DataStore; Response - StatusCode=00h(insufficient_Space) Activate() method with non-align DataStore; Response - Pass Activate() method with DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery ReActivate() method with DataStore size (- maxDSSize); Response - Pass Reactivate() method with DataStore size (- maxDSSize); Response - Pass Reactivate() method with DataStore size (- maxDSSize); Response - Pass Reactivate() method with non-align DataStore; Response - StatusCode=00h(insufficient_Space) Reactivate() method with non-align DataStore is equal to the 'Maximum total size of DataStore' from Level0_Discovery Opal SSC Feature Set: Single User Mode Check the feature support of Single User Mode Check the feature support of Single User Mode Check the feature support of Single User M	PASS PASS PASS PASS PASS PASS PASS PASS

Activate() method with SingleUserMode for Range1 and Range2 if LockingSP = mfg state; Response - Succeed	PASS
The method shall have no effect: 'SingleUserModeRanges' and 'RangeStartLengthPolicy' keep the previous values	PASS
Activate() method with SingleUserMode for Range1 and Range2 after LockingSP.Revert; Response - Pass	PASS
Verify 'SingleUserModeRanges'=Range1/Range2 and 'RangeStartLengthPolicy'=0 from the LockingInfo table	PASS
Verify 'Policy'=0; 'All'=0; 'Any'=1 from Level0_Discovery	PASS
Locking_Range1.Set Request in LockingSP as User2; Response - Pass	PASS
Locking_Range1.Set Request in LockingSP as Admin1; Response - StatusCode = 01h(Not_Authorized)	PASS
Activate() method with entire Locking table and 'RangeStartLengthPolicy'=0; Response - Pass	PASS
Activate w/ entireLocking; Verify 'SingleUserModeRanges'=EntireLocking and 'RangeStartLengthPolicy'=0 from the LockingInfo table	PASS
Activate w/ entireLocking: Verify 'Policy'=0; 'All'=1; 'Any'=1 from Level0_Discovery	PASS
Activate w/ entireLocking: Range1-GlobalRange.Set Request in LockingSP as User1-(N+1); Response - StatusCode = 01h(Not_Authorized)	PASS
Activate w/ entireLocking: GlobalRange-RangeN.Set Request in LockingSP as User1-(N+1); Response - Pass	PASS
Activate() method with all Locking Objects and 'RangeStartLengthPolicy'=0; Response - Pass	PASS
Activate w/ allLockingObj: Verify 'SingleUserModeRanges'=all objects and 'RangeStartLengthPolicy'=0 from the LockingInfo table	PASS
Activate w/ allLockingObj: Verify 'Policy'=0; 'All'=1; 'Any'=1 from Level0_Discovery	PASS
Activate w/ allLockingObj: Range1-GlobalRange.Set Request in LockingSP as User1-(N+1); Response - StatusCode = 01h(Not_Authorized)	PASS
Activate w/ allLockingObj: GlobalRange-RangeN.Set Request in LockingSP as User1-(N+1); Response - Pass	PASS
ReActivate() method with Read/WriteLockEnabled=True; Response - StatusCode=3Fh(Fail)	PASS
ReActivate() method with ReadLockEnabled=True; Response - StatusCode=3Fh(Fail)	PASS
ReActivate() method with WriteLockEnabled=True; Response - StatusCode=3Fh(Fail)	PASS
ReActivate() method with LockingObject not included in Locking table; Response - StatusCode=0Ch(Invalid_Param)	PASS
ReActivate() method with RangeN(N=LockingInfo.MaxRanges/2); Response - Pass	PASS
Verify: StartSession to Locking SP as UserN(N=MaxRanges/2); SyncSession - StatusCode=01h(Not_Authorized)	PASS
Verify: StartSession to Locking SP as User(N+1)(N=MaxRanges/2); SyncSession - Pass	PASS
ReActivate() method with RangeN(N=LockingInfo.MaxRanges); Response - Pass	PASS
Verify: StartSession to Locking SP as UserN(N=MaxRanges); SyncSession - StatusCode=01h(Not_Authorized)	PASS
Verify: StartSession to Locking SP as User(N+1)(N=MaxRanges); SyncSession - Pass	PASS
ReActivate() with Admin1PIN=omitted; Response - Pass	PASS
ReActivate() w/ Admin1PIN=omitted effect: The session - Abort (no data returned)	PASS
ReActivate() w/ Admin1PIN=omitted effect: The LifeCycleState of the LockingSP remains the same	PASS
ReActivate() w/ Admin1PIN=omitted effect: The value of 'C_PIN_Admin1.PIN' remains at their current values	PASS
ReActivate() w/ Admin1PIN=omitted effect: RangeStart and RangeLength remain at their current values	PASS
ReActivate() w/ Admin1PIN=omitted effect: The media encryption keys remain at their current values	PASS
ReActivate() with Admin1PIN; Response - Pass	PASS
ReActivate() w/ Admin1PIN effect: The session - Abort (no data returned)	PASS
ReActivate() w/ Admin1PIN effect: The LifeCycleState of the LockingSP remains the same	PASS
· · · · · · · · · · · · · · · · · · ·	
ReActivate() w/ Admin1PIN effect: The value of 'C_PIN_Admin1.PIN' is new AdminPIN	PASS
ReActivate() w/ Admin1PIN effect: RangeStart and RangeLength remain at their current values	PASS
ReActivate() w/ Admin1PIN effect: The media encryption keys remain at their current values	PASS
ReActivate() method with empty ObjList and 'RangeStartLengthPolicy'=0; Response - pass	PASS
ReActivate() w/ emptyObj and RSLP=0 effect: The session - Abort (no data returned)	PASS
ReActivate() w/ emptyObj and RSLP=0 effect: The LifeCycleState of the LockingSP remains the same	PASS
ReActivate() w/ emptyObj and RSLP=0 effect: The value of 'C_PIN_Admin1.PIN' remains at their current values	PASS
ReActivate() w/ emptyObj and RSLP=0 effect: RangeStart and RangeLength remain at their current values	PASS
$Re Activate () \ w/empty Obj \ and \ RSLP=0 \ effect: The \ media \ encryption \ keys \ remain \ at \ their \ current \ values$	PASS
Verify 'SingleUserModeRanges'=empty and 'RangeStartLengthPolicy'=1 from the LockingInfo table	PASS
Verify 'Policy'=1; 'All'=0; 'Any'=0 from Level0_Discovery	PASS
ReActivate() method with empty ObjList and 'RangeStartLengthPolicy'=1; Response - pass	PASS
ReActivate() w/ emptyObj and RSLP=1 effect: The session - Abort (no data returned)	PASS
ReActivate() w/ emptyObj and RSLP=1 effect: The LifeCycleState of the LockingSP remains the same	PASS
ReActivate() w/ emptyObj and RSLP=1 effect: The value of 'C_PIN_Admin1.PIN' remains at their current values	PASS
ReActivate() w/ emptyObj and RSLP=1 effect: RangeStart and RangeLength remain at their current values	PASS
	PASS
ReActivate() w/ emptyObj and RSLP=1 effect: The media encryption keys remain at their current values	
Verify 'SingleUserModeRanges'=empty and 'RangeStartLengthPolicy'=1 from the LockingInfo table	PASS
Verify 'Policy'=1; 'All'=0; 'Any'=0 from Level0_Discovery	PASS
Reactivate() method with SingleUserMode for Range1 and Ragne2; Response - Pass	PASS
ReActivate() w/ Range1/2 effect: The session - Abort (no data returned)	PASS
ReActivate() w/ Range1/2 and RSLP=0 effect: The LifeCycleState of the LockingSP remains the same	PASS
ReActivate() w/ Range1/2 and RSLP=0 effect: The value of 'C_PIN_Admin1.PIN' remains at their current values	PASS
ReActivate() w/ Range 1/2 and RSLP=0 effect: RangeStart and RangeLength remain at their current values	PASS
ReActivate() w/ Range1/2 and RSLP=0 effect: The media encryption keys remain at their current values	PASS
, ,	
Verify 'SingleUserModeRanges' and 'RangeStartLengthPolicy' from the LockingInfo table	PASS
Verify 'Policy'=0; 'All'=0; 'Any'=1 from Level0_Discovery	PASS
ReActivate() method with entire Locking table and 'RangeStartLengthPolicy'=0; Response - Pass	PASS
ReActivate w/ entireLocking: The session - Abort (no data returned)	PASS
ReActivate w/ entireLocking: The LifeCycleState of the LockingSP remains the same	
9 ,	PASS
ReActivate w/ entireLocking: The value of 'C_PIN_Admin1.PIN' remains at their current values	PASS
ReActivate w/ entireLocking: The media encryption keys remain at their current values	PASS
ReActivate w/ entireLocking: Verify 'SingleUserModeRanges'=EntireLocking and 'RangeStartLengthPolicy'=0 from the LockingInfo table	PASS
ReActivate w/ entireLocking: Verify 'Policy'=0; 'All'=1; 'Any'=1 from Level0_Discovery	PASS
ReActivate w/ entireLocking: Range1-GlobalRange.Set Request in LockingSP as User1-(N+1); Response - StatusCode = 01h(Not_Authorized)	PASS
ReActivate w/ entireLocking: GlobalRange-RangeN.Set Request in LockingSP as User1-(N+1); Response - Pass	PASS
ReActivate() method with all Locking Objects and 'RangeStartLengthPolicy'=0; Response - Pass	PASS
ReActivate w/ allLockingObj: The session - Abort (no data returned)	PASS
ReActivate w/ allLockingObj: The LifeCycleState of the LockingSP remains the same	PASS
ReActivate w/ allLockingObj: The value of 'C_PIN_Admin1.PIN' remains at their current values	PASS
ReActivate w/ allLockingObj: The media encryption keys remain at their current values	PASS
ReActivate w/ allLockingObj: Verify 'SingleUserModeRanges'=all objects and 'RangeStartLengthPolicy'=0 from the LockingInfo table	PASS
ReActivate w/ allLockingObj: Verify 'Policy'=0; 'All'=1; 'Any'=1 from Level0_Discovery	
	PASS
ReActivate w/ allLockingObj: Range1-GlobalRange.Set Request in LockingSP as User1-(N+1); Response - StatusCode = 01h(Not_Authorized)	
ReActivate w/ allLockingObj: Range1-GlobalRange.Set Request in LockingSP as User1-(N+1); Response - StatusCode = 01h(Not_Authorized) ReActivate w/ allLockingObj: GlobalRange-RangeN.Set Request in LockingSP as User1-(N+1); Response - Pass	PASS
ReActivate w/ allLockingObj: GlobalRange-RangeN.Set Request in LockingSP as User1-(N+1); Response - Pass	PASS PASS PASS
	PASS PASS

Erase() effect: Locking_Range(X-1).Erase Request; Response - Pass	PASS	
Erase() effect: Read/WriteLockEnabled and Read/WriteLocked = 0	PASS	
Erase() effect: RangeStart and RangeLength are not changed	PASS	
Erase() effect: Generate a new media encryption key for LBA range	PASS	
Erase(): C_PIN.UserX = empty	PASS	
Erase(): Tries = 0 from the C_PIN table	PASS	
Feature Set: Block SID Authentication	PASS	
Check the support of Block SID 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: statusCode=01h	PASS	
Authenticate - SID (authority UID); Authenticate Response - StatusCode/AuthStatus=00h/00h(SUCCESS/False)	PASS	
The Tries column of the SID C PIN shall not be incremented after Block SID Authentication command	PASS	
Clear Events: Revert AdminSP	PASS	
Check SID Blocked State(=0) after Revert	PASS	
Clear Events: Power Cycle	PASS	
Check SID Blocked State(=0) after power cycle	PASS	
Block SID Authentication command with Hardware Reset bit=1: Pass	N/A	
Check SID Blocked State(=0) after Hardware Reset	N/A	
Block SID Authentication command with Hardware Reset(PERST#) bit=1: Pass	PASS	
Check SID Blocked State(=0) after Hardware Reset(PERST#)	PASS	
Subsequent invocation of Block SID Authentication command: Fail with 'Other Invalid Command Parameter'	PASS	
Check Locking SP Freeze Lock State/Supported bit from Level0_Discovery	PASS	
Verify Locking SP Freeze Lock State bit and 'Frozen' value in the SP table	PASS	
Verify SID State Value(=1) if SID C PIN credential is NOT the same as the value of the MSID C PIN credential	PASS	
Verify SID State Value(=0) if SID C PIN credential is the same as the value of the MSID C PIN credential	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	
# Tested	1184	
# Passed	1184	
# Failed	0	
# Not Tested	46	
Script End Date: Mon	February 0 2024	
Time: 12:57:32 PM		
Total Runtime:	0:46:10	