|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_001 |
| **Test Objective** | Establishing Connection with EndpointUrl |
| **Reference** | OPC-UA,Part-6-Mappings#section-7.1.3  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
|  | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a HelloMessage Message containing ProtocolVersion indicating value 0,  ReceivedBufferSize indicating value largest ReceivedMessageChunk size,  SendBufferSize indicating value largest SendMessageChunk size,  MaxMessageSize indicating value maximum RequestMessage size,  MaxChunkCount indicating value maximum RequestMessageChunk,  EndpointUrl indicating value URL;  }  then {  the IUT sends a AckMessage containing ProtocolVersion indicating value 0,  ReceivedBufferSize indicating value largest ReceivedMessageChunk size,  SendBufferSize indicating value largest SendMessageChunk size,  MaxMessageSize indicating value maximum RequestMessage size,  MaxChunkCount indicating value maximum RequestMessageChunk ;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_002 |
| **Test Objective** | Establishing Connection without EndpointUrl |
| **Reference** | OPC-UA,Part-6-Mappings#section-7.1.3  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
|  | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a HelloMessage Message containing ProtocolVersion indicating value 0,  ReceivedBufferSize indicating value largest ReceivedMessageChunk size,  SendBufferSize indicating value largest SendMessageChunk size,  MaxMessageSize indicating value maximum RequestMessage size,  MaxChunkCount indicating value maximum RequestMessageChunk,  EndpointUrl indicating value URL;  }  then {  the IUT sends a AckMessage containing ProtocolVersion indicating value 0,  ReceivedBufferSize indicating value largest ReceivedMessageChunk size,  SendBufferSize indicating value largest SendMessageChunk size,  MaxMessageSize indicating value maximum RequestMessage size,  MaxChunkCount indicating value maximum RequestMessageChunk;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_03 |
| **Test Objective** | Find Servers |
| **Reference** | OPC-UA,Part-4  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a FindServersRequest containing requestHeader,endpointUrl;  }  then {  the IUT sends a FindServersResponse containing  responseHeader containing serviceResults indicating value 0,  ListOfApplicationDescription indicating value ListofServers;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_004 |
| **Test Objective** | Open a communication channel with None Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.5.2  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a OpenSecureChannelRequest containing RequestHeader,  requestType indicating value ISSUE\_0, /\*SecurityTokenType\*/  securityMode indicating value NONE, /\*SecurityType\*/  clientNonce, /\*A random number not used in any request\*/  requestedLifeTime indicating value 6000;  }  then {  the IUT sends a OpenSecureChannelResponse containing  responseHeader containing serviceResults indicating value 0,  securityToken,/\*SecurityToken issued by server\*/  channelId, /\*A unique identifier for secure channel\*/  tokenId;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_005 |
| **Test Objective** | Open a communication channel with SignOnly Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.5.2  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a OpenSecureChannelRequest containing RequestHeader,  requestType indicating value ISSUE\_0, /\*SecurityTokenType\*/  securityMode indicating value SIGN\_2, /\*SecurityType\*/  clientNonce, /\*A random number not used in any request\*/  requestedLifeTime indicating value 6000;  }  then {  the IUT sends a OpenSecureChannelResponse containing  responseHeader containing serviceResults indicating value 0,  securityToken,/\*SecurityToken issued by server\*/  channelId, /\*A unique identifier for secure channel\*/  tokenId;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_006 |
| **Test Objective** | Open a communication channel with SignAndEncrypt Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.5.2  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a OpenSecureChannelRequest containing RequestHeader,  requestType indicating value ISSUE\_0, /\*SecurityTokenType\*/  securityMode indicating value SIGN\_2, /\*SecurityType\*/  clientNonce, /\*A random number not used in any request\*/  requestedLifeTime indicating value 6000;  }  then {  the IUT sends a OpenSecureChannelResponse containing  responseHeader containing serviceResults indicating value 0,  securityToken,/\*SecurityToken issued by server\*/  channelId, /\*A unique identifier for secure channel\*/  tokenId;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_007 |
| **Test Objective** | Creating a Session for CLIENT application with None Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.6.2  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage and  the IUT sends a OpenSecureResponse on receiving OpenSecureRequest containing  securityMode indicating value NONE;  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a CreateSessionRequest containing requestHeader,  clientDescription containing applicationName indicating value CLIENT\_1, /\* Application is Client\*/  endpointUrl, /\*Network Address for client\*/  sessionName, /\*Name of the Session \*/  clientNonce;;  }  then {  the IUT sends a CreateSessionResponse containing  responseHeader containing serviceResults indicating value 0,  sessionId, /\*A unique NodeId assigned by Server\*/  authenticationToken;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_008 |
| **Test Objective** | Creating a Session for CLIENT application with SignOnly Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.6.2  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage and  the IUT sends a OpenSecureResponse on receiving OpenSecureRequest containing  securityMode indicating value SIGN\_2;  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a CreateSessionRequest containing requestHeader,  clientDescription containing applicationName indicating value CLIENT\_1, /\* Application is Client\*/  endpointUrl, /\*Network Address for client\*/  sessionName, /\*Name of the Session \*/  clientNonce;;  }  then {  the IUT sends a CreateSessionResponse containing  responseHeader containing serviceResults indicating value 0,  sessionId, /\*A unique NodeId assigned by Server\*/  authenticationToken;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_008 |
| **Test Objective** | Creating a Session for CLIENT application with SignAndEncrypt Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.6.2  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage and  the IUT sends a OpenSecureResponse on receiving OpenSecureRequest containing  securityMode indicating value SIGNANDENCRYPT\_3;  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a CreateSessionRequest containing requestHeader,  clientDescription containing applicationName indicating value CLIENT\_1, /\* Application is Client\*/  endpointUrl, /\*Network Address for client\*/  sessionName, /\*Name of the Session \*/  clientNonce;;  }  then {  the IUT sends a CreateSessionResponse containing  responseHeader containing serviceResults indicating value 0,  sessionId, /\*A unique NodeId assigned by Server\*/  authenticationToken;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_009 |
| **Test Objective** | Creating a Session for CLIENTANDSERVER application with None Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.6.2  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage and  the IUT sends a OpenSecureResponse on receiving OpenSecureRequest containing  securityMode indicating value NONE\_1;  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a CreateSessionRequest containing requestHeader,  clientDescription containing applicationName indicating value CLIENTANDSERVER, /\* Application is CLIENTANDSERVER\*/  endpointUrl, /\*Network Address for client\*/  sessionName, /\*Name of the Session \*/  clientNonce;;  }  then {  the IUT sends a CreateSessionResponse containing  responseHeader containing serviceResults indicating value 0,  sessionId, /\*A unique NodeId assigned by Server\*/  authenticationToken;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_010 |
| **Test Objective** | Creating a Session for CLIENTANDSERVER application with SignOnly Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.6.2  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage and  the IUT sends a OpenSecureResponse on receiving OpenSecureRequest containing  securityMode indicating value SIGN\_2;  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a CreateSessionRequest containing requestHeader,  clientDescription containing applicationName indicating value CLIENTANDSERVER, /\* Application is CLIENTANDSERVER\*/  endpointUrl, /\*Network Address for client\*/  sessionName, /\*Name of the Session \*/  clientNonce;;  }  then {  the IUT sends a CreateSessionResponse containing  responseHeader containing serviceResults indicating value 0,  sessionId, /\*A unique NodeId assigned by Server\*/  authenticationToken;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_011 |
| **Test Objective** | Creating a Session for CLIENTANDSERVER application with SIGNANDENCRYPT Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.6.2  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage and  the IUT sends a OpenSecureResponse on receiving OpenSecureRequest containing  securityMode indicating value SIGNANDENCRYPT\_3;  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a CreateSessionRequest containing requestHeader,  clientDescription containing applicationName indicating value CLIENTANDSERVER, /\* Application is CLIENTANDSERVER\*/  endpointUrl, /\*Network Address for client\*/  sessionName, /\*Name of the Session \*/  clientNonce;;  }  then {  the IUT sends a CreateSessionResponse containing  responseHeader containing serviceResults indicating value 0,  sessionId, /\*A unique NodeId assigned by Server\*/  authenticationToken;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_012 |
| **Test Objective** | Creating a Session for SERVER application with None Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.6.2  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage and  the IUT sends a OpenSecureResponse on receiving OpenSecureRequest containing  securityMode indicating value NONE\_1;  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a CreateSessionRequest containing requestHeader,  clientDescription containing applicationName indicating value SERVER, /\* Application is SERVER\*/  endpointUrl, /\*Network Address for SERVER\*/  sessionName, /\*Name of the Session \*/  clientNonce;;  }  then {  the IUT sends a CreateSessionResponse containing  responseHeader containing serviceResults indicating value 0,  sessionId, /\*A unique NodeId assigned by Server\*/  authenticationToken;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_013 |
| **Test Objective** | Creating a Session for SERVER application with SignOnly Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.6.2  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage and  the IUT sends a OpenSecureResponse on receiving OpenSecureRequest containing  securityMode indicating value SIGN\_2;  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a CreateSessionRequest containing requestHeader,  clientDescription containing applicationName indicating value SERVER, /\* Application is SERVER\*/  endpointUrl, /\*Network Address for SERVER\*/  sessionName, /\*Name of the Session \*/  clientNonce;;  }  then {  the IUT sends a CreateSessionResponse containing  responseHeader containing serviceResults indicating value 0,  sessionId, /\*A unique NodeId assigned by Server\*/  authenticationToken;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_014 |
| **Test Objective** | Creating a Session for SERVER application with SIGNANDENCRYPT Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.6.2  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage and  the IUT sends a OpenSecureResponse on receiving OpenSecureRequest containing  securityMode indicating value SIGNANDENCRYPT\_3;  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a CreateSessionRequest containing requestHeader,  clientDescription containing applicationName indicating value SERVER, /\* Application is SERVER\*/  endpointUrl, /\*Network Address for SERVER\*/  sessionName, /\*Name of the Session \*/  clientNonce;;  }  then {  the IUT sends a CreateSessionResponse containing  responseHeader containing serviceResults indicating value 0,  sessionId, /\*A unique NodeId assigned by Server\*/  authenticationToken;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_015 |
| **Test Objective** | Endpoints supported by SERVER application with NONE Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.4.4  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage and  the IUT sends a OpenSecureResponse on receiving OpenSecureRequest containing  securityMode indicating value NONE\_1;  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a GetEndpointsRequest containing  requestHeader,  endpointUrl, /\*Network Address for SERVER\*/  localIds, /\*list of locales to use\*/  profileUris /\*List of Transport Profile that the returned Endpoints shall support\*/;  }  then {  the IUT sends a GetEndpointsResponse containing  responseHeader containing serviceResults indicating value 0,  endpointDescription;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_016 |
| **Test Objective** | Endpoints supported by SERVER application with SignOnly Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.4.4  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage and  the IUT sends a OpenSecureResponse on receiving OpenSecureRequest containing  securityMode indicating value SIGN\_2;  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a GetEndpointsRequest containing  requestHeader,  endpointUrl, /\*Network Address for SERVER\*/  localIds, /\*list of locales to use\*/  profileUris /\*List of Transport Profile that the returned Endpoints shall support\*/;  }  then {  the IUT sends a GetEndpointsResponse containing  responseHeader containing serviceResults indicating value 0,  endpointDescription;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_017 |
| **Test Objective** | Endpoints supported by SERVER application with SignandEncrypt Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.4.4  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage and  the IUT sends a OpenSecureResponse on receiving OpenSecureRequest containing  securityMode indicating value SIGNANDENCRYPT\_3;  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a GetEndpointsRequest containing  requestHeader,  endpointUrl, /\*Network Address for SERVER\*/  localIds, /\*list of locales to use\*/  profileUris /\*List of Transport Profile that the returned Endpoints shall support\*/;  }  then {  the IUT sends a GetEndpointsResponse containing  responseHeader containing serviceResults indicating value 0,  endpointDescription;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_018 |
| **Test Objective** | Activate Session supported by SERVER application with NONE Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.6.3  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage and  the IUT sends a OpenSecureResponse on receiving OpenSecureRequest containing securityMode indicating value NONE; /\*SecurityType\*/ and  the IUT sends a CreateSessionResponse on receiving CreateSessionRequest containing requestHeader,  clientDescription containing applicationName indicating value CLIENT, /\* Application is SERVER\*/  endpointUrl, /\*Network Address for SERVER\*/  sessionName, /\*Name of the Session \*/  clientNonce;;  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a ActivateSessionRequest containing  requestHeader,  userIdToken,  usertokenSignature;  }  then {  the IUT sends a ActivateSessionResponse containing  responseHeader containing serviceResults indicating value 0,  serverNonce, /\*A random number not used in any request\*/  results, /\*List of Validation results\*/  diagnosticInfos;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_019 |
| **Test Objective** | Activate Session supported by SERVER application with SignOnly Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.6.3  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage and  the IUT sends a OpenSecureResponse on receiving OpenSecureRequest containing securityMode indicating value SIGN\_2; /\*SecurityType\*/ and  the IUT sends a CreateSessionResponse on receiving CreateSessionRequest containing  requestHeader,  clientDescription containing applicationName indicating value CLIENT, /\* Application is SERVER\*/  endpointUrl, /\*Network Address for SERVER\*/  sessionName, /\*Name of the Session \*/  clientNonce;;  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a ActivateSessionRequest containing  requestHeader,  userIdToken,  usertokenSignature;  }  then {  the IUT sends a ActivateSessionResponse containing  responseHeader containing serviceResults indicating value 0,  serverNonce, /\*A random number not used in any request\*/  results, /\*List of Validation results\*/  diagnosticInfos;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_020 |
| **Test Objective** | Activate Session supported by SERVER application with SignAndEncrypt Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.6.3  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage and  the IUT sends a OpenSecureResponse on receiving OpenSecureRequest containing securityMode indicating value SIGNANDENCRYPT\_3; /\*SecurityType\*/ and  the IUT sends a CreateSessionResponse on receiving CreateSessionRequest containing  requestHeader,  clientDescription containing applicationName indicating value CLIENT, /\* Application is SERVER\*/  endpointUrl, /\*Network Address for SERVER\*/  sessionName, /\*Name of the Session \*/  clientNonce;;  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a ActivateSessionRequest containing  requestHeader,  userIdToken,  usertokenSignature;  }  then {  the IUT sends a ActivateSessionResponse containing  responseHeader containing serviceResults indicating value 0,  serverNonce, /\*A random number not used in any request\*/  results, /\*List of Validation results\*/  diagnosticInfos;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_021 |
| **Test Objective** | Close SecureChannel supported by SERVER application with NONE Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.5.3  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage and  the IUT sends a OpenSecureResponse on receiving OpenSecureRequest containing  securityMode indicating value NONE\_1;  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a CloseSecureChannelRequest containing requestHeader,  secureChannelId;  }  then {  the IUT sends a CloseSecureChannelResponse containing  responseHeader containing serviceResults indicating value 0;;  }  } | |
| **Final Conditions** | |
|  | |

|  |  |
| --- | --- |
| **TP Id** | TP\_OPC\_UA\_Server\_022 |
| **Test Objective** | Terminate a Session with NONE Security |
| **Reference** | OPC-UA,Part-4-Services#section-5.6.4  https://opcfoundation.org/developer-tools/specifications-unified-architecture |
| **PICS Selection** |  |
| **Initial Conditions** | |
| with {  the IUT sends a AckMessage on receiving HelloMessage and  the IUT sends a OpenSecureResponse on receiving OpenSecureRequest containing securityMode indicating value NONE\_1; /\*SecurityType\*/ and  the IUT sends a CreateSessionResponse on receiving CreateSessionRequest containing  requestHeader,  clientDescription containing applicationName indicating value CLIENT, /\* Application is SERVER\*/  endpointUrl, /\*Network Address for SERVER\*/  sessionName, /\*Name of the Session \*/  clientNonce;; and /\*A random number not used in any request\*/  the IUT sends a ActivateSessionResponse on receiving ActivateSessionRequest  } | |
| **Expected Behaviour** | |
| ensure that {  when {  the IUT receives a CloseSessionRequest containing requestHeader,  deleteSubscriptions;  }  then {  the IUT sends a CloseSessionResponse containing  responseHeader containing serviceResults indicating value 0;;  }  } | |
| **Final Conditions** | |
|  | |