이더넷 SOME/IP 통신 평가 사양서

|  |  |
| --- | --- |
| **Author:** | Jangsoo Kim, KOEM RM |
|  |  |
| **Revision:** | 3 |
| **Status:** | <~~draft~~/ released /~~obsolete~~> |
| **File:** |  |

**History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Revision** | **Date** | **Author, Editor** | **Approver** | **Reason** |
| 3 | 2022-08-12 | Jangsoo Kim | Cheeyoung Yoon | Initial version:derived from Version 3 of customer specification |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 목적

이 기술 사양은 이더넷 네트워크에 연결된 ECU의 이더넷 SOME/IP 테스트를 정의합니다. ECU 설계 명세와 이 문서의 내용이 다른 경우, ECU 설계 명세에 쓰여진 내용이 이 문서보다 우선합니다.

# 문서 범위

이 문서는 (표10. HMC/KMC 출판물)을 기반으로 이더넷 네트워크에 연결된 ECU 내의 이더넷 SOME/IP & SD 통신에 대한 기본 테스트 사례를 설명합니다.

이 문서는 이더넷 네트워크 및 각 테스트 조건에 연결된 ECU 테스트를위한 기본 테스트 환경을 정의합니다. 이 문서는 아래와 같이 두가지 테스트방법을 정의합니다:

* Black-box test 방식 : 테스트 대상 제어기 (DUT) 에서 별도 테스트 어플리케이션 실행 불필요
* White-box test 방식 : 테스트 대상 제어기 (DUT) 에 대한 심화된 평가를 위해 DUT에 특정 SOME/IP 서비스 기반 테스트 어플리케이션 실행 필요

# 용어, 정의 & 약어

## 약어

표 1 용어 정의

| **약어** | **설명** |
| --- | --- |
| CAN | Controller Area Network |
| DUT | Device Under Test |
| ECU | Electronic Control Unit |
| ETS | Enhanced Testability Service |
| ID | Identifier |
| IP or IPv4 | Internet Protocol Version 4 |
| PDU | Protocol Data Unit |
| SOME/IP | Service Oriented MiddlewarE over IP |
| SD | Service Discovery |
| SP | Service Primitive |
| SW | Software |
| TCP | Transmission Control Protocol |
| TP | Transport Protocol |
| TS | Test System |
| TTL | Time To Live |
| UDP | User Datagram Protocol |

## 시험 항목 구조 정의

테스트 그룹의 각 시험 항목은 하나 이상의 설계 요구사항을 확인하기 위해 정의됩니다. 시험 항목 별 명확한 식별을 위해 각각 고유한 테스트 케이스 ID로 설정합니다.

표 2 시험항목 구조

| **Item** | **Description** |
| --- | --- |
| Purpose | 시험 항목의 목적에 대한 간단한 설명 |
| Preconditions | 적절한 테스트 실행에 필요한 DUT(제어기) 조건 목록 |
| Test Setup | 테스트 설정 |
| Test Variables | 적절한 테스트 실행에 필요한 DUT(제어기) 구성 항목, 시간 초과 등과 같은 사전 정의 된 테스트 매개 변수 |
| Test Procedure | 적용 테스트 설정 및 측정을 위한 평가 절차 정의 |
| Verification | 시험 항목의 pass/fail 판정을 결정하기 위한 평가 기준 |
| Reference | ES96595-11 이더넷 SOME/IP 통신 설계 사양서 요구사항 ID 에 대한 참조 |
| Notes | 부가 설명 |

# 시험 환경

## 시험 설정

이 장에서는 5장에 정의된 시험 항목들에 사용되는 시험 설정을 정의합니다.

### Blackbox Test

Blackbox Test 는 아래와 같이 두 부분으로 구성된 간단한 시험 설정이 필요합니다.

DUT(제어기)와 SOME/IP 를 통해 DUT(제어기)를 관찰 및 평가할 수 위한 Tester(시험 장비) 로 구성되어 있습니다. 제어기와 평가장비는 이더넷을 통해 연결되어야 합니다.

SOME/IP

**Tester**

**DUT**

Ethernet

(SOME/IP)

TCP

UDP

그림 1 시험 설정 - Blackbox Test

#### 초기 조건

Blackbox Test 를 위한 DUT(제어기)의 초기조건은 다음과 같습니다.

* DUT(제어기) 또는 적어도 SOME/IP 스택 및 인터페이스가 reset 되어야 합니다.

### Whitebox Test (Enhanced Testability Service)

Whitebox Test 설정은 DUT(제어기) 에서 평가를 위한 추가 응용프로그램이 필요하다는 점을 제외하면 이전 설정과 유사합니다. 이 응용 프로그램은 일부 테스트 시나리오에 필요한 실행중인 서비스의 중지 또는 제어기 종료 및 재시작과 같은 특정 시험환경을 트리거하는데 사용됩니다.

Enhanced Testability Service 사양에 대해서는 4.3 절을 참조하여야 합니다.

SOME/IP

**Tester**

**DUT**

Ethernet

(SOME/IP)

TCP

UDP

ETS

그림 2 시험 설정 - Whitebox Test (ETS)

#### 초기 조건

Whitebox Test 를 위한 DUT(제어기)의 초기조건은 다음과 같습니다.

* DUT(제어기) 또는 적어도 SOME/IP 스택 및 인터페이스가 reset 되어야 하며, ETS 가 실행중이어야 합니다.

### CAN Message Transport

SOME/IP

**Tester**

**DUT**

Ethernet

(SOME/IP)

TCP/IP

UDP

ETS

CAN - Bus

CAN

CAN - Bus

그림 3 시험 설정 – CAN 메시지 전송

## 시험 변수

### 설정 파라미터

표 3 설정 파라미터

| **파라미터** | **설명** |
| --- | --- |
| TimeoutStartupSD | 제어기가 전원을 켠 후 Tester(시험장비)가 최초의 SOME/IP-SD 메시지를 기다리는 최대시간 |
| TimeoutFindService | Specifies the maximum duration the Tester shall wait for Find Service Entry. Shall cover initial wait and repetition phase where Find Service entries are allowed to be sent. |
| TimeoutOfferService | DUT(제어기)의 CYCLIC\_OFFER\_DELAY 설정에 따라 Tester(시험장비)가 다음 주기적으로 전송된 Offer Service Entry 를 기다리는 최대 시간 |
| TimeoutStopOfferService | DUT(제어기) 가 전송한 Stop Offer Service Entry 를 Tester (시험장비)가 기다리는 최대 시간 |
| TimeoutRequest | DUT(제어기)가 전송한 request 메시지를 Tester(시험장비) 가 기다리는 최대 시간 |
| TimeoutResponse | DUT(제어기)가 선행 request에 대한 응답으로 전송한 response 메시지를 Tester(시험장비) 가 기다리는 최대 시간 |
| TimeoutSubscribe | DUT(제어기)가 전송한 Subscribe Eventgroup Entry 를 Tester(시험장비) 가 기다리는 최대 시간 |
| TimeoutSubscribeAck | DUT(제어기)가 선행Subscribe Eventgroup Entry의 응답으로 전송한 Subscribe Eventgroup Acknowledge/Negative Acknowledge Entry 를 Tester(시험장비) 가 기다리는 최대 시간 |
| TimeoutStopSubscribe | DUT(제어기)가 전송한 Stop Subscribe Eventgroup Entry 를 Tester(시험장비) 가 기다리는 최대 시간 |
| TimeoutInitialEvents | DUT(제어기)가 전송한 초기 이벤트의 notification 을 Tester(시험장비) 가 기다리는 최대 시간 |
| TimeoutCloseTcpConnection | TCP 연결이 DUT에 의해 종료될 때까지 Tester(시험장비)가 대기해야하는 최대 시간 |
| TimingTolerance | 이벤트와 관련된 공차 시간.  대기 또는 청취 할 때 이 파라미터는 실제 대기 시간 또는 청취 시간과 함께 추가됩니다. 이것은 현재 DUT 응답이 두 번째 간격 인 경우에 사용됩니다. |
| TimingToleranceMillisec | 일부 이벤트와 관련된 공차 시간 (ms).  대기 또는 청취 할 때 이 파라미터는 실제 대기 시간 또는 청취 시간과 함께 추가됩니다. 이것은 현재 DUT 응답이 밀리 초 간격 인 경우에 사용됩니다. |
| ETS\_CyclicOfferDelay | Specifies the maximum duration the Tester shall wait for the next cyclic sent Offer Service Entry. |
| ETS\_ServiceID\_0 | Enhanced Testability Service의 Service ID |
| ETS\_ServiceID\_1 | A second Service ID of the Enhanced Testability Service. |
| ETS\_ClientID | Enhanced Testability Service에서 사용하는 기본 SOME/IP Client ID |
| ETS\_InstanceID\_0 | DUT(제어기)에서 실행중인 Enhanced Testability Service 인스턴스의 Instance ID |
| ETS\_InstanceID\_1 | The second Instance ID of the Enhanced Testability Service instance which is running on the DUT. |
| ETS\_InstanceID\_2 | The third Instance ID of the Enhanced Testability Service static configuration instance which is running on the DUT. |
| ETS\_MajorVersion\_0 | Enhanced Testability Service에서 사용하는 Major Version |
| ETS\_MajorVersion\_1 | A second Major Version of the used Enhanced Testability Service. |
| ETS\_MinorVersion\_0 | Enhanced Testability Service에서 사용하는 Minor Version |
| ETS\_MinorVersion\_1 | A second Minor Version of the used Enhanced Testability Service. |
| ETS\_Lifetime | ETS 관련 Offer Service또는 Fine Service가 기본값으로 s 단위 표시 |
| ETS\_SubscriptionLifetime | ETS 관련 Subscribe Eventgroup Entry 기본 값은 s 단위 로 유효한 것으로 간주 |
| ETS\_IPv4Address | Default configurated IP address of the ETS system. |
| ETS\_StaticIPv4MulticastAddress | Statically configured multicast IP address for related ETS test cases. |
| ETS\_IPv4MulticastAddress | Default configured multicast IP address of ETS system. |
| Tester\_IPv4Address | Default configurated IP address of the Tester system |
| ETS\_PortUDP\_0 | Default UDP Port number for related ETS test cases |
| ETS\_PortUDP\_1 | A second UDP Port number for related ETS test cases when having more than one Instance of a service. |
| ETS\_PortTCP\_0 | Default TCP Port number for related ETS test cases |
| ETS\_PortTCP\_1 | A second TCP Port number for related ETS test cases when having more than one Instance of a service and need a second TCP connection for it |
| ETS\_StaticMulticastPort | Statically configured multicast port for related ETS test cases. |
| TESTER\_PortUDP | Default configurated UDP Port of the Tester system |
| TESTER\_PortTCP | Default configurated TCP Port of the Tester system. |

## Enhanced Testability Service (ETS)

### 개요

모든 SOME/IP-ETS 시험을 실행하려면 정의된 인터페이스가 필요합니다. 일반적인 테스트 도구 및 테스트 사례를 허용하기 위해 이 인터페이스는 OPEN TC8 내에서 표준화되었으며 Enhanced Testability Service (ETS)라고 합니다.

Enhanced Testability Service 에서 다루는 프로토콜 부분은 다음과 같습니다.

* SOME/IP Stack - Service Discovery
* SOME/IP Stack - Serialization
* SOME/IP Stack - Remote Procedure Call
* SOME/IP Stack - Service Discovery
* SOME/IP Stack - Publish/Subscribe

OPEN TC8 에서 표준화 정의된 부분 외 HMC/KMC의 요구사양 기반 테스트 범위를 강화하기 위해 추가 방법이 정의되어 있습니다. (4.3.3)

Enhanced Testability Service는 다양한 시험 범주를 허용합니다. (예: DUT(테스트 대상장치)를 위한 테스트 시나리오의 구성요소로 사용) 여기는 positive test (유효한 메시지를 사용하는 테스트), negative test (오류 처리 테스트), load test, regression test가 포함됩니다.

SOME / IP 클라이언트 동작의 모든 측면을 다루려면 여러 SOME / IP 클라이언트가 필요합니다. 이러한 클라이언트는 테스트 시스템에서 서로 다른 시뮬레이션 된 ETS 인스턴스와 통신하여 올바른 동작을 보장합니다. 필요한 SOME / IP 클라이언트에 대한 자세한 설명은 (표 8 : SOME / IP Client Instance 목록)을 참조하십시오.

### Standardized Interfaces

이 절에는 OPEN Alliance TC8 에 정의 된 표준화 된 ETS 방법이 나와 있습니다.

#### 개요

표 4 methods 목록

| **Method** | **ID** | **Fire & Forget** |
| --- | --- | --- |
| checkByteOrder | 31 (0x1F) |  |
| clientServiceActivate | 47 (0x2F) | x |
| clientServiceDeactivate | 48 (0x30) | x |
| clientServiceSubscribeEventgroup | 50 (0x32) | x |
| echoCommonDatatypes | 35 (0x23) |  |
| echoENUM | 23 (0x17) |  |
| echoFLOAT64 | 18 (0x12) |  |
| echoInt64 | 52 (0x34) |  |
| echoINT8 | 14 (0x0E) |  |
| echoStaticUINT8Array | 54 (0x36) |  |
| echoUINT8 | 8 (0x08) |  |
| echoUINT8Array | 9 (0x09) |  |
| echoUINT8Array8BitLength | 62 (0x3E) |  |
| echoUINT8Array16BitLength | 63 (0x3F) |  |
| echoUINT8Array2Dim | 53 (0x35) |  |
| echoUINT8ArrayMinSize | 55 (0x37) |  |
| echoUINT8E2E | 11 (0x0B) |  |
| echoUINT8RELIABLE | 10 (0x0A) |  |
| echoUNION | 25 (0x19) |  |
| echoUTF16DYNAMIC | 22 (0x16) |  |
| echoUTF16FIXED | 20 (0x14) |  |
| echoUTF8DYNAMIC | 21 (0x15) |  |
| echoUTF8FIXED | 19 (0x13) |  |
| resetInterface | 1 (0x01) | x |
| suspendInterface | 2 (0x02) | x |
| triggerEventUINT8 | 3 (0x03) | x |
| triggerEventUINT8Array | 4 (0x04) | x |
| triggerEventUINT8E2E | 6 (0x06) | x |
| triggerEventUINT8Reliable | 5 (0x05) | x |
| triggerEventUINT8Multicast | 58 (0x3A) | x |
| clientServiceGetLastValueOfEventTCP | 59 (0x3B) |  |
| clientServiceGetLastValueOfEventUDPUnicast | 60 (0x3C) |  |
| clientServiceGetLastValueOfEventUDPMulticast | 61 (0x3D) |  |
| echoBitfields | 65 (0x41) |  |

표 5 Events 및 Fields 목록

| **Type** | **Name** | **ID** | **Eventgroups** | | |
| --- | --- | --- | --- | --- | --- |
| 0x0002 | 0x0005 | 0x0006 |
| Event | TestEventUINT8 | 0x8001 | x | x |  |
| Event | TestEventUINT8Array | 0x8002 | x | x |  |
| Event | TestEventUINT8E2E | 0x8004 | x | x |  |
| Event | TestEventUINT8Reliable | 0x8003 | x |  |  |
| Event | TestEventUINT8Multicast | 0x800B |  |  | x |
| Field | InterfaceVersion | 0x8005 (Notify) | x | x |  |
| 0x25 (Getter) | x | x |  |
| Field | TestFieldUINT8 | 0x8006 (Notify) | x | x |  |
| 0x26 (Getter) | x | x |  |
| 0x27 (Setter) | x | x |  |
| Field | TestFieldUINT8Array | 0x8007 (Notify) | x | x |  |
| 0x28 (Getter) | x | x |  |
| 0x29 (Setter) | x | x |  |
| Field | TestFieldUINT8Reliable | 0x8008 (Notify) | x |  |  |
| 0x2A (Getter) | x |  |  |
| 0x2B (Setter) | x |  |  |

#### Method 정의

##### checkByteOrder

|  |  |
| --- | --- |
| *Request* | checkByteOrder\_Req |
| *Response* | checkByteOrder\_Res |
| *Group* | SOME/IP Serialization |
| *ID* | 31 (0x1F) |
| *Description* | This method is used to test the correct handling of the byte order.  The input parameters include a uint8 and a big endian uint16, which shall be added and returned as output parameter in uint32 big endian. |
| *Method Result* | returnValue = sum of uint8\_value + uint16\_value |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| uint8\_value | Uint8 | Byte order request test parameter 1 |
| uint16\_value | Uint16 | Byte order request test parameter 2 |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| returnValue | Uint32 | Byte order result test parameter = sum  of uint8\_value + uint16\_value |

##### clinetServiceActivate

|  |  |
| --- | --- |
| *Request* | clinetServiceActivate |
| *Response* |  |
| *Group* | Client Interaction |
| *ID* | 47 (0x2F) |
| *Description* | The commands clientServiceActivate shall instruct the DUT to start its SOME/IP default client; thus, they control when the system shall start finding the Enhanced Testability Service with the special configurable Instance ID (for example 0x00f4). |
| *Method Result* | Message Type is REQ\_NO\_RETURN |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| delay | Uint8 | Delay for activating the client service |

##### clinetServiceDeactivate

|  |  |
| --- | --- |
| *Request* | clientServiceDeactivate\_Req |
| *Response* |  |
| *Group* | Client Interaction |
| *ID* | 48 (0x30) |
| *Description* | The command clientServiceDeactivate shall instruct the DUT to stop the currently selected and active SOME/IP client; thus, they control when the system shall start finding the Enhanced Testability Service with the special configurable Instance ID (for example 0x00F4). |
| *Method Result* | Message Type is REQ\_NO\_RETURN |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| delay | Uint8 | Delay for deactivating the client service |

##### clientServiceSubscribeEventgroup

|  |  |
| --- | --- |
| *Request* | clientServiceSubscribeEventgroup\_Req |
| *Response* |  |
| *Group* | Client Interaction |
| *ID* | 50 (0x32) |
| *Description* | The clientServiceSubscribeEventgroup shall trigger the subscription behavior in Service Discovery. All ETS related Eventgroups shall be subscribed.  Else the Subscription running so long as it the duration timed out. If the Duration of the Subscription timed out, the Stop subscription will be sent automatically. |
| *Method Result* | Message Type is REQ\_NO\_RETURN |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| delay | Uint32 | Delay for the eventgroup |
| duration | Uint32 | Duration for the eventgroup subscription |

##### echoCommonDatatypes

|  |  |
| --- | --- |
| *Request* | EchoCommonDatatypes\_Req |
| *Response* | EchoCommonDatatypes\_Res |
| *Group* | SOME/IP Serialization |
| *ID* | 35 (0x23) |
| *Description* | This method can be used to test the common data types. |
| *Method Result* | The input parameters shall be echoed back in reversed order.  bool\_value = float64\_returnValue  uint8\_value = float32\_returnValue  etc. |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| bool\_value | Boolean | Common datatypes Boolean request test parameter. Data width is 8 bit. |
| uint8\_value | Uint8 | Common datatypes Uint8 request test parameter |
| uint16\_value | Uint16 | Common datatypes Uint16 request test parameter |
| uint32\_value | Uint32 | Common datatypes Uint32 request test parameter |
| int8\_value | Int8 | Common datatypes Int8 request test parameter |
| int16\_value | Int16 | Common datatypes Int16 request test parameter |
| int32\_value | Int32 | Common datatypes Int32 request test parameter |
| float32\_value | Float32 | Common datatypes Float32 request test parameter |
| float64\_value | Float64 | Common datatypes Float64 request test parameter |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| float64\_returnValue | Float64 | = float64\_value |
| float32\_returnValue | Float32 | = float32\_value |
| int32\_returnValue | Int32 | = int32\_value |
| int16\_returnValue | Int16 | = int16\_value |
| int8\_returnValue | Int8 | = int8\_value |
| uint32\_returnValue | Uint32 | = uint32\_value |
| uint16\_returnValue | Uint16 | = uint16\_value |
| uint8\_returnValue | Unit8 | = uint8\_value |
| bool\_returnValue | Boolean | = bool\_value |

##### echoENUM

|  |  |
| --- | --- |
| *Request* | echoENUM\_Req |
| *Response* | echoENUM\_Res |
| *Group* | SOME/IP Serialization |
| *ID* | 23 (0x17) |
| *Description* | This method can be used to test enum data types. |
| *Method Result* | returnValue = requestValue |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| requestValue | Uint8 | Enum request test parameter |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| returnValue | Uint32 | Enum response test parameter  = requestValue |

##### echoFLOAT64

|  |  |
| --- | --- |
| *Request* | echoFLOAT64\_Req |
| *Response* | echoFLOAT64\_Req |
| *Group* | SOME/IP Serialization |
| *ID* | 18(0x12) |
| *Description* | This method can be used to test float64 data types. |
| *Method Result* | returnValue = requestValue |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| requestValue | Float64 | Float64 request test parameter |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| returnValue | Float64 | Foat64 response test parameter |

##### echoINT8

|  |  |
| --- | --- |
| *Request* | echoINT8\_Req |
| *Response* | echoINT8\_Res |
| *Group* | SOME/IP Serialization |
| *ID* | 14(0x0E) |
| *Description* | This method can be used to test init8 data types. |
| *Method Result* | returnValue = requestValue |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| requestValue | Int8 | Int8 request test parameter |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| returnValue | Int8 | = requestValue |

##### echoStaticUINT8Array

|  |  |
| --- | --- |
| *Request* | echoStaticUINT8Array\_Req |
| *Response* | echoStaticUINT8Array\_Res |
| *Group* | SOME/IP Serialization |
| *ID* | 54(0x36) |
| *Description* | This method can be used to test static uint8 array data types. |
| *Method Result* | Static array of 5 Uint8 elements (without array lenght field)  returnValue = requestValue |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| requestValue | Array | static uint8 array request test parameter |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| returnValue | Array | = requestValue |

##### echoUINT8

|  |  |
| --- | --- |
| *Request* | echoUINT8\_Req |
| *Response* | echoUINT8\_Req |
| *Group* | SOME/IP Serialization |
| *ID* | 8(0x08) |
| *Description* | This method can be used to test uint8 data types.  The method returns the UINT8 value. In this case it sent via UDP. |
| *Method Result* | returnValue = requestValue |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| requestValue | uint8 | uint8 request test parameter |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| returnValue | uint8 | = requestValue |

##### echoUINT8Array

|  |  |
| --- | --- |
| *Request* | echoUINT8Array\_Req |
| *Response* | echoUINT8Array\_Res |
| *Group* | SOME/IP Serialization |
| *ID* | 9(0x09) |
| *Description* | This method can be used to test uint8 array data types. In this case it  sent via UDP. |
| *Method Result* | Dynamic array of n Uint8 elements  returnDataLength = dataLength, returnData = data |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| dataLength | Uint32 | Length of uint8 array request test parameter |
| data | Array | uint8 array request test parameter |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| returnDataLength | Uint32 | Length of uint8 array repsonse test parameter |
| returnData | Array | uint8 array response test parameter |

##### echoUINT8Array8BitsLength

|  |  |
| --- | --- |
| *Request* | echoUINT8Array8BitLength\_Req |
| *Response* | echoUINT8Array8BitLength\_Res |
| *Group* | SOME/IP Serialization |
| *ID* | 62 (0x3E) |
| *Description* | This method can be used to test uint8 array 8bit length data types. |
| *Method Result* | Dynamic array of n Uint8 elements  returnDataLength = dataLength, returnData = data |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| dataLength | Uint8 | Length of uint8 array 8bit length request test parameter |
| data | Array | uint8 array 8bit length request test parameter |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| returnDataLength | Uint8 | Length of uint8 array 8bit length response test parameter |
| returnData | Array | uint8 array 8bit length response test parameter |

##### echoUINT8Array16BitsLength

|  |  |
| --- | --- |
| *Request* | echoUINT8Array16BitLength\_Req |
| *Response* | echoUINT8Array16BitLength\_Res |
| *Group* | SOME/IP Serialization |
| *ID* | 63 (0x3F) |
| *Description* | This method can be used to test uint8 array 16bit length data types. |
| *Method Result* | Dynamic array of n Uint8 elements  returnDataLength = dataLength, returnData = data |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| dataLength | Uint8 | Length of uint8 array 16bit length request test parameter |
| data | Array | uint8 array 16bit length request test parameter |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| returnDataLength | Uint16 | Length of uint8 array 16bit length response test parameter |
| returnData | Array | uint8 array 16bit length response test parameter |

##### echoUINT8Array2Dim

|  |  |
| --- | --- |
| *Request* | echoUINT8Array2Dim\_Req |
| *Response* | echoUINT8Array2Dim\_Res |
| *Group* | SOME/IP Serialization |
| *ID* | 53 (0x35) |
| *Description* | This method can be used to test uint8 array 2 dimensional data types. |
| *Method Result* | Array of n arrays of n Uint8 elements  returnDataLength = dataLength, returnData = data |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| dataLength | Uint32 | Length of uint8 array 2 dim request test parameter |
| data | Array of arrays | uint8 array of arrays 2 dim request test parameter |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| returnDataLength | Uint32 | Length of uint8 array 2 dim length response test parameter |
| returnData | Array of Uint8Arrays | uint8 array of arrays 2 dim response test parameter |

##### echoUINT8ArrayMinSize

|  |  |
| --- | --- |
| *Request* | echoUINT8ArrayMinSize\_Req |
| *Response* | echoUINT8ArrayMinSize\_Res |
| *Group* | SOME/IP Serialization |
| *ID* | 55 (0x37) |
| *Description* | This method can be used to test uint8 array min size data types. |
| *Method Result* | Dynamic array of n Uint8 elements with n = 3 .. 5  returnDataLength = dataLength, returnData = data |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| dataLength | Uint32 | Length of uint8 array min size request test parameter |
| data | Array | uint8 array min size request test parameter |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| returnDataLength | Uint32 | Length of uint8 array min size response test parameter |
| returnData | Array | uint8 array min size response test parameter |

##### echoUINT8E2E

|  |  |
| --- | --- |
| *Request* | echoUINT8E2E\_Req |
| *Response* | echoUINT8E2E\_Res |
| *Group* | SOME/IP Serialization |
| *ID* | 11 (0x0B) |
| *Description* | This method can be used to test uint8 array E2E data types.  This method uses End-to-End-protection |
| *Method Result* | returnValue = requestValue |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| length | Uint32 | uint8E2E request test parameter length |
| counter | Uint32 | uint8E2E request test parameter counter |
| dataID | Uint32 | uint8E2E request test parameter dataID |
| crc | Uint32 | uint8E2E request test parameter crc |
| value | Uint8 | uint8E2E request test parameter |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| returnLength | Uint32 | uint8E2E response test parameter length |
| returnCounter | Uint32 | uint8E2E response test parameter counter |
| returnDataID | Uint32 | uint8E2E response test parameter dataID |
| returnCRC | Uint32 | uint8E2E response test parameter crc |
| returnValue | Uint8 | uint8E2E response test parameter |

##### echoUINT8RELIABLE

|  |  |
| --- | --- |
| *Request* | echoUINT8RELIABLE\_ReqArg1 |
| *Response* | echoUINT8RELIABLE\_ResArg1 |
| *Group* | SOME/IP Serialization |
| *ID* | 10 (0x0A) |
| *Description* | This method can be used to test uint8 reliable data types.  The method uses TCP |
| *Method Result* | returnValue = requestValue |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| requestValue | Uint8 | uint8 reliable request test parameter |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| returnValue | Uint8 | uint8 reliable response test parameter |

##### echoUTF16DYNAMIC

|  |  |
| --- | --- |
| *Request* | echoUTF16DYNAMIC\_Req |
| *Response* | echoUTF16DYNAMIC\_Res |
| *Group* |  |
| *ID* | 22 (0x16) |
| *Description* | This method can be used to test UTF16 dynamic data types. |
| *Method Result* | returnDataLength = dataLength  returnData = data |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| dataLength | Uint32 | UTF16 dynamic request test parameter length |
| data | UTF16 string | UTF16 dynamic request test parameter |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| returnDataLength | Uint32 | UTF16 dynamic response test parameter length |
| returnData | UTF16 string | UTF16 dynamic response test parameter |

##### echoUTF16FIXED

|  |  |
| --- | --- |
| *Request* | echoUTF16FIXED\_Req |
| *Response* | echoUTF16FIXED\_Res |
| *Group* | SOME/IP Serialization |
| *ID* | 20 (0x14) |
| *Description* | This method can be used to test UTF16 fixed data types. |
| *Method Result* | UTF16Fixed = 64 bytes of string including BOM  returnData = data |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| data | UTF16Fixed | UTF16 fixed request test parameter |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| returnData | UTF16Fixed | UTF16 fixed response test parameter |

##### echoUTF8DYNAMIC

|  |  |
| --- | --- |
| *Request* | echoUTF8DYNAMIC\_Req |
| *Response* | echoUTF8DYNAMIC\_Res |
| *Group* | SOME/IP Serialization |
| *ID* | 21 (0x15) |
| *Description* | This method can be used to test UTF8 dynamic data types. |
| *Method Result* | returnDataLength = dataLength  returnData = data |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| dataLength | Uint32 | UTF8 dynamic request test parameter length |
| data | UTF8 string | UTF8 dynamic request test parameter |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| returnDataLength | Uint32 | UTF8 dynamic response test parameter length |
| returnData | UTF8 string | UTF8 dynamic response test parameter |

##### echoUTF8FIXED

|  |  |
| --- | --- |
| *Request* | echoUTF8FIXED\_Req |
| *Response* | echoUTF8FIXED\_Res |
| *Group* | SOME/IP Serialization |
| *ID* | 19 (0x13) |
| *Description* | This method can be used to test UTF8 fixed data types. |
| *Method Result* | UTF8Fixed = 64 bytes of string including BOM  returnData = data |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| data | UTF8Fixed | UTF8 fixed request test parameter |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| retrunData | UTF8Fixed | UTF8 fixed response test parameter |

##### resetInterface

|  |  |
| --- | --- |
| *Request* | resetInterface\_Req |
| *Response* |  |
| *Group* | General |
| *ID* | 1 (0x01) |
| *Description* | This method resets the interface to default values. |
| *Method Result* | Message Type is REQ\_NO\_RETURN without parameters |

##### suspendInterface

|  |  |
| --- | --- |
| *Request* | suspendInterface\_Req |
| *Response* |  |
| *Group* | General |
| *ID* | 2 (0x02) |
| *Description* | This method allows to suspend the Enhanced Testability Service; thus, forcing the DUT to stop offering the service and start to reoffer the service after a given time. This method use UDP. |
| *Method Result* | Message Type is REQ\_NO\_RETURN |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| delay | Uint32 | wait time in seconds before the service shall be stopped |
| duration | Uint32 | wait time in seconds during between stopping the service and reoffering it again. |

##### triggerEventUNIT8

|  |  |
| --- | --- |
| *Request* | triggerEventUINT8\_Req |
| *Response* |  |
| *Group* | Events |
| *ID* | 3 (0x03) |
| *Description* | After the method triggerEventUINT8 (delay, duration, debounceTime) was invoked, the service Testability waits <delay > seconds and triggers a periodical event testEventUINT8 (uINT8Value) with the passed duration and debounce time.  The transport protocol for the trigger method and the event is UDP. |
| *Method Result* | Message Type is REQ\_NO\_RETURN |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| delay | Uint32 | wait time in seconds before the event shall be triggered |
| duration | Uint32 | wait time in seconds for the duration the event shall be triggered |
| debounceTime | Uint32 | Debounce time in seconds |

##### triggerEventUNIT8Array

|  |  |
| --- | --- |
| *Request* | triggerEventUINT8Array\_Req |
| *Response* |  |
| *Group* | Events |
| *ID* | 4 (0x04) |
| *Description* | After the method triggerEventUINT8Array (delay, duration, debounceTime) was invoked, the service Testability waits <delay > seconds and triggers a periodical event testEventUINT8Array (UINT8ArrayValue) with the passed duration and debounce time.  The transport protocol for the trigger method and the event is UDP. |
| *Method Result* | Message Type is REQ\_NO\_RETURN |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| delay | Uint32 | wait time in seconds before the event shall be triggered |
| duration | Uint32 | wait time in seconds for the duration the event shall be triggered |
| debounceTime | Uint32 | Debounce time in seconds |

##### triggerEventUNIT8E2E

|  |  |
| --- | --- |
| *Request* | triggerEventUINT8E2E\_Req |
| *Response* |  |
| *Group* | Events |
| *ID* | 6 (0x06) |
| *Description* | After the method triggerEventUINT8E2E (delay, duration, debounceTime) was invoked, the service Testability waits <delay > seconds and triggers a periodical event testEventUINT8E2E (UINT8E2EValue) with the passed duration and debounce time.  E2E methods always use End-to-End protection. |
| *Method Result* | Message Type is REQ\_NO\_RETURN |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| delay | Uint32 | wait time in seconds before the event shall be triggered |
| duration | Uint32 | wait time in seconds for the duration the event shall be triggered |
| debounceTime | Uint32 | Debounce time in seconds |

##### triggerEventUNIT8Reliable

|  |  |
| --- | --- |
| *Request* | triggerEventUINT8Reliable\_Req |
| *Response* |  |
| *Group* | Events |
| *ID* | 5 (0x05) |
| *Description* | After the method triggerEventUINT8Reliable (delay, duration, debounceTime) was invoked, the service Testability waits <delay > seconds and triggers a periodical event testEventUINT8Reliable(UINT8Value) with the passed duration and debounce time.  The transport protocol of the event message is TCP. |
| *Method Result* | Message Type is REQ\_NO\_RETURN |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| delay | Uint32 | wait time in seconds before the event shall be triggered |
| duration | Uint32 | wait time in seconds for the duration the event shall be triggered |
| debounceTime | Uint32 | Debounce time in seconds |

##### triggerEventUNIT8Multicast

|  |  |
| --- | --- |
| *Request* | triggerEventUINT8Multicast\_Req |
| *Response* |  |
| *Group* | Events |
| *ID* | 58 (0x3A) |
| *Description* | After the method triggerEventUINT8Multicast (delay, duration, debounceTime) was invoked, the service Testability waits <delay > seconds and triggers a periodical event testEventUINT8 (UINT8Value) with the passed duration and debounce time. |
| *Method Result* | Message Type is REQ\_NO\_RETURN |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| delay | Uint32 | wait time in seconds before the event shall be triggered |
| duration | Uint32 | wait time in seconds for the duration the event shall be triggered |
| debounceTime | Uint32 | Debounce time in seconds |

##### clientServiceGetLastValueOfEventTCP

|  |  |
| --- | --- |
| *Request* | clientServiceGetLastValueOfEventTCP\_Req |
| *Response* | clientServiceGetLastValueOfEventTCP\_Res |
| *Group* | Client Interaction |
| *ID* | 59 (0x3B) |
| *Description* | Returns the last received value of TestEventUINT8Reliable, TestEventUINT8, and TestEventUINT8Multicast of clientServiceGetLastValueOfEventTCP |
| *Method Result* |  |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| value | Uint8 | Response value of the last received TestEvent method over TCP |

##### clientServiceGetLastValueOfEventUDPUnicast

|  |  |
| --- | --- |
| *Request* | clientServiceGetLastValueOfEventUDPUnicast\_Req |
| *Response* | clientServiceGetLastValueOfEventUDPUnicast\_Res |
| *Group* | Client Interaction |
| *ID* | 60 (0x3C) |
| *Description* | Returns the last received value of TestEventUINT8Reliable, TestEventUINT8, and TestEventUINT8Multicast of clientServiceGetLastValueOfUDPUnicast |
| *Method Result* |  |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| value | Uint8 | Response value of the last received TestEvent method over UDP unicast |

##### clientServiceGetLastValueOfEventUDPMulticast

|  |  |
| --- | --- |
| *Request* | clientServiceGetLastValueOfEventUDPMulticast\_Req\_Req |
| *Response* | clientServiceGetLastValueOfEventUDPMulticast\_Res |
| *Group* | Client Interaction |
| *ID* | 61 (0x3D) |
| *Description* | Returns the last received value of TestEventUINT8Reliable, TestEventUINT8, and TestEventUINT8Multicast of clientServiceGetLastValueOfUDPMulticast |
| *Method Result* |  |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| value | Uint8 | Response value of the last received TestEvent method over UDP multicast |

##### echoBitfields

|  |  |
| --- | --- |
| *Request* | echoBitfields\_Req |
| *Response* | echoBitfields\_Res |
| *Group* | SOME/IP Serialization |
| *ID* | 65 (0x41) |
| *Description* | This method can be used to test bitfield data types. |
| *Method Result* | returnValue1 = reversed bit order of requestValue1  returnValue2 = reversed bit order of requestValue2  returnValue3 = reversed bit order of requestValue3 |

|  |  |  |
| --- | --- | --- |
| *Request Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| requestValue1 | uint8 | 8bit bit field request test parameter |
| requestValue2 | uint16 | 16bit bit field request test parameter |
| requestValue3 | uint32 | 32bit bit field request test parameter |
| *Response Parameters* | | |
| *Name* | *Type {Range}* | *Description* |
| returnValue1 | uint8 | 8bit bit field response test parameter |
| returnValue2 | uint16 | 16bit bit field response test parameter |
| returnValue3 | uint32 | 32bit bit field response test parameter |

#### Event and Field 정의

##### TestEventUINT8

|  |  |
| --- | --- |
| *Type* | Event |
| *ID* | 0x8001 |
| *Data Type* | UINT8 |
| *Description* | This event can be used to test an UINT8 event with UDP binding.  A notification could be triggered by method triggerEventUINT8. |

##### TestEventUINT8Array

|  |  |
| --- | --- |
| *Type* | Event |
| *ID* | 0x8002 |
| *Data Type* | UINT8 |
| *Description* | This event can be used to test an UINT8 Array event with UDP binding.  A notification could be triggered by method triggerEventUINT8Array. |

##### TestEventUINT8E2E

|  |  |
| --- | --- |
| *Type* | Event |
| *ID* | 0x8004 |
| *Data Type* | UINT8 |
| *Description* | This event can be used to test an UINT8 event with UDP binding and E2E protection.  A notification could be triggered by method triggerEventUINT8E2E. |

##### TestEventUINT8Reliable

|  |  |
| --- | --- |
| *Type* | Event |
| *ID* | 0x8003 |
| *Data Type* | UINT8 |
| *Description* | This event can be used to test an UINT8 event with TCP binding.  A notification could be triggered by method triggerEventUINT8Reliable.. |

##### TestEventUINT8Multicast

|  |  |
| --- | --- |
| *Type* | Event |
| *ID* | 0x800B |
| *Data Type* | UINT8 |
| *Description* | This event can be used to test an UINT8 event with UDP Multicast binding.  A notification could be triggered by method triggerEventUINT8Multicast.  An appropriate IPv4 Multicast Option is required to be configured for this event on the DUT. |

##### InterfaceVersion

|  |  |
| --- | --- |
| *Type* | Field |
| *ID* | 0x8005 |
| *Data Type* | UINT8 |
| *Description* | Returns major version of the ETS. |

##### TestFieldUINT8

|  |  |
| --- | --- |
| *Type* | Field |
| *ID* | 0x8006 |
| *Data Type* | UINT8 |
| *Description* | Can be used to test an UINT8 field with UDP binding. |

##### TestFieldUINT8Array

|  |  |
| --- | --- |
| *Type* | Field |
| *ID* | 0x8007 |
| *Data Type* | UINT8 Array |
| *Description* | Can be used to test an UINT8 Array field with UDP binding. |

##### TestFieldUINT8Reliable

|  |  |
| --- | --- |
| *Type* | Field |
| *ID* | 0x8008 |
| *Data Type* | UINT8 |
| *Description* | Can be used to test an UINT8 field with TCP binding. |

#### Data Type 정의

##### TestArrayUNIT8Static

|  |  |
| --- | --- |
| *Type* | UINT8 array |
| *Length* | 5 elements |
| *Description* | This type defines an array of UINT8 values with a fixed length of 5 elements, which is used in context of testing the serialization/deserialization of data types. |

##### TestArrayUNIT8Fixed

|  |  |
| --- | --- |
| *Type* | UTF8 string |
| *Length* | 64 characters |
| *Description* | This type defines an UTF8 encoded string with a fixed length of 64 characters, which is used in context of testing the serialization/deserialization of data types. |

##### TestArrayUNIT16Fixed

|  |  |
| --- | --- |
| *Type* | UTF16 string |
| *Length* | 64 characters |
| *Description* | This type defines an UTF16 encoded string with a fixed length of 64 characters, which is used in context of testing the serialization/deserialization of data types. |

##### TestEnum

|  |  |
| --- | --- |
| *Type* | enum |
| *Description* | This type defines an enumeration of 5 members. |

|  |  |
| --- | --- |
| *Enum Members* | |
| *Name* | *Value* |
| member1 | 0x01 |
| member2 | 0x02 |
| member3 | 0x03 |
| member4 | 0x04 |
| member5 | 0xAA |

### Extended Interfaces

이 절에는 HMC/KMC 이더넷 SOME/IP 평가 사양의 시험 커버리지를 향상시키기 위해 정의된 비표준 ETS 방법이 나와 있습니다.

#### 개요

표 6 extended Methods 목록

| **Method** | **ID** | **Transport Protocol Binding** | **Fire & Forget** |
| --- | --- | --- | --- |
| shutdownECU | 1280 (0x500) | UDP | X |
| suspendInterfaceReliable | 1281 (0x501) | TCP | X |
| triggerCyclicEventUINT8 | 1282 (0x502) | UDP | X |
| changeNetworkMode | 1283 (0x503) | UDP | X |
| echoUINT8ArrayReliable | 1284 (0x504) | TCP |  |
| echoStructFixed | 1285 (0x505) | UDP |  |
| echoStructFixed8BitLength | 1286 (0x506) | UDP |  |
| echoStructFixed16BitLength | 1287 (0x507) | UDP |  |
| echoStructFixed32BitLength | 1288 (0x508) | UDP |  |
| echoStructDynamic | 1289 (0x509) | UDP |  |
| echoStructNested | 1290 (0x50A) | UDP |  |
| echoUTF8DYNAMIC8BitLength | 1296 (0x510) | UDP |  |
| echoUTF8DYNAMIC16BitLength | 1297 (0x511) | UDP |  |
| echoUTF16DYNAMIC8BitLength | 1298 (0x512) | UDP |  |
| echoUTF16DYNAMIC16BitLength | 1299 (0x513) | UDP |  |
| provideConfigurationOption | 1307 (0x51B) | UDP | X |
| provideSDEndpointOption | 1308 (0x51C) | UDP | X |
| setUINT8Array | 1292 (0x50C) | UDP | X |
| setUINT8ArrayReliable | 1293 (0x50D) | TCP | X |
| setEventUINT8Change | 1294 (0x50E) | UDP |  |
| setUTF8DYNAMICReliable | 1305 (0x519) | TCP | X |
| echoUINT8Conditional | 1304 (0x518) | UDP |  |
| triggerEventUINT8Implicit | 1311 (0x51F) | UDP | X |
| clientServiceSelectInstance | 1312 (0x520) | UDP | X |
| clientServiceGetLastError | 1313 (0x521) | UDP |  |
| clientServiceSubscribeEventgroupSingle | 1314 (0x522) | UDP | X |
| clientServiceCallEchoUINT8 | 1315 (0x523) | UDP | X |
| clientServiceCallEchoUINT8Reliable | 1331 (0x533) | TCP | X |
| clientServiceCallEchoUINT8Array | 1316 (0x524) | TCP | X |
| clientServiceCallEchoUTF8DYNAMIC | 1317 (0x525) | UDP | X |
| clientServiceCallFieldUTF8DynamicReliableSetter | 1332 (0x534) | TCP | X |
| clientServiceCallEchoBitfields | 1318 (0x526) | UDP | X |
| clientServiceCallSetUINT8Array | 1322 (0x52A) | TCP | X |
| clientServiceCallFieldUINT8Getter | 1328 (0x530) | UDP | X |
| clientServiceCallFieldUINT8Setter | 1329 (0x531) | UDP | X |
| clientServiceGetCANMessageInfo | 1344 (0x540) | UDP |  |
| clientServiceGetCANErrorCount | 1345 (0x541) | UDP |  |
| serviceActivate | 2001 (0x7D1) | UDP | X |
| serviceDeactivate | 2002 (0x7D2) | UDP | X |
| echoUTF16FIXED\_LE | 2003 (0x7D3) | UDP |  |
| echoUTF16DYNAMIC\_LE | 2004 (0x7D4) | UDP |  |
| echoUTF16DYNAMIC8BitLength\_LE | 2005 (0x7D5) | UDP |  |
| echoUTF16DYNAMIC16BitLength\_LE | 2006 (0x7D6) | UDP |  |
| suspendEthernetInterface | 2007 (0x7D7) | UDP | X |
| setSessionID | 2008 (0x7D8) | UDP | X |
| setRepetitionMaxCount | 2009 (0x7D9) | UDP | X |
| enableTP | 2010 (0x7DA) | UDP | X |

표 7 extended Events 및 Fields 목록

| **Type** | **Name** | **ID** | **Eventgroups** | | |
| --- | --- | --- | --- | --- | --- |
| 0x0002 | 0x0005 | 0x0006 |
| Event | TestEventUINT8Implicit | 0x8501 | x | x |  |
| Event | TestEventUINT8Cyclic | 0x8502 | x | x |  |
| Event | TestEventUINT8Change | 0x8503 | x | x |  |
| Field | TestFieldUTF8DynamicReliable | 0x850A (Notify) | x |  |  |
|  |  | 0x0560 (Getter) | x |  |  |
|  |  | 0x0561 (Setter) | x |  |  |

표 8 SOME/IP Client Instance 목록

| **Client Instance** | **Service** | **Instance** | **Major Version** | **Minor Version** | **Description** |
| --- | --- | --- | --- | --- | --- |
| Default instance | 0x2001 | 0x2001 | 0x01 | 0x00000000 | This instance will be started by default in case no other instance was explicitly selected prior to the activation. |
| Duplicated instance | 0x2001 | 0x2001 | 0x01 | 0x00000000 | This instance has the same properties as the default instance. It will be started whenever a second client instance with the same attributes as the default instance on the DUT is required. |
| Specific service version instance | 0x2001 | 0x2001 | 0x02 | 0x00000000 | This instance differs from the default instance only in its interface version. It will be started whenever a instance with a specific interface version is required, e.g. when two client services are required that are using different service instances. |
| Static configuration instance | 0x2001 | 0x2003 | 0x01 | 0x00000000 | This instance is by default using a statically configured service description, i.e. is could use services without SOME/IP-SD.  *Note: Statically defined services could be overwritten by SOME/IP-SD.* |
| CAN default instance | 0x2001 | 0x2001 | 0x01 | 0x00000000 | Dedicated instance used for CAN Message Transport related test cases. It expects to receive a periodic CAN Encapsulated SOME/IP message.  The instance does not use any service or Service Discovery to find services. |
| CAN E2E Profile 5 instance | 0x2001 | 0x2001 | 0x01 | 0x00000000 | Dedicated instance used for CAN Message Transport related test cases. It expects to receive a periodic CAN Encapsulated SOME/IP message which is encoded accordingly to AUTOSAR 4.2.2 E2E Profile 5.  The instance does not use any service or Service Discovery to find services. |
| CAN E2E Profile 11 instance | 0x2001 | 0x2001 | 0x01 | 0x00000000 | Dedicated instance used for CAN Message Transport related test cases. It expects to receive a periodic CAN Encapsulated SOME/IP message which is encoded accordingly to AUTOSAR 4.3.1 E2E Profile 11.  The instance does not use any service or Service Discovery to find services. |

표 9 SOME/IP Service Instance 목록

| **Service Instance** | **Service** | **Instance** | **Major Version** | **Minor Version** | **Description** |
| --- | --- | --- | --- | --- | --- |
| Default service | 0x1001 | 0x1001 | 0x01 | 0x00000000 | This instance will be started by default. |
| Alternative service | 0x1002 | 0x1001 | 0x01 | 0x00000000 | This instance differs from the default instance only in its alternative service. It will be started whenever an instance with an alternative service is required, e.g. when two services are required that are using different service instances. |
| Alternative instance service | 0x1001 | 0x1002 | 0x01 | 0x00000000 | This instance differs from the default instance only in its alternative instance. It will be started whenever an instance with an alternative instance is required, e.g. when two services are required that are using different service instances. |
| Alternative major version service | 0x1001 | 0x1001 | 0x02 | 0x00000000 | This instance differs from the default instance only in its alternative major version. It will be started whenever an instance with an alternative major version is required, e.g. when two services are required that are using different service instances. |
| Alternative minor version service | 0x1001 | 0x1001 | 0x01 | 0x00000001 | This instance differs from the default instance only in its alternative minor version. It will be started whenever an instance with an alternative minor version is required, e.g. when two services are required that are using different service instances. |

\* Hexadecimal numbers are used as arbitrary values in this table.

#### Method 정의

##### ETS\_StaticIPv4Address

|  |  |
| --- | --- |
| *Type* | IPv4 Address (UINT32) |
| *Description* | This constant is used for testing static SOME/IP configuration and provides an arbitrary value chosen by the ETS developer.  From Server side: The static default IP Address, where the Server sends Events without a subscribe to the client.  From Client side: The static default IP Address, where the Client listen whether Events from the Server arrive. |
|  | |

##### ETS\_StaticPortUDP

|  |  |
| --- | --- |
| *Type* | UINT16 |
| *Description* | This constant is used for testing static SOME/IP configuration and provides an arbitrary value chosen by the ETS developer.  From Server side: The static default UDP Port, where the Server sends Events without a subscribe to the client.  From Client side: The static default UDP Port, where the Client listen whether Events from the Server arrive. |
|  | |

#### Method Definition

##### shutdownECU

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | shutdownECU\_Req | | |
| *Response* | - | | |
| *Group* | General | | |
| *ID* | 1280 (0x500) | | |
| *Description* | This method can be used to force a shutdown/reboot of the hosting ECU. | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| delay | | UINT32 | wait time in seconds before the shutdown shall be triggered |

##### suspendInterfaceReliable

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | suspendInterfaceRelliable\_Req | | |
| *Response* |  | | |
| *Group* | General | | |
| *ID* | 1281 (0x501) | | |
| *Description* | This method allows to suspend the Enhanced Testability Service; thus, forcing the DUT to stop offering the service and start to reoffer the service after a given time. The method uses TCP | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type {Range}* | *Description* |
| delay | | Uint32 | wait time in seconds before the service shall be stopped |
| duration | | Uint32 | wait time in seconds during between stopping the service and reoffering it again. |

##### triggerCyclicEventUINT8

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | triggerCyclicEventUINT8\_Req | | |
| *Response* |  | | |
| *Group* | Events | | |
| *ID* | 1282 (0x502) | | |
| *Description* | After the method triggerCyclicEventUINT8 (delay, duration, debounceTime) was invoked, the service Testability waits <delay > seconds and triggers a Cyclic event testEventUINT8Cyclic (uINT8Value) with the passed duration, debounce time and sent cyclically every 1000 [ms].  The transport protocol for the trigger method and the event is UDP. | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type {Range}* | *Description* |
| delay | | Uint32 | wait time in seconds before the event shall be triggered |
| duration | | Uint32 | wait time in seconds for the duration the event shall be triggered |
| debounceTime | | Uint32 | Debounce time in seconds |

##### changeNetworkMode

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | | changeNetworkMode\_Req | |
| *Response* | |  | |
| *Group* | | General | |
| *ID* | | 1283 (0x503) | |
| *Description* | | This method instructs the ETS to temporary force a specific ECU network mode for testing. After a given duration, the network mode shall be automatically reset to its previous value. | |
| *Method Result* | | Message Type is REQ\_NO\_RETURN | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | *Type {Range}* | | *Description* |
| mode | Uint8 | | Indicates the requested network mode. Must be one of the following values:   * 0x01: inactive * 0x02: error state |
| duration | Uint32 | | Time in milliseconds [ms] the requested network mode shall be applied to DUT. |

##### echoUINT8ArrayReliable

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Request* | echoUINT8ArrayReliable\_Req | | | | |
| *Response* | echoUINT8ArrayReliable\_Res | | | | |
| *Group* | SOME/IP Serialization | | | | |
| *ID* | 1284 (0x504) | | | | |
| *Description* | This method can be used to test uint8 array data types.  The method uses TCP | | | | |
| *Method Result* | Dynamic array of n Uint8 elements  returnDataLength = dataLength, returnData = data | | | | |
|  | | | | | |
| *Request Parameters* | | | | | |
| *Name* | | | *Type {Range}* | | *Description* |
| dataLength | | | Uint32 | | Length of uint8 array request test parameter |
| data | | | Array | | uint8 array request test parameter |
| *Response Parameters* | | | | | |
| *Name* | | *Type {Range}* | | *Description* | |
| returnDataLength | | Uint32 | | Length of uint8 array response reliable test parameter | |
| returnData | | Array | | uint8 array response reliable test parameter | |

##### echoStructFixed

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | echoStructFixed\_Req | | |
| *Response* | echoStructFixed\_Res | | |
| *Group* | SOME/IP Serialization | | |
| *ID* | 1285 (0x505) | | |
| *Description* | This method can be used to test serialization/deserialization of simple struct data types where all of the members consist of fixed length data types only. | | |
| *Method Result* | Value of type TestStructFixed (see chapter 4.4.3.5.1) which is equal to value given as input argument for this method (returnValue = requestValue). | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| requestValue | | struct | Test parameter of ETS related type TestStructFixed. |
| *Response Parameters* | | | |
| *Name* | | *Type* | *Description* |
| returnValue | | struct | Test parameter of ETS related type TestStructFixed. |

##### echoStructFixed8BitLength

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | echoStructFixed8BitLength\_Req | | |
| *Response* | echoStructFixed8BitLength\_Res | | |
| *Group* | SOME/IP Serialization | | |
| *ID* | 1286 (0x506) | | |
| *Description* | This method can be used to test serialization/deserialization of simple struct data types where all of the members consist of 8 bit length data types only. | | |
| *Method Result* | Value of type TestStructFixed (see chapter 4.4.3.5.1) which is equal to value given as input argument for this method (returnValue = requestValue). | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| requestValue | | struct | Test parameter of ETS related type TestStructFixed |
| *Response Parameters* | | | |
| *Name* | | *Type* | *Description* |
| returnValue | | struct | Test parameter of ETS related type TestStructFixed. |

##### echoStructFixed16BitLength

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | echoStructFixed16BitLength\_Req | | |
| *Response* | echoStructFixed16BitLength\_Res | | |
| *Group* | SOME/IP Serialization | | |
| *ID* | 1287 (0x507) | | |
| *Description* | This method can be used to test serialization/deserialization of simple struct data types where all of the members consist of 16 bit length data types only | | |
| *Method Result* | Value of type TestStructFixed (see chapter 4.4.3.5.1) which is equal to value given as input argument for this method (returnValue = requestValue). | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| requestValue | | struct | Test parameter of ETS related type TestStructFixed |
| *Response Parameters* | | | |
| *Name* | | *Type* | *Description* |
| returnValue | | struct | Test parameter of ETS related type TestStructFixed. |

##### echoStructFixed32BitLength

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | echoStructFixed32BitLength\_Req | | |
| *Response* | echoStructFixed32BitLength\_Res | | |
| *Group* | SOME/IP Serialization | | |
| *ID* | 1288 (0x508) | | |
| *Description* | This method can be used to test serialization/deserialization of simple struct data types where all of the members consist of 32 bit length data types only | | |
| *Method Result* | Value of type TestStructFixed (see chapter 4.4.3.5.1) which is equal to value given as input argument for this method (returnValue = requestValue). | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| requestValue | | struct | Test parameter of ETS related type TestStructFixed |
| *Response Parameters* | | | |
| *Name* | | *Type* | *Description* |
| returnValue | | struct | Test parameter of ETS related type TestStructFixed. |

##### echoStructDynamic

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | echoStructDynamic\_Req | | |
| *Response* | echoStructDynamic\_Res | | |
| *Group* | SOME/IP Serialization | | |
| *ID* | 1289 (0x509) | | |
| *Description* | This method can be used to test serialization/deserialization of struct data types where at least one member is a dynamic length data type. | | |
| *Method Result* | Value of type TestStructDynamic (see chapter 4.4.3.5.24.4.3.5.1) which is equal to value given as input argument for this method (returnValue = requestValue). | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| requestValue | | struct | Test parameter of ETS related type TestStructDynamic |
| *Response Parameters* | | | |
| *Name* | | *Type* | *Description* |
| returnValue | | struct | Test parameter of ETS related type TestStructDynamic. |

##### echoStructNested

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | echoStructNested\_Req | | |
| *Response* | echoStructNested\_Res | | |
| *Group* | SOME/IP Serialization | | |
| *ID* | 1290 (0x50A) | | |
| *Description* | This method can be used to test serialization/deserialization of struct data types where at least one member is another struct. | | |
| *Method Result* | Value of type TestStructNested (see chapter 4.4.3.5.34.4.3.5.1) which is equal to value given as input argument for this method (returnValue = requestValue). | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| requestValue | | struct | Test parameter of ETS related type TestStructNested |
| *Response Parameters* | | | |
| *Name* | | *Type* | *Description* |
| returnValue | | struct | Test parameter of ETS related type TestStructNested |

##### echoUTF8DYNAMIC8BitLength

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | echoUTF8DYNAMIC8BitLength\_Req | | |
| *Response* | echoUTF8DYNAMIC8BitLength\_Res | | |
| *Group* | SOME/IP Serialization | | |
| *ID* | 1296 (0x510) | | |
| *Description* | This method can be used to test serialization/deserialization of UTF8 (8 bit length) data types. | | |
| *Method Result* | Value of type UTF8 string (8bit) which is equal to value given as input argument for this method (returnValue = requestValue). | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| dataLength | | UINT8 | length of UTF8 string request test parameter |
| requestValue | | UTF8 string | UTF8 string (8 bit) dynamic request test parameter |
| *Response Parameters* | | | |
| *Name* | | *Type* | *Description* |
| dataLength | | UINT8 | length of UTF8 string response test parameter |
| returnValue | | UTF8 string | UTF8 string (8 bit) dynamic response test parameter |

##### echoUTF8DYNAMIC16BitLength

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | echoUTF8DYNAMIC16BitLength\_Req | | |
| *Response* | echoUTF8DYNAMIC16BitLength\_Res | | |
| *Group* | SOME/IP Serialization | | |
| *ID* | 1297 (0x511) | | |
| *Description* | This method can be used to test serialization/deserialization of UTF8 (16 bit length) data types. | | |
| *Method Result* | Value of type UTF8 string (16bit) which is equal to value given as input argument for this method (returnValue = requestValue). | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| dataLength | | UINT8 | length of UTF8 string request test parameter |
| requestValue | | UTF8 string | UTF8 string (16 bit) dynamic request test parameter |
| *Response Parameters* | | | |
| *Name* | | *Type* | *Description* |
| dataLength | | UINT8 | length of UTF8 string response test parameter |
| returnValue | | UTF8 string | UTF8 string (16 bit) dynamic response test parameter |

##### echoUTF16DYNAMIC8BitLength

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | echoUTF16DYNAMIC8BitLength\_Req | | |
| *Response* | echoUTF16DYNAMIC8BitLength\_Res | | |
| *Group* | SOME/IP Serialization | | |
| *ID* | 1298 (0x512) | | |
| *Description* | This method can be used to test serialization/deserialization of UTF16 (8 bit length) data types.  The method is able to process UTF16BE as well as UTF16LE strings. | | |
| *Method Result* | Value of type UTF16 string (8bit) which is equal to value given as input argument for this method (returnValue = requestValue). | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| dataLength | | UINT8 | length of UTF16 string request test parameter |
| requestValue | | UTF16 string | UTF16 (8 bit) dynamic request test parameter |
| *Response Parameters* | | | |  | *Response Parameters* |
| *Name* | | *Type* | *Description* |
| dataLength | | UINT8 | length of UTF16 string response test parameter |
| returnValue | | UTF16 string | UTF16 (8 bit) dynamic response test parameter |

##### echoUTF16DYNAMIC16BitLength

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | echoUTF16DYNAMIC16BitLength\_Req | | |
| *Response* | echoUTF16DYNAMIC16BitLength\_Res | | |
| *Group* | SOME/IP Serialization | | |
| *ID* | 1299 (0x513) | | |
| *Description* | This method can be used to test serialization/deserialization of UTF16 (16 bit length) data types.  The method is able to process UTF16BE as well as UTF16LE strings. | | |
| *Method Result* | Value of type UTF16 string (16bit) which is equal to value given as input argument for this method (returnValue = requestValue). | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| dataLength | | UINT16 | length of UTF16 string request test parameter |
| requestValue | | UTF16 string | UTF16 (16 bit) dynamic request test parameter |
| *Response Parameters* | | | |
| *Name* | | *Type* | *Description* |
| dataLength | | UINT16 | length of UTF16 string response test parameter |
| returnValue | | UTF16 string | UTF16 (16 bit) dynamic response test parameter |

##### provideConfigurationOption

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | provideConfigurationOption\_Req | | |
| *Response* | - | | |
| *Group* | SOME/IP-SD Options | | |
| *ID* | 1307 (0x51B) | | |
| *Description* | This method shall be used to configure the ETS to append the respective configuration options to each sent SOME/IP-SD entry.  When period given by 1st parameter of this method is elapsed, the ETS will stop appending the option. | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| duration | | UINT32 | Time in seconds for the duration the options shall be appended. |
| keyLength | | UINT32 | Length of the key related to the key/value pair that shall be configured. |
| key | | UTF8 string | Key of the related key/value pair that shall be configured.  *Note: String value of key must not be empty.* |
| valueLength | | UINT32 | Length of the value related to the key/value pair that shall be configured. |
| value | | UTF8 string | Value of the related key/value pair that shall be configured. |
| key2Length | | UINT32 | Length of the key related to the key-only item that shall be configured. |
| key2 | | UTF8 string | Key of the related key-only item that shall be configured.  *Note: If the string value is empty it will not be considered.* |

##### provideSDEndpointOption

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | provideSDEndpointOption\_Req | | |
| *Response* | - | | |
| *Group* | SOME/IP-SD Options | | |
| *ID* | 1308 (0x51C) | | |
| *Description* | This method shall be used to configure the ETS to append the respective SD endpoint options to each sent SOME/IP-SD message.  When period given by 1st parameter of this method is elapsed, the ETS will stop appending the option. | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| duration | | UINT32 | Time in seconds for the duration the options shall be appended. |
| ip\_address | | UINT32 | The respective IPv4 address to be set in the SD endpoint options. The address consists of 32 bits where a group of 8 bits represents a single part in the well-known dot-decimal notation: The most significant 8 bits represent the first part, the following 8 bits the second part and so on. |
| portNumber | | UINT16 | The respective UDP port number to be set in the SD endpoint options |

##### setUINT8Array

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | setUINT8Array\_Req | | |
| *Response* | - | | |
| *Group* | SOME/IP Serialization | | |
| *ID* | 1292 (0x50C) | | |
| *Description* | This method can be used to set a specific value to a UINT8 array field TestFieldUINT8Array. | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| dataLength | | UINT32 | Length of the array |
| data | | UINT8 array | Specific data to be set. |

##### setUINT8ArrayReliable

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | setUINT8ArrayReliable\_Req | | |
| *Response* | - | | |
| *Group* | SOME/IP Serialization | | |
| *ID* | 1293 (0x50D) | | |
| *Description* | This method can be used to set a specific value to a UINT8 array field TestFieldUINT8Array over TCP. | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| dataLength | | UINT32 | Length of the array |
| data | | UINT8 array | Specific data to be set. |

##### setEventUINT8Change

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | setEventUINT8Change\_Req | | |
| *Response* | - | | |
| *Group* | SOME/IP Serialization | | |
| *ID* | 1294 (0x50E) | | |
| *Description* | This method can be used to trigger event notification for TestEventUINT8Change. | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| value | | UINT8 | UINT8 value |

##### setUTF8DYNAMICReliable

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | setUTF8DYNAMICReliable\_Req | | |
| *Response* | - | | |
| *Group* | SOME/IP Serialization | | |
| *ID* | 1305 (0x519) | | |
| *Description* | This method can be used to set a global UTF8 variable over TCP. In this case the variable is a String which is filled by the method (setUTF8DYNAMICReliable) and so just need to call the global variable to use this String value. | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| dataLength | | UINT32 | Length of the UTF8 string |
| data | | UTF8 string | Specific UTF8 string to be set. |

##### echoUINT8Conditional

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | echoUINT8Conditional\_Req | | |
| *Response* | echoUINT8Conditional \_Res | | |
| *Group* | General | | |
| *ID* | 1304 (0x518) | | |
| *Description* | This method could be used to trigger error messages for testing purposes. In general, it simply returns the given request value when it lies in the allowed range of 0x00 0x07F.  In case of the value is out of range, a SOME/IP error message with error code 0x20 is returned. In addition the error message provides an UTF8 string as error specific data which corresponds to the current value of TestFieldUTF8DynamicReliable (see 4.4.3.4.4). | | |
| *Method Result* | *returnvalue = requestvalue.* | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| requestValue | | uint8 | uint8 request test parameter. Should be in range 0x00 – 0x7F. |
| *Response Parameters* | | | |
| *Name* | | *Type* | *Description* |
| returnValue | | Uint8 | = requestValue |
| errorMessage | | UTF8 string | UTF8 string as error specific data which corresponds to the current value of TestFieldUTF8DynamicReliable. |

##### triggerEventUINT8Implicit

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | triggerEventUINT8Implicit\_Req | | |
| *Response* | - | | |
| *Group* | Events | | |
| *ID* | 1311 (0x51F) | | |
| *Description* | After the method triggerEventUINT8implicit (delay, duration, debounceTime) was invoked, the service Testability waits <delay> seconds and triggers a periodical event testEventUINT8 (uINT8Value) with the passed duration and debounce time.  The transport protocol for the trigger method and the event is UDP. The event will be only sent to statically configured clients, i.e. clients which were not subscribing via SOME/IP-SD. | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| delay | | UINT32 | wait time in seconds before the service shall be triggered |
| duration | | UINT32 | time in seconds for the duration the event shall be triggered |
| debounceTime | | UINT32 | Debounce time in seconds |

##### clientServiceSelectInstance

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | clientServiceSelectInstance\_Req | | |
| *Response* | - | | |
| *Group* | Client Interaction | | |
| *ID* | 1312 (0x520) | | |
| *Description* | This method shall be used to select the respective instance of the client service which shall receive and process the subsequent client service calls. | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| instanceKey | | UINT8 | Instance that shall be selected.  0x00: Default instance  0x01: Duplicated instance  0x02: specific Service version instance  0x03: Static configuration instance  0x10: CAN default instance  0x11: CAN E2E Profile 5 instance  0x12: CAN E2E Profile 11 instance |

##### clientServiceGetLastError

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | clientServiceGetLastError\_Req | | |
| *Response* | clientServiceGetLastError\_Res | | |
| *Group* | Client Interaction | | |
| *ID* | 1313 (0x521) | | |
| *Description* | Returns the last error occurred in the ETS. | | |
| *Method Result* |  | | |
|  | | | |
| *Response Parameters* | | | |
| *Name* | | *Type* | *Description* |
| errorCode | | UINT8 | Error code of the last occurred error. |

##### clientServiceSubscribeEventgroupSingle

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | | clientServiceSubscribeEventgroupSingle\_Req | |
| *Response* | | - | |
| *Group* | | Client Interaction | |
| *ID* | | 1314 (0x522) | |
| *Description* | | The clientServiceSubscribeEventgroupSingle shall trigger the subscription behavior in Service Discovery for a certain single event group.  Else the Subscription running so long as it the duration timed out. If the Duration of the Subscription timed out, the Stop subscription will be sent automatically. | |
| *Method Result* | | Message Type is REQ\_NO\_RETURN | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | *Type* | | *Description* |
| delay | UINT32 | | time in seconds before the service shall be subscribed |
| duration | UINT32 | | time in seconds for the duration the event shall be subscribed. When Duration is set to **DO\_NOT\_EXPIRE** the TTL is set to the maximum value (0xFFFFFF). |
| eventgroupID | UINT16 | | Event group ID of the service that shall be subscribed |

##### clientServiceCallEchoUINT8

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | clientServiceCallEchoUINT8\_Req | | |
| *Response* | - | | |
| *Group* | Client Interaction | | |
| *ID* | 1315 (0x523) | | |
| *Description* | This method shall trigger the client service of the ETS to call the method echoUINT8 (see 4.4.2.2.10). | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| delay | | UINT32 | time in seconds before the value shall be echoed |
| value | | UINT8 | UINT8 value to be echoed |

##### clientServiceCallEchoUINT8Reliable

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | clientServiceCallEchoUINT8Reliable\_Req | | |
| *Response* | - | | |
| *Group* | Client Interaction | | |
| *ID* | 1331 (0x533) | | |
| *Description* | This method shall trigger the client service of the ETS to call the method echoUINT8 with TCP Binding (see 4.4.2.2.10). | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| delay | | UINT32 | time in seconds before the value shall be echoed |
| value | | UINT8 | UINT8 value to be echoed |

##### clientServiceCallEchoUINT8Array

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | clientServiceCallEchoUINT8Array\_Req | | |
| *Response* | - | | |
| *Group* | Client Interaction | | |
| *ID* | 1316 (0x524) | | |
| *Description* | This method shall trigger the client service of the ETS to call the method echoUINT8Array (see 4.4.2.2.11). | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| delay | | UINT32 | time in seconds before the value shall be echoed |
| dataLength | | UINT32 | Length of the UINT8 array value |
| data | | UTF8 array | UINT8 array value to be echoed |

##### clientServiceCallEchoUTF8DYNAMIC

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | clientServiceCallEchoUTF8DYNAMIC\_Req | | |
| *Response* | - | | |
| *Group* | Client Interaction | | |
| *ID* | 1317 (0x525) | | |
| *Description* | This method shall trigger the client service of the ETS to call the method echoUTF8DYNAMIC (see 4.4.3.3.12). | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| delay | | UINT32 | time in seconds before the value shall be echoed |
| dataLength | | UINT32 | Length of the UTF8 string value |
| data | | UTF8 string | UTF8 string value to be echoed |

##### clientServiceCallFieldUTF8DynamicReliableSetter

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | clientServiceCallFieldUTF8DynamicReliableSetter\_Req | | |
| *Response* |  | | |
| *Group* | Client Interaction | | |
| *ID* | 1329 (0x534) | | |
| *Description* | This method shall trigger the client service of the ETS to call the setter of TestFieldUTF8DynamicReliable.This method use TCP. | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| delay | | UINT32 | time in seconds before the setter shall be called. |
| dataLength | | UINT32 | Length of the UTF8 string value |
| data | | UTF8 string | UTF8 string value to be set |

##### clientServiceCallEchoBitfields

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | clientServiceCallEchoBitfields\_Req | | |
| *Response* | - | | |
| *Group* | Client Interaction | | |
| *ID* | 1332 (0x526) | | |
| *Description* | This method shall trigger the client service of the ETS to call the method echoBitfields (see 4.4.2.2.32). | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| delay | | UINT32 | time in seconds before the value shall be echoed |
| echoBitfields\_ReqArg1 | | UINT8 | 8bit bit field request test parameter |
| echoBitfields\_ReqArg2 | | UINT16 | 16bit bit field request test parameter |
| echoBitfields\_ReqArg3 | | UINT32 | 32bit bit field request test parameter |

##### clientServiceCallSetUINT8Array

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | clientServiceCallSetUINT8Array\_Req | | |
| *Response* | - | | |
| *Group* | Client Interaction | | |
| *ID* | 1322 (0x52A) | | |
| *Description* | This method shall trigger the client service of the ETS to call the method setUINT8Array (see 4.4.3.3.18). | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| delay | | UINT32 | time in seconds before the value shall be set |
| dataLength | | UINT32 | Length of the UINT8 array value |
| data | | UINT8 array | UINT8 array value to be set |

##### clientServiceCallFieldUINT8Getter

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | clientServiceCallFieldUINT8Getter\_Req | | |
| *Response* | - | | |
| *Group* | Client Interaction | | |
| *ID* | 1328 (0x530) | | |
| *Description* | This method shall trigger the client service of the ETS to call the getter of TestFieldUINT8. | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| delay | | UINT32 | time in seconds before the value shall be set |

##### clientServiceCallFieldUINT8Setter

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | clientServiceCallFieldUINT8Setter\_Req | | |
| *Response* |  | | |
| *Group* | Client Interaction | | |
| *ID* | 1329 (0x531) | | |
| *Description* | This method shall trigger the client service of the ETS to call the setter of TestFieldUINT8. | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| delay | | UINT32 | time in seconds before the setter shall be called. |
| value | | UINT8 | UINT8 value to be set |

##### clientServiceGetCANMessageInfo

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | clientServiceGetCANMessageInfo\_Req | | |
| *Response* | clientServiceGetCANMessageInfo\_Res | | |
| *Group* | Client Interaction | | |
| *ID* | 1344 (0x540) | | |
| *Description* | This method is related to CAN Message Transport and returns information about received CAN messages of a particular CAN Message ID from the ETS client instance.  In detail, the signal values of the latest received message and the number, how many times they were updated in total, are provided. Furthermore, a value is returned that indicates whether a timeout of the CAN Message has been detected since the last call of this method.  Method is supported by ETS client instances only which are dedicated to CAN Message Transport related test cases. | | |
| *Method Result* | Please refer to the list of Response Parameters | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| messageID | | UINT32 | Base identifier (11 bit) or extended identifier (29 bit) of a CAN message that shall be addressed. |
| *Response Parameters* | | | |
| *Name* | | *Type* | *Description* |
| isTimeoutOccurred | | Boolean  (8-bit data width) | True when at least one timeout of the addressed CAN message is detected by client instance since the last call of clientServiceGetCANMessageInfo, otherwise false |
| signalValue1 | | UINT8 | Latest value of the UINT8 signal which was received by the addressed CAN message.  When no value has been received so far, parameter is set to the signal’s default value defined in the CAN database. |
| signalUpdates1 | | UINT8 | Number of times that value of the UINT8 signal was updated. |
| signalValue2 | | UINT32 | Latest value of UINT32 signal which was received by the addressed CAN message.  When no value has been received so far, parameter is set to the signal’s default value defined in the CAN database. |
| signalUpdates2 | | UINT8 | Number of times that value of the UINT32 signal was updated. |

##### clientServiceGetCANErrorCount

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | clientServiceGetCANErrorCount\_Req | | |
| *Response* | clientServiceGetCANErrorCount\_Res | | |
| *Group* | Client Interaction | | |
| *ID* | 1345 (0x541) | | |
| *Description* | This method is related to CAN Message Transport and returns two values, the DUT’s global error count (Global\_ERR\_COUNT) and the individual error count (ERR\_COUNT) of a particular CAN Message ID.  Method is supported by ETS client instances only which are dedicated to CAN Message Transport related test cases. | | |
| *Method Result* | globalErrorCount = Global\_ERR\_COUNT,  errorCount = ERR\_COUNT of addressed CAN message | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| messageID | | UINT32 | Base identifier (11 bit) or extended identifier (29 bit) of a CAN message that shall be addressed. |
| *Response Parameters* | | | |
| *Name* | | *Type* | *Description* |
| globalErrorCount | | UINT8 | Value of Global\_ERR\_COUNT. |
| errorCount | | UINT8 | Value of ERR\_COUNT for the CAN message which is addressed by request parameter ‘messageID’. |

##### serviceActivate

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | serviceActivate\_Req | | |
| *Response* | - | | |
| *Group* | General | | |
| *ID* | 2001 (0x7D1) | | |
| *Description* | This method shall activate the selected addtional ETS service instance. | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| delay | | UINT8 | Time in seconds before activate. |
| instanceKey | | UINT8 | Service instance key to activate.  InstanceKey identifies the service that has different feature from other service.  ( 0x01: alternative service,  0x02: alternativce instance,  0x03: alternative major version,  0x04: alternative minor version ) |

##### serviceDeactivate

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | serviceDeactivate\_Req | | |
| *Response* | - | | |
| *Group* | General | | |
| *ID* | 2002 (0x7D2) | | |
| *Description* | This method shall deactivate the selected addtional ETS service instance. | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| delay | | UINT8 | Time in seconds before deactivate. |
| instanceKey | | UINT8 | Service instance key to deactivate.  InstanceKey identifies the service that has different feature from other service.  ( 0x01: alternative service,  0x02: alternativce instance,  0x03: alternative major version,  0x04: alternative minor version ) |

##### echoUTF16FIXED\_LE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Request* | echoUTF16FIXED\_LE\_Req | | | | |
| *Response* | echoUTF16FIXED\_LE\_Res | | | | |
| *Group* | SOME/IP Serialization | | | | |
| *ID* | 2003 (0x7D3) | | | | |
| *Description* | This method can be used to test UTF16 Little Endian fixed data types. The method is able to process UTF16LE\_Fixed strings. | | | | |
| *Method Result* | returnData = data UTF16LE\_Fixed = 64 bytes of string including BOM | | | | |
|  | | | | | |
| *Request Parameters* | | | | | |
| *Name* | | | *Type {Range}* | | *Description* |
| data | | | UTF16LEFixed | | UTF16LEFixed request test parameter |
| *Response Parameters* | | | | | |
| *Name* | | *Type {Range}* | | *Description* | |
| returnData | | UTF16LEFixed | | UTF16LEFixed response test parameter | |

##### echoUTF16DYNAMIC\_LE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Request* | echoUTF16DYNAMIC\_LE\_Req | | | |
| *Response* | echoUTF16DYNAMIC\_LE\_Res | | | |
| *Group* | SOME/IP Serialization | | | |
| *ID* | 2004 (0x7D4) | | | |
| *Description* | This method can be used to test UTF16 Little Endian dynamic data types. The method is able to process UTF16LE dynamic strings. | | | |
| *Method Result* | returnData = data  returnDataLength = dataLength | | | |
|  | | | | |
| *Request Parameters* | | | | |
| *Name* | | *Type {Range}* | | *Description* |
| dataLength | | Uint32 | | UTF16LE dynamic request test parameter length |
| data | | UTF16LE string | | UTF16LE dynamic request test parameter |
| *Response Parameters* | | | | |
| *Name* | | *Type {Range}* | *Description* | |
| returnDataLength | | Uint32 | UTF16LE dynamic response test parameter length | |
| data | | UTF16LE string | UTF16LE dynamic response test parameter | |

##### echoUTF16DYNAMIC8BitLength\_LE

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | echoUTF16DYNAMIC8BitLength\_LE\_Req | | |
| *Response* | echoUTF16DYNAMIC8BitLength\_LE\_Res | | |
| *Group* | SOME/IP Serialization | | |
| *ID* | 2005 (0x7D5) | | |
| *Description* | This method can be used to test serialization/deserialization of UTF16 Little Endian with 8bit data length. The method is able to process UTF16LE (8bit). | | |
| *Method Result* | argument for this method (returnValue = requestValue) | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| dataLength | | UINT8 | Length of UTF16LE string request test parameter |
| requestValue | | UTF16LE string | UTF16LE (8bit) dynamic request test parameter. |
| *Response Parameters* | | | |
| *Name* | | *Type* | *Description* |
| dataLength | | UINT8 | Length of UTF16LE string response test parameter. |
| returnValue | | UTF16LE string | UTF16LE (8bit) dynamic response test parameter. |

##### echoUTF16DYNAMIC16BitLength\_LE

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | echoUTF16DYNAMIC16BitLength\_LE\_Req | | |
| *Response* | echoUTF16DYNAMIC16BitLength\_LE\_Res | | |
| *Group* | SOME/IP Serialization | | |
| *ID* | 2006 (0x7D6) | | |
| *Description* | This method can be used to test serialization/deserialization of UTF16 Little Endian (16bit length). The method is able to process UTF16LE. | | |
| *Method Result* | argument for this method (returnValue = requestValue) | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| dataLength | | UINT16 | Length of UTF16LE string request test parameter |
| requestValue | | UTF16LE string | UTF16LE (16bit) dynamic request test parameter. |
| *Response Parameters* | | | |
| *Name* | | *Type* | *Description* |
| dataLength | | UINT16 | Length of UTF16LE string response test parameter. |
| returnValue | | UTF16LE string | UTF16LE (16bit) dynamic response test parameter. |

##### suspendEthernetInterface

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | suspendEthernetInterface\_Req | | |
| *Response* | - | | |
| *Group* | General | | |
| *ID* | 2007 (0x7D7) | | |
| *Description* | This method suspends Ethernet interface for certain period of time.  This method can be implemented by calling system command.  (For example: ifconfig up/down) | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| delay | | UINT32 | Wait time in seconds before Ethernet interface gets suspended. |
| durationMSec | | UINT32 | Wait time in milliseconds while Ethernet interface’s state being down and up again. |

##### setSessionID

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | setSessionID\_Req | | |
| *Response* | - | | |
| *Group* | General | | |
| *ID* | 2008 (0x7D8) | | |
| *Description* | This method shall set the session ID of all ETS services.  (If called, session ID in SOME/IP SD message and SOME/IP request message shall be set to sessionID of request parameter) | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| sessionID | | UINT16 | Session ID to be set. |

##### setRepetitionMaxCount

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | setRepetitionMaxCount\_Req | | |
| *Response* | - | | |
| *Group* | General | | |
| *ID* | 2009 (0x7D9) | | |
| *Description* | This method shall set DUT’s repetition max count to max\_count of request parameter. | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| max\_count | | UINT8 | Value to be set for DUT’s repetition max count. |

##### enableTP

|  |  |  |  |
| --- | --- | --- | --- |
| *Request* | enableTP\_Req | | |
| *Response* | - | | |
| *Group* | General | | |
| *ID* | 2010 (0x7DA) | | |
| *Description* | This method enables TP of specified method for all ETS services and clients. | | |
| *Method Result* | Message Type is REQ\_NO\_RETURN | | |
|  | | | |
| *Request Parameters* | | | |
| *Name* | | *Type* | *Description* |
| methodID | | UINT16 | TP method ID to be applied. |
| enable | | BOOL | TP enable flag |

#### Event and Field Definition

##### TestEventUINT8Implicit

|  |  |
| --- | --- |
| *Type* | Event |
| *ID* | 0x8501 |
| *Data Type* | UINT8 |
| *Description* | This event can be used to test an UINT8 event with UDP binding.  A notification could be triggered by method triggerEventUINT8Implicit. |

##### TestEventUINT8Cyclic

|  |  |
| --- | --- |
| *Type* | Event |
| *ID* | 0x8502 |
| *Data Type* | UINT8 |
| *Description* | This event can be used to test an UINT8 event with UDP binding.  Notifications are sent cyclically every 1000 [ms]. |

##### TestEventUINT8Change

|  |  |
| --- | --- |
| *Type* | Event |
| *ID* | 0x8503 |
| *Data Type* | UINT8 |
| *Description* | This event can be used to test an UINT8 event with UDP binding.  A notification could be triggered by method setEventUINT8Change. |

##### TestFieldUTF8DynamicReliable

|  |  |
| --- | --- |
| *Type* | Field |
| *ID* | 0x850A |
| *Data Type* | UTF8 string of dynamic length |
| *Description* | Can be used to test an UTF8 string field with TCP binding.  Further more, the value of the field is updated internally every time the ETS receives a configuration option with the latest configuration string. So proper processing of received configuration options could be checked.  In addition, the value of the field is also used as error specific data for SOME/IP error messages in context of echoUINT8Conditional (see 4.4.3.3.22). Also, it is updated internally with the payload of the latest received SOME/IP error message with error code 0x20. |

#### Data Type Definition

##### TestStructFixed

|  |  |  |  |
| --- | --- | --- | --- |
| *Type* | struct | | |
| *Description* | This type defines a struct whose member consists of fixed length data types only. | | |
|  | | | |
| *Struct Members* | | | |
| *Name* | | *Type* | *Description* |
| member1 | | UINT8 | Member of fixed length data type UINT8. |
| member2 | | Float32 | Member of fixed length data type Float32. |
| member3 | | Boolean | Member of fixed length data type Boolean. |
| member4 | | UINT32 | Member of fixed length data type UINT32. |

##### TestStructDynamic

|  |  |  |  |
| --- | --- | --- | --- |
| *Type* | struct | | |
| *Description* | This type defines a struct data type where member2 and member3 are dynamic length data types. | | |
|  | | | |
| *Struct Members* | | | |
| *Name* | | *Type* | *Description* |
| member1 | | UINT8 | Member of fixed length data type UINT8. |
| member2 | | UINT8 array | Member of dynamic length data type UINT8 array. |
| member3 | | UTF8 string | Member of dynamic length data type UTF8 string. |

##### TestStructNested

|  |  |  |  |
| --- | --- | --- | --- |
| *Type* | struct | | |
| *Description* | This type defines a struct data type where member2 is another struct. | | |
|  | | | |
| *Struct Members* | | | |
| *Name* | | *Type* | *Description* |
| member1 | | UINT8 | Member of fixed length data type UINT8. |
| member2 | | struct | Member of struct data type TestStructFixed. |
| member3 | | boolean | Member of fixed length data type boolean. |

### CAN Database

The following table provides the required CAN message used for testing CAN Message Transport.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Message ID** | **DLC** | **Transmission Type** | **Interval** | **Start Offset Delay** |
| BaseFrameOnEvent | 0x050 | 5 | On Event | - | - |
| BaseFrameOnEventMulticast | 0x051 | 5 | On Event | - | - |
| BaseFrameOnEventFiltered | 0x052 | 5 | On Event | - | - |
| BaseFrameOnEventFiltered2 | 0x053 | 5 | On Event | - | - |
| BaseFramePeriodicOnEvent | 0x054 | 5 | Periodic and On Event | 200 ms | 1000 ms |
| BaseFramePeriodic | 0x055 | 5 | Periodic | 200 ms | 1600 ms |
| BaseFramePeriodicE2EProfile5 | 0x05A | 5 | Periodic | 200 ms | 500 ms |
| BaseFramePeriodicE2EProfile11 | 0x05B | 5 | Periodic | 200 ms | 400 ms |
| BaseFrameOnEventE2EProfile11 | 0x05C | 5 | On Event | - | - |
| ExtendedFrameOnEvent | 0x0000 0060 | 5 | On Event | - | - |
| ExtendedFramePeriodic | 0x0000 0061 | 5 | Periodic | 200 ms | 1200 ms |

Table 7: CAN messages for testing message transport

#### BaseFrameOnEvent

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Message Type* | | CAN base frame | | | |
| *Message ID* | | 0x050 | | | |
| *Transmission Type* | | On Event | | | |
| *Transmission Interval* | | - | | | |
| *Start Offset Delay* | | - | | | |
| *Data Length* | | 5 bytes | | | |
|  | | | | | |
| *Payload Data* | | | | | |
| *Name* | *Type* | *Default Value* | *Trigger Signal* | *Duplicated Value Check* | *ETS is receiver* |
| signalUINT8 | UINT8 | 0x00 | yes | disabled | yes |
| signalUINT32 | UINT32 | 0x0000 0000 | no | disabled | yes |

#### BaseFrameOnEventMulticast

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Message Type* | | CAN base frame | | | |
| *Message ID* | | 0x051 | | | |
| *Transmission Type* | | On Event | | | |
| *Transmission Interval* | | - | | | |
| *Start Offset Delay* | | - | | | |
| *Data Length* | | 5 bytes | | | |
|  | | | | | |
| *Payload Data* | | | | | |
| *Name* | *Type* | *Default Value* | *Trigger Signal* | *Duplicated Value Check* | *ETS is receiver* |
| signalUINT8 | UINT8 | 0x00 | yes | disabled | yes |
| signalUINT32 | UINT32 | 0x0000 0000 | no | disabled | yes |

#### BaseFrameOnEventFiltered

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Message Type* | | CAN base frame | | | |
| *Message ID* | | 0x052 | | | |
| *Transmission Type* | | On Event | | | |
| *Transmission Interval* | | - | | | |
| *Start Offset Delay* | | - | | | |
| *Data Length* | | 5 bytes | | | |
|  | | | | | |
| *Payload Data* | | | | | |
| *Name* | *Type* | *Default Value* | *Trigger Signal* | *Duplicated Value Check* | *ETS is receiver* |
| signalUINT8 | UINT8 | 0x00 | yes | disabled | no |
| signalUINT32 | UINT32 | 0x0000 0000 | no | disabled | no |

#### BaseFrameOnEventFiltered2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Message Type* | | CAN base frame | | | |
| *Message ID* | | 0x053 | | | |
| *Transmission Type* | | On Event | | | |
| *Transmission Interval* | | - | | | |
| *Start Offset Delay* | | - | | | |
| *Data Length* | | 5 bytes | | | |
|  | | | | | |
| *Payload Data* | | | | | |
| *Name* | *Type* | *Default Value* | *Trigger Signal* | *Duplicated Value Check* | *ETS is receiver* |
| signalUINT8 | UINT8 | 0x00 | yes | disabled | yes |
| signalUINT32 | UINT32 | 0x0000 0000 | no | disabled | no |

#### BaseFramePeriodicOnEvent

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Message Type* | | CAN base frame | | | |
| *Message ID* | | 0x054 | | | |
| *Transmission Type* | | Periodic and On Event | | | |
| *Transmission Interval* | | 200 milliseconds [ms] | | | |
| *Start Offset Delay* | | 1000 milliseconds [ms] | | | |
| *Data Length* | | 5 bytes | | | |
|  | | | | | |
| *Payload Data* | | | | | |
| *Name* | *Type* | *Default Value* | *Trigger Signal* | *Duplicated Value Check* | *ETS is receiver* |
| signalUINT8 | UINT8 | 0x00 | yes | disabled | yes |
| signalUINT32 | UINT32 | 0x0000 0000 | no | disabled | yes |

#### BaseFramePeriodic

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Message Type* | | CAN base frame | | | |
| *Message ID* | | 0x055 | | | |
| *Transmission Type* | | Periodic | | | |
| *Transmission Interval* | | 200 milliseconds [ms] | | | |
| *Start Offset Delay* | | 1600 milliseconds [ms] | | | |
| *Data Length* | | 5 bytes | | | |
|  | | | | | |
| *Payload Data* | | | | | |
| *Name* | *Type* | *Default Value* | *Trigger Signal* | *Duplicated Value Check* | *ETS is receiver* |
| signalUINT8 | UINT8 | 0x00 | no | disabled | yes |
| signalUINT32 | UINT32 | 0x0000 0000 | no | disabled | yes |

#### BaseFramePeriodicE2EProfile5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Message Type* | | CAN base frame | | | |
| *Message ID* | | 0x05A | | | |
| *Transmission Type* | | Periodic | | | |
| *Transmission Interval* | | 200 milliseconds [ms] | | | |
| *Start Offset Delay* | | 500 milliseconds [ms] | | | |
| *Data Length* | | 5 bytes | | | |
|  | | | | | |
| *Payload Data* | | | | | |
| *Name* | *Type* | *Default Value* | *Trigger Signal* | *Duplicated Value Check* | *ETS is receiver* |
| signalUINT8 | UINT8 | 0x00 | no | disabled | yes |
| signalUINT32 | UINT32 | 0x0000 0000 | no | disabled | yes |

#### BaseFramePeriodicE2EProfile11

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Message Type* | | CAN base frame | | | |
| *Message ID* | | 0x05B | | | |
| *Transmission Type* | | Periodic | | | |
| *Transmission Interval* | | 200 milliseconds [ms] | | | |
| *Start Offset Delay* | | 400 milliseconds [ms] | | | |
| *Data Length* | | 5 bytes | | | |
|  | | | | | |
| *Payload Data* | | | | | |
| *Name* | *Type* | *Default Value* | *Trigger Signal* | *Duplicated Value Check* | *ETS is receiver* |
| signalUINT8 | UINT8 | 0x00 | no | disabled | yes |
| signalUINT32 | UINT32 | 0x0000 0000 | no | disabled | yes |

#### BaseFrameOnEventE2EProfile11

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Message Type* | | CAN base frame | | | |
| *Message ID* | | 0x05C | | | |
| *Transmission Type* | | Periodic | | | |
| *Transmission Interval* | | - | | | |
| *Start Offset Delay* | | - | | | |
| *Data Length* | | 5 bytes | | | |
|  | | | | | |
| *Payload Data* | | | | | |
| *Name* | *Type* | *Default Value* | *Trigger Signal* | *Duplicated Value Check* | *ETS is receiver* |
| signalUINT8 | UINT8 | 0x00 | yes | disabled | yes |
| signalUINT32 | UINT32 | 0x0000 0000 | no | disabled | yes |

#### ExtendedFrameOnEvent

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Message Type* | | CAN extended frame | | | |
| *Message ID* | | 0x0000 0060 | | | |
| *Transmission Type* | | On Event | | | |
| *Transmission Interval* | | - | | | |
| *Start Offset Delay* | | - | | | |
| *Data Length* | | 5 bytes | | | |
|  | | | | | |
| *Payload Data* | | | | | |
| *Name* | *Type* | *Default Value* | *Trigger Signal* | *Duplicated Value Check* | *ETS is receiver* |
| signalUINT8 | UINT8 | 0x00 | yes | enabled | yes |
| signalUINT32 | UINT32 | 0x0000 0000 | no | enabled | yes |

#### ExtendedFramePeriodic

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Message Type* | | CAN extended frame | | | |
| *Message ID* | | 0x0000 0061 | | | |
| *Transmission Type* | | Periodic | | | |
| *Transmission Interval* | | 200 milliseconds [ms] | | | |
| *Start Offset Delay* | | 1200 milliseconds [ms] | | | |
| *Data Length* | | 5 bytes | | | |
|  | | | | | |
| *Payload Data* | | | | | |
| *Name* | *Type* | *Default Value* | *Trigger Signal* | *Duplicated Value Check* | *ETS is receiver* |
| signalUINT8 | UINT8 | 0x00 | no | disabled | yes |
| signalUINT32 | UINT32 | 0x0000 0000 | no | disabled | yes |

# 시험 항목

## SOME/IP 기본동작

### Message Header

#### [SOMEIPGEN\_HEADER\_01] Header contains Service ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a SOME/IP message header contains a 16 bit field which provides the unique identifier of the addressed service. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends SOME/IP message. 3. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP message is received within **TimeoutStartupSD** and contains:    1. Service ID is equal to 0xFFFF (reserved identifier for Service Discovery). |
| Reference | DS\_SP\_0002, DS\_SP\_0004, DS\_SP\_0006, DS\_SD\_0005, DS\_SD\_0006 |
| Notes | The test case expects to check the first message a SOME/IP server will send out, which should be actual a SOME/IP-SD message. Therefore appropriate values of a valid SOME/IP-SD message header are assumed for verification. |

#### [SOMEIPGEN\_HEADER\_02] Header contains Method ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a SOME/IP message header contains a 16 bit field which provides the identifier of the addressed method/event. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service.. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends SOME/IP message. 3. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP message is received within **TimeoutStartupSD** and contains: 2. Service ID is equal to 0xFFFF (reserved identifier for Service Discovery). 3. Method ID is equal to 0x8100 when parsed with byte order Big Endian. |
| Reference | DS\_SP\_0002, DS\_SP\_0004, DS\_SP\_0006, DS\_SP\_0079, DS\_SD\_0005, DS\_SD\_0007 |
| Notes | The test case expects to check the first message a SOME/IP server will send out, which should be actual a SOME/IP-SD message. Therefore appropriate values of a valid SOME/IP-SD message header are assumed for verification. |

#### [SOMEIPGEN\_HEADER\_03] Header contains Client ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a SOME/IP message header contains a 16 bit field which provides the identifier of the addressed client instance. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends SOME/IP message. 3. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP message is received within **TimeoutStartupSD** and contains: 2. Service ID is equal to 0xFFFF (reserved identifier for Service Discovery). 3. Client ID is equal to 0x0000. |
| Reference | DS\_SP\_0010, DS\_SP\_0013, DS\_SD\_0005, DS\_SD\_0009 |
| Notes | The test case expects to check the first message a SOME/IP server will send out, which should be actual a SOME/IP-SD message. Therefore appropriate values of a valid SOME/IP-SD message header are assumed for verification. |

#### [SOMEIPGEN\_HEADER\_04] Header contains Session ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a SOME/IP message header contains a 16 bit field which provides the Session ID. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends SOME/IP message. 3. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP message is received within **TimeoutStartupSD** and contains: 2. Service ID is equal to 0xFFFF (reserved identifier for Service Discovery). 3. Session ID is equal to or greater than 0x0001 when parsed with byte order Big Endian. |
| Reference | DS\_SP\_0010, DS\_SP\_0013, DS\_SP\_0079, DS\_SD\_0005, DS\_SD\_0010 |
| Notes | The test case expects to check the first message a SOME/IP server will send out, which should be actual a SOME/IP-SD message. Therefore appropriate values of a valid SOME/IP-SD message header are assumed for verification. |

#### [SOMEIPGEN\_HEADER\_05] Header contains Protocol Version

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a SOME/IP message header contains an 8 bit field which provides the used SOME/IP Protocol Version. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends SOME/IP message. 3. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP message is received within **TimeoutStartupSD** and contains: 2. Service ID is equal to 0xFFFF (reserved identifier for Service Discovery). 3. SOME/IP Protocol Version is equal to 0x01. |
| Reference | DS\_SP\_0022, DS\_SD\_0005, DS\_SD\_0015 |
| Notes | The test case expects to check the first message a SOME/IP server will send out, which should be actual a SOME/IP-SD message. Therefore appropriate values of a valid SOME/IP-SD message header are assumed for verification. |

#### [SOMEIPGEN\_HEADER\_06] Header contains Interface Version

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a SOME/IP message header contains an 8 bit field which provides the interface version number of the addressed service. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends SOME/IP message. 3. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP message is received within **TimeoutStartupSD** and contains: 2. Service ID is equal to 0xFFFF (reserved identifier for Service Discovery). 3. Interface Version is equal to 0x01. |
| Reference | DS\_SP\_0023, DS\_SD\_0005, DS\_SD\_0016 |
| Notes | The test case expects to check the first message a SOME/IP server will send out, which should be actual a SOME/IP-SD message. Therefore appropriate values of a valid SOME/IP-SD message header are assumed for verification. |

#### [SOMEIPGEN\_HEADER\_07] Header contains Message Type

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a SOME/IP message header contains an 8 bit field which provides the type of the message. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends SOME/IP message. 3. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP message is received within **TimeoutStartupSD** and contains: 2. Service ID is equal to 0xFFFF (reserved identifier for Service Discovery). 3. Message Type is equal to 0x02 (NOTIFICATION). |
| Reference | DS\_SP\_0024, DS\_SD\_0005, DS\_SD\_0017 |
| Notes | The test case expects to check the first message a SOME/IP server will send out, which should be actual a SOME/IP-SD message. Therefore appropriate values of a valid SOME/IP-SD message header are assumed for verification. |

#### [SOMEIPGEN\_HEADER\_08] Header contains Return Code

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a SOME/IP message header contains an 8 bit field which provides a return code to signal errors. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends SOME/IP message. 3. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP message is received within **TimeoutStartupSD** and contains: 2. Service ID is equal to 0xFFFF (reserved identifier for Service Discovery). 3. Return Code is equal to 0x00 (E\_OK). |
| Reference | DS\_SP\_0029, DS\_SD\_0005, DS\_SD\_0018 |
| Notes | The test case expects to check the first message a SOME/IP server will send out, which should be actual a SOME/IP-SD message. Therefore appropriate values of a valid SOME/IP-SD message header are assumed for verification. |

### Serialization of Parameters and Data Structures

#### [SOMEIPGEN\_DATA\_01] Parameters shall be carried in payload field

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Parameters shall be carried in payload field. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUINT8** with the following parameters:    * **value** is set to **UINT8\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains: 2. Service ID is equal to **ETS\_ServiceID\_0**. 3. Method ID is equal to 0x08 (**echoUINT8**) 4. Return Code is equal to 0x00 (E\_OK). 5. Payload provides an UINT8 value equal to **UINT8\_VALUE**. |
| Reference | DS\_SP\_0034 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_02] Length field covers payload

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a SOME/IP message header contains a 32 bit field which provides the length of the message in number of bytes. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUINT8** with the following parameters:    * **value** is set to **UINT8\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains: 2. Service ID is equal to **ETS\_ServiceID\_0**. 3. Method ID is equal to 0x08 (**echoUINT8**) 4. Return Code is equal to 0x00 (E\_OK). 5. Length is equal to 0x00000009. |
| Reference | DS\_SP\_0008 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_03] E2E header is placed after Return Code

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that an E2E header is placed after Return Code field of the SOME/IP message header. The following predefined profiles for E2E Protection could be applied:   * AUTOSAR 4.2.2 E2E Profile 5 {CRC 16(16bits ), AliveCounter(8bits)} * AUTOSAR 4.3.1 E2E Profile 11 {CRC 8(8bits), AliveCounter(4bits)} |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **E2E\_LENGTH:** Length of the E2E the E2E protected data  **E2E\_COUNTER:** Value of the E2E counter  **E2E\_DATA\_ID:** Unique data id of the data element  **E2E\_CRC:** Checksum of the E2E protected data  **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUINT8E2E** with the following parameters:    * **length** is set to **E2E\_LENGTH**.    * **counter** is set to **E2E\_COUNTER**.    * **dataID** is set to **E2E\_DATA\_ID**.    * **crc** is set to **E2E\_CRC**.    * **value** is set to **UINT8\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains: 2. Service ID is equal to ETS\_ServiceID\_0. 3. Method ID is equal to 0x0B (**echoUINT8E2E**) 4. Return Code is equal to 0x00 (E\_OK). 5. E2E header starts after at least 16 bytes from beginning of the message. |
| Reference | DS\_SP\_0001 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_04] Basic 32bit Datatypes

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of basic datatype parameters and their order (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UINT16\_VALUE**: 16-bit unsigned integer used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test.  **INT8\_VALUE**: 8-bit signed integer used as arbitrary payload for the test.  **INT16\_VALUE**: 16-bit signed integer used as arbitrary payload for the test.  **INT32\_VALUE**: 32-bit signed integer used as arbitrary payload for the test.  **FLOAT32\_VALUE**: 32-bit floating point number used as arbitrary payload for the test.  **FLOAT64\_VALUE**: 64-bit floating point number used as arbitrary payload for the test.  **BOOL\_VALUE**: Boolean value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoCommonDatatypes** with the following parameters:    * **uint8\_value** is set to **UINT8\_VALUE**.    * **uint16\_value** is set to **UINT16\_VALUE**.    * **uint32\_value** is set to **UINT32\_VALUE**.    * **int8\_value** is set to **INT8\_VALUE**.    * **int16\_value** is set to **INT16\_VALUE**.    * **int32\_value** is set to **INT32\_VALUE**.    * **float32\_value** is set to **FLOAT32\_VALUE**.    * **float64\_value** is set to **FLOAT64\_VALUE**.    * **bool\_value** is set to **BOOL\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains: 2. Service ID is equal to **ETS\_ServiceID\_0**. 3. Method ID is equal to 0x23 (**echoCommonDatatypes**) 4. Return Code is equal to 0x00 (E\_OK). 5. Payload provides an UINT8 value equal to **UINT8\_VALUE**. 6. Payload provides an UINT16 value equal to **UINT16\_VALUE**. 7. Payload provides an UINT32 value equal to **UINT32\_VALUE**. 8. Payload provides an INT8 value equal to **INT8\_VALUE**. 9. Payload provides an INT16 value equal to **INT16\_VALUE**. 10. Payload provides an INT32 value equal to **INT32\_VALUE**. 11. Payload provides a FLOAT32 value equal to **FLOAT32\_VALUE**. 12. Payload provides a FLOAT64 value equal to **FLOAT64\_VALUE**. 13. Payload provides a BOOL value equal the lowest bit of **BOOL\_VALUE.** |
| Reference | DS\_SP\_0040 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_05] Ignore all other bits of Boolean

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Ignore all undefined bits of Boolean datatype. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UINT16\_VALUE**: 16-bit unsigned integer used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test.  **INT8\_VALUE**: 8-bit signed integer used as arbitrary payload for the test.  **INT16\_VALUE**: 16-bit signed integer used as arbitrary payload for the test.  **INT32\_VALUE**: 32-bit signed integer used as arbitrary payload for the test.  **FLOAT32\_VALUE**: 32-bit floating point number used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoCommonDatatypes** with the following parameters:    * **uint8\_value** is set to **UINT8\_VALUE**.    * **uint16\_value** is set to **UINT16\_VALUE**.    * **uint32\_value** is set to **UINT32\_VALUE**.    * **int8\_value** is set to **INT8\_VALUE**.    * **int16\_value** is set to **INT16\_VALUE**.    * **int32\_value** is set to **INT32\_VALUE**.    * **float32\_value** is set to **FLOAT32\_VALUE**.    * **float64\_value** is set to **FLOAT64\_VALUE**.    * **bool\_value** is set to 0xFE. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains: 2. Service ID is equal to **ETS\_ServiceID\_0**. 3. Method ID is equal to 0x23 (**echoCommonDatatypes**) 4. Return Code is equal to 0x00 (E\_OK). 5. Payload provides an UINT8 value equal to **UINT8\_VALUE**. 6. Payload provides an UINT16 value equal to **UINT16\_VALUE**. 7. Payload provides an UINT32 value equal to **UINT32\_VALUE**. 8. Payload provides an INT8 value equal to **INT8\_VALUE**. 9. Payload provides an INT16 value equal to **INT16\_VALUE**. 10. Payload provides an INT32 value equal to **INT32\_VALUE**. 11. Payload provides a FLOAT32 value equal to **FLOAT32\_VALUE**. 12. Payload provides a FLOAT64 value equal to **FLOAT64\_VALUE**. 13. Payload provides a BOOL value equal to 0x00. |
| Reference | DS\_SP\_0041 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_06] Bitfield

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of bitfield parameters and their order (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Bitfield of 8 bit length used as arbitrary payload for the test.  **UINT16\_VALUE**: Bitfield of 16 bit length used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoBitfields** with the following parameters:    * **value1** is set to **UINT8\_VALUE**.    * **value2** is set to **UINT16\_VALUE**.    * **Value3** is set to **UINT32\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains: 2. Service ID is equal to **ETS\_ServiceID\_0**. 3. Method ID is equal to 0x41 (**echoBitfields**) 4. Return Code is equal to 0x00 (E\_OK). 5. Payload provides an UINT8 value equal to **UINT8\_VALUE** but in reverse bit order. 6. Payload provides an UINT16 value equal to **UINT16\_VALUE** but in reverse bit order. 7. Payload provides an UINT32 value equal to **UINT32\_VALUE** but in reverse bit order. |
| Reference | DS\_SP\_0077 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_07] Enumeration

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of enumeration parameters and their order (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **ENUM\_VALUE**: Value of type TestEnum using an 8-bit unsigned integer as underlying data type which is used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoENUM** with the following parameters:    * **value** is set to **ENUM\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains: 2. Service ID is equal to **ETS\_ServiceID\_0**. 3. Method ID is equal to 0x17 (**echoENUM**) 4. Return Code is equal to 0x00 (E\_OK). 5. Payload provides an UINT8 value equal to **UINT8\_VALUE**. |
| Reference | DS\_SP\_0076 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_08] Fixed Length Array of 1 dimension

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of an UINT8 array parameter with fixed length (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_ARRAY**: Array of UINT8 values used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoStaticUINT8Array** with the following parameters:    * **values** is set to **UINT8\_ARRAY**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains: 2. Service ID is equal to **ETS\_ServiceID\_0**. 3. Method ID is equal to 0x36 (**echoStaticUINT8Array**) 4. Return Code is equal to 0x00 (E\_OK). 5. Payload provides an UINT8 array equal to **UINT8\_ARRAY**. |
| Reference | DS\_SP\_0067 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_09] Fixed Length Array of 2 dimensions

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of a multidimensional UINT8 array parameter with fixed length (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_ARRAY**: Array of Fixed length array used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUINT8Array2Dim** with the following parameters:    * **values** is set to **UINT8\_ARRAY**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains: 2. Service ID is equal to ETS\_ServiceID\_0. 3. Method ID is equal to 0x35 (**echoUINT8Array2Dim**) 4. Return Code is equal to 0x00 (E\_OK). 5. Payload provides an UINT8 array equal to **UINT8\_ARRAY**. |
| Reference | DS\_SP\_0068 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_10] Fixed Length Array greater than expected

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Fixed Length Array greater than expected. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_ARRAY**: Unsigned integer value used as arbitrary payload for the test.  **UINT8\_ARRAY\_MODIFIED**: UINT8\_ARRAY additional with n+ Numbers |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoStaticUINT8Array** with the following parameters:    * **values** is set to **UINT8\_ARRAY\_MODIFIED.** 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x36 (**echoStaticUINT8Array**)    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 array equal to **UINT8\_ARRAY**. |
| Reference | DS\_SP\_0065 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_11] Fixed Length Array shorter than expected

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Fixed Length Array shorter than expected. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE\_ARRAY\_MODIEFIED**: Unsigned integer value, which is shorter as the specified array (consists of 5 UINT8 elements), as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoStaticUINT8Array** with the following parameters:    * **values** is set to **UINT8\_VALUE\_ARRAY\_ MODIEFIED**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x36 (**echoStaticUINT8Array**)    3. Return Code is equal to 0x09 (E\_MALFORMED\_MESSAGE). |
| Reference | DS\_SP\_0066 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_12] Dynamic Length Array of 1 dimension

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of an UINT8 array parameter (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **ARRAY \_LENGTH**: Length of UINT8 array value.  **UINT8\_ARRAY**: Array of UINT8 values used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUINT8Array** with the following parameters:    * **length** is set to number of elements contained in **ARRAY \_LENGTH**.    * **values** is set to **UINT8\_ARRAY**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x09 (**echoUINT8Array**)    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an array length equal to **ARRAY\_LENGTH**.    5. Payload provides an UINT8 array equal to **UINT8\_ARRAY**. |
| Reference | DS\_SP\_0069, DS\_SP\_0070, DS\_SP\_0073 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_13] Dynamic Length Array of 2 dimensions

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of a multidimensional UINT8 array parameter (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **ARRAY\_2DIM\_LENGTH:** Overall length of the two dimensional Array.  **UINT8\_2DIM\_ARRAY:** Array of UINT8 arrays where each subarray have got the length of ARRAY\_2DIM\_ELEMENT\_LENGTH. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUINT8Array2Dim** with the following parameters:    * **length** is set to **ARRAY\_2DIM\_LENGTH**.    * **values** is set to **UINT8\_2DIM\_ARRAY**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x35 (**echoUINT8Array2Dim**)    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an array length equal to **ARRAY\_2DIM\_LENGTH**.    5. Payload provides an Array of UINT8 Arrays equal to **UINT8\_2DIM\_ARRAY**. |
| Reference | DS\_SP\_0069, DS\_SP\_0070, DS\_SP\_0074 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_14] Dynamic Length Array with length field of 8 bits

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of an UINT8 array parameter with a length field of 8 bits (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **ARRAY \_LENGTH**: Length of UINT8 array value  **UINT8\_VALUE\_ARRAY:** UINT8 array with dynamic length |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUINT8Array8BitLength** with the following parameters:    * **length** is set to **ARRAY\_LENGTH**.    * **values** is set to **UINT8\_VALUE\_ARRAY**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x3E (**echoUINT8Array8BitLength**)    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an array length equal to **ARRAY\_LENGTH**.    5. Payload provides an UINT8 array equal to **UINT8\_VALUE\_ARRAY**. |
| Reference | DS\_SP\_0071, DS\_SP\_0072 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_15] Dynamic Length Array with length field of 16 bits

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of an UINT8 array parameter with a length field of 16 bits (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **ARRAY \_LENGTH**: Length of UINT8 array value  **UINT8\_VALUE\_ARRAY:** UINT8 array with dynamic length |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUINT8Array16BitLength** with the following parameters:    * **length** is set to **ARRAY\_LENGTH**.    * **values** is set to **UINT8\_VALUE\_ARRAY**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x3F (**echoUINT8Array16BitLength**) .    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an array length equal to **ARRAY\_LENGTH**.    5. Payload provides an UINT8 array equal to **UINT8\_VALUE\_ARRAY**. |
| Reference | DS\_SP\_0071, DS\_SP\_0072 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_16] Dynamic Length Array with length longer as message length allows it

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the DUT truncates the array when it is longer then expected. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **ARRAY\_LENGTH**: Array length greater as the SOME/IP message.  **UINT8\_VALUE\_ARRAY:** Array of UINT8 Values with the size according to ARRAY\_LENGTH.  **ARRAY\_LENGTH\_TRUNCATED**: Array length truncated to the SOME/IP message length.  **UINT8\_VALUE\_ARRAY\_TRUNCATED**: Array of UINT8 Values with the size according to ARRAY\_LENGTH\_TRUNCATED. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUINT8Array** with the following parameters:    * **length** is set to **ARRAY \_LENGTH**.    * **values** is set to **UINT8\_VALUE\_ARRAY**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x09 (**echoUINT8Array**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an array length equal to **ARRAY\_LENGTH\_TRUNCATED**.    5. Payload provides an UINT8 array equal to **UINT8\_VALUE\_ARRAY\_TRUNCATED**. |
| Reference | DS\_SP\_0075 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_17] Fixed Length UTF-8 String

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of an UTF-8 string parameter (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING\_VALUE**: Arbitrary non-empty character string with UTF8 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 3 byte values: 0xEF, 0xBB and 0xBF. Must be terminated with character “\0”. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF8FIXED** with the following parameters:    * **value** is set to **UTF8\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0013 (**echoUTF8FIXED**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides a character string equal to **UTF8\_STRING\_VALUE**. |
| Reference | DS\_SP\_0048, DS\_SP\_0052, DS\_SP\_0053, DS\_SP\_0054, DS\_SP\_0055 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_18] Fixed Length UTF-8 String greater than expected

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Fixed Length UTF-8 String greater than expected. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING\_VALUE**: Arbitrary non-empty character string with UTF8 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 3 byte values: 0xEF, 0xBB and 0xBF. Must be terminated with character “\0”.Length must be greater than the expected one of the used fixed length data type. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF8FIXED** with the following parameters:    * **value** is set to **UTF8\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0013 (**echoUTF8FIXED**).    3. Return Code is equal to 0x09 (E\_MALFORMED\_MESSAGE). |
| Reference | DS\_SP\_0056 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_19] Fixed Length UTF-8 String shorter than expected

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Fixed Length UTF-8 String shorter than expected. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING\_VALUE**: Arbitrary non-empty character string with UTF8 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 3 byte values: 0xEF, 0xBB and 0xBF. Must be terminated with character “\0”.Length must be shorter than the expected one of the used fixed length data type. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF8FIXED** with the following parameters:    * **value** is set to **UTF8\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0013 (**echoUTF8FIXED**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides a character string equal to **UTF8\_STRING\_VALUE**. |
| Reference | DS\_SP\_0057 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_20] Fixed Length UTF-8 String shorter than expected and not terminated

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Fixed Length UTF-8 String shorter than expected and not terminated. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING\_VALUE**: Arbitrary non-empty character string with UTF8 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 3 byte values: 0xEF, 0xBB and 0xBF. Must be non-terminated.Length must be shorter than the expected one of the used fixed length data type. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF8FIXED** with the following parameters:    * **value** is set to **UTF8\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0013 (**echoUTF8FIXED**).    3. Return Code is equal to 0x09 (E\_MALFORMED\_MESSAGE). |
| Reference | DS\_SP\_0058 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_21] Dynamic Length UTF-8 String

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of an UTF-8 string parameter with dynamic length (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRING\_LENGTH:** (UINT32)Length of the UTF8 string used for the test.  **UTF8\_STRING\_VALUE**: Arbitrary non-empty character string with UTF8 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 3 byte values: 0xEF, 0xBB and 0xBF. Must be terminated with character “\0”. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF8DYNAMIC** with the following parameters:    * **length** is set to **STRING\_LENGTH**.    * **value** is set to **UTF8\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse**  and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0015 (**echoUTF8DYNAMIC**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT32 value equal to **STRING\_LENGTH**.    5. Payload provides a character string equal to **UTF8\_STRING\_VALUE**. |
| Reference | DS\_SP\_0048, DS\_SP\_0052, DS\_SP\_0053, DS\_SP\_0059, DS\_SP\_0060, DS\_SP\_0061, DS\_SP\_0063, DS\_SP\_0064 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_22] Dynamic Length UTF-8 String with length field of 8 bits

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Dynamic Length UTF-8 String with length field of 8 bits. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRING\_LENGTH:** (UINT8)Length of the UTF8 string used for the test.  **UTF8\_STRING\_VALUE**: Arbitrary non-empty character string with UTF8 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 3 byte values: 0xEF, 0xBB and 0xBF. Must be terminated with character “\0”. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF8DYNAMIC8BitLength** with the following parameters:    * **length** is set to **STRING\_LENGTH**.    * **value** is set to **UTF8\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0510 (**echoUTF8DYNAMIC8BitLength**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **STRING\_LENGTH**.    5. Payload provides a character string equal to **UTF8\_STRING\_VALUE**. |
| Reference | DS\_SP\_0062 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_23] Dynamic Length UTF-8 String with length field of 16 bits

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Dynamic Length UTF-8 String with length field of 16 bits. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRING\_LENGTH:** Length of the UTF8 string used for the test.  **UTF8\_STRING\_VALUE**: Arbitrary non-empty character string with UTF8 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 3 byte values: 0xEF, 0xBB and 0xBF. Must be terminated with character “\0”. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF8DYNAMIC16BitLength** with the following parameters:    * **length** is set to **STRING\_LENGTH**.    * **value** is set to **UTF8\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0511 (**echoUTF8DYNAMIC16BitLength**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT16 value equal to **STRING\_LENGTH**.    5. Payload provides a character string equal to **UTF8\_STRING\_VALUE**. |
| Reference | DS\_SP\_0062 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_24] Fixed Length UTF-16BE String

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of an UTF-16BE string parameter with fixed length (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF16BE\_STRING\_VALUE:** Arbitrary non-empty character string with UTF16 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 2 byte values: 0xFE and 0xFF. Must be terminated with character “\0”. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF16FIXED** with the following parameters:    * **value** is set to **UTF16BE\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0.**    2. Method ID is equal to 0x0014 (**echoUTF16FIXED**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides a character string equal to **UTF16BE\_STRING\_VALUE**. |
| Reference | DS\_SP\_0048, DS\_SP\_0049, DS\_SP\_0052, DS\_SP\_0053, DS\_SP\_0054, DS\_SP\_0055 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_25] Fixed Length UTF-16BE String greater than expected

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Fixed Length UTF-16BE String greater than expected. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF16BE\_STRING\_VALUE:** Arbitrary non-empty character string with UTF16 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 2 byte values: 0xFE and 0xFF. Must be terminated with character “\0”. Length must be greater than the expected one of the used fixed length data type. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF16FIXED** with the following parameters:    * **value** is set to **UTF16BE\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0014 (**echoUTF16FIXED**).    3. Return Code is equal to 0x09 (E\_MALFORMED\_MESSAGE). |
| Reference | DS\_SP\_0056 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_26] Fixed Length UTF-16BE String shorter than expected

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Fixed Length UTF-16BE String shorter than expected. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF16BE\_STRING\_VALUE:** Arbitrary non-empty character string with UTF16 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 2 byte values: 0xFE and 0xFF. Must be terminated with character “\0”. Length must be shorter than the expected one of the used fixed length data type. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF16FIXED** with the following parameters:    * **value** is set to **UTF16BE\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0014 (**echoUTF16FIXED**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides a character string equal to **UTF16BE\_STRING\_VALUE**. |
| Reference | DS\_SP\_0057 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_27] Fixed Length UTF-16BE String shorter than expected and not terminated

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Fixed Length UTF-16BE String shorter than expected and not terminated. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF16BE\_STRING\_VALUE:** Arbitrary non-empty character string with UTF16 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 2 byte values: 0xFE and 0xFF. Must be non-terminated.Length must be shorter than the expected one of the used fixed length data type. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF16FIXED** with the following parameters:    * **value** is set to **UTF16BE\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0014 (**echoUTF16FIXED**).    3. Return Code is equal to 0x09 (E\_MALFORMED\_MESSAGE). |
| Reference | DS\_SP\_0058 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_28] Dynamic Length UTF-16BE String

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of an UTF-16BE string parameter (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRING\_LENGTH:** (UINT32)Length of the UTF16BE string used for the test.  **UTF16BE\_STRING\_VALUE**: Arbitrary non-empty character string with UTF16 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 2 byte values: 0xFE and 0xFF. Must be terminated with character “\0”. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF16DYNAMIC** with the following parameters:    * **length** is set to **STRING\_LENGTH**.    * **value** is set to **UTF16BE\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0016 (**echoUTF16DYNAMIC**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT32 value equal to **STRING\_LENGTH**.    5. Payload provides a character string equal to **UTF16BE\_STRING\_VALUE**. |
| Reference | DS\_SP\_0048, DS\_SP\_0049, DS\_SP\_0052, DS\_SP\_0053, DS\_SP\_0059, DS\_SP\_0060, DS\_SP\_0061, DS\_SP\_0063, DS\_SP\_0064 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_29] Dynamic Length UTF-16BE String with length field of 8 bits

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Dynamic Length UTF-16BE String with length field of 8 bits. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRING\_LENGTH:** (UINT8)Length of the UTF16BE string used for the test.  **UTF16BE\_STRING\_VALUE**: Arbitrary non-empty character string with UTF16 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 2 byte values: 0xFE and 0xFF. Must be terminated with character “\0”. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF16DYNAMIC8BitLength** with the following parameters:    * **length** is set to **STRING\_LENGTH**.    * **value** is set to **UTF16BE\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0512 (**echoUTF16DYNAMIC8BitLength**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **STRING\_LENGTH**.    5. Payload provides a character string equal to **UTF16BE\_STRING\_VALUE**. |
| Reference | DS\_SP\_0062 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_30] Dynamic Length UTF-16BE String with length field of 16 bits

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Dynamic Length UTF-16BE String with length field of 16 bits. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRING\_LENGTH:** (UINT16)Length of the UTF16BE string used for the test  **UTF16BE\_STRING\_VALUE**: Arbitrary non-empty character string with UTF16 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 2 byte values: 0xFE and 0xFF. Must be terminated with character “\0”. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF16DYNAMIC16BitLength** with the following parameters:    * **length** is set to **STRING\_LENGTH**.    * **value** is set to **UTF16BE\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0513 (**echoUTF16DYNAMIC16BitLength**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT16 value equal to **STRING\_LENGTH**.    5. Payload provides a character string equal to **UTF16BE\_STRING\_VALUE**. |
| Reference | DS\_SP\_0062 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_31] Dynamic Length UTF-16BE String with odd number of bytes

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Dynamic Length UTF-16BE String with odd number of bytes. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRING\_LENGTH:** (UINT32)Length of the UTF16BE string used for the test  **UTF16BE\_STRING\_VALUE**: Arbitrary non-empty character string with UTF16 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 2 byte values: 0xFE and 0xFF. Must be terminated with character “\0”.  **ODD\_STRING\_VALUE**: Value of test variable UTF16BE\_STRING\_VALUE with a single arbitrary byte value appended. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF16DYNAMIC** with the following parameters:    * **length** is set to **STRING\_LENGTH**.    * **value** is set to **ODD\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0016 (**echoUTF16DYNAMIC**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT32 value equal to **STRING\_LENGTH**.    5. Payload provides a character string equal to **UTF16BE\_STRING\_VALUE**. |
| Reference | DS\_SP\_0050, DS\_SP\_0051 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_32] Fixed Length UTF-16LE String

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of an UTF-16LE string parameter with fixed length (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF16LE\_STRING\_VALUE:** Arbitrary non-empty character string with UTF16 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 2 byte values: 0xFF and 0xFE. Must be terminated with character “\0”. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF16FIXED\_LE** with the following parameters:    * **value** is set to **UTF16LE\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x07D3 (**echoUTF16FIXED\_LE**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides a character string equal to **UTF16LE\_STRING\_VALUE**. |
| Reference | DS\_SP\_0048, DS\_SP\_0049, DS\_SP\_0052, DS\_SP\_0053, DS\_SP\_0054, DS\_SP\_0055 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_33] Fixed Length UTF-16LE String greater than expected

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Fixed Length UTF-16LE String greater than expected. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF16LE\_STRING\_VALUE:** Arbitrary non-empty character string with UTF16 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 2 byte values: 0xFF and 0xFE. Must be terminated with character “\0”. Length must be greater than the expected one of the used fixed length data type. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF16FIXED\_LE** with the following parameters:    * **value** is set to **UTF16LE\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x07D3 (**echoUTF16FIXED\_LE**).    3. Return Code is equal to 0x09 (E\_MALFORMED\_MESSAGE). |
| Reference | DS\_SP\_0056 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_34] Fixed Length UTF-16LE String shorter than expected

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Fixed Length UTF-16LE String shorter than expected. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF16LE\_STRING\_VALUE:** Arbitrary non-empty character string with UTF16 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 2 byte values: 0xFF and 0xFE. Must be terminated with character “\0”. Length must be shorter than the expected one of the used fixed length data type. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF16FIXED\_LE** with the following parameters:    * **value** is set to **UTF16LE\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x07D3 (**echoUTF16FIXED\_LE**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides a character string equal to **UTF16LE\_STRING\_VALUE**. |
| Reference | DS\_SP\_0057 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_35] Fixed Length UTF-16LE String shorter than expected and not terminated

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Fixed Length UTF-16LE String shorter than expected and not terminated. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF16LE\_STRING\_VALUE:** Arbitrary non-empty character string with UTF16 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 2 byte values: 0xFF and 0xFE. Must be non-terminated.Length must be shorter than the expected one of the used fixed length data type. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF16FIXED\_LE** with the following parameters:    * **value** is set to **UTF16LE\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x07D3 (**echoUTF16FIXED\_LE**).    3. Return Code is equal to 0x09 (E\_MALFORMED\_MESSAGE). |
| Reference | DS\_SP\_0058 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_36] Dynamic Length UTF-16LE String

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of an UTF-16LE string parameter (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRING\_LENGTH:** (UINT32)Length of the UTF16LE string used for the test.  **UTF16LE\_STRING\_VALUE:** Arbitrary non-empty character string with UTF16 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 2 byte values: 0xFF and 0xFE. Must be terminated with character “\0”. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF16DYNAMIC\_LE** with the following parameters:    * **length** is set to **STRING\_LENGTH**.    * **value** is set to **UTF16LE\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x07D4 (**echoUTF16DYNAMIC\_LE**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT32 value equal to **STRING\_LENGTH**.    5. Payload provides a character string equal to **UTF16LE\_STRING\_VALUE**. |
| Reference | DS\_SP\_0048, DS\_SP\_0049, DS\_SP\_0052, DS\_SP\_0053, DS\_SP\_0059, DS\_SP\_0060, DS\_SP\_0061, DS\_SP\_0063, DS\_SP\_0064 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_37] Dynamic Length UTF-16LE String with length field of 8 bits

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Dynamic Length UTF-16LE String with length field of 8 bits. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRING\_LENGTH**: (UINT8)Length of the UTF16LE string used for the test.  **UTF16LE\_STRING\_VALUE:** Arbitrary non-empty character string with UTF16 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 2 byte values: 0xFF and 0xFE. Must be terminated with character “\0”. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF16DYNAMIC8BitLength\_LE** with the following parameters:    * **length** is set to **STRING\_LENGTH**.    * **value** is set to **UTF16LE\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x07D5 (**echoUTF16DYNAMIC8BitLength\_LE**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **STRING\_LENGTH**.    5. Payload provides a character string equal to **UTF16LE\_STRING\_VALUE**. |
| Reference | DS\_SP\_0062 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_38] Dynamic Length UTF-16LE String with length field of 16 bits

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Dynamic Length UTF-16LE String with length field of 16 bits. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRING\_LENGTH:** (UINT16)Length of the UTF16LE string used for the test.  **UTF16LE\_STRING\_VALUE:** Arbitrary non-empty character string with UTF16 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 2 byte values: 0xFF and 0xFE. Must be terminated with character “\0”. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF16DYNAMIC16BitLength\_LE** with the following parameters:    * **length** is set to **STRING\_LENGTH**.    * **value** is set to **UTF16LE\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x07D6 (**echoUTF16DYNAMIC16BitLength\_LE**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT16 value equal to **STRING\_LENGTH**.    5. Payload provides a character string equal to **UTF16LE\_STRING\_VALUE**. |
| Reference | DS\_SP\_0062 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_39] Dynamic Length UTF-16LE String with odd number of bytes

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Dynamic Length UTF-16LE String with odd number of bytes. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRING\_LENGTH:** (UINT32)Length of the UTF16LE string used for the test.  **UTF16LE\_STRING\_VALUE:** Arbitrary non-empty character string with UTF16 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 2 byte values: 0xFF and 0xFE. Must be terminated with character “\0”.  **ODD\_STRING\_VALUE**: Value of test variable UTF16LE\_STRING\_VALUE with a single arbitrary byte value appended. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoUTF16DYNAMIC\_LE** with the following parameters:    * **length** is set to **STRING\_LENGTH +1**.    * **value** is set to **ODD\_STRING\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x07D4 (**echoUTF16DYNAMIC\_LE**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT32 value equal to **STRING\_LENGTH**.    5. Payload provides a character string equal to **UTF16LE\_STRING\_VALUE**. |
| Reference | DS\_SP\_0050, DS\_SP\_0051 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_40] Struct with fixed length members

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of a struct parameter where all members are basic datatypes (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRUCT\_FIXED\_VALUE:** Fixed struct value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoStructFixed** with the following parameters:    * **value** is set to **STRUCT\_FIXED\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0505 (**echoStructFixed**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides a value of struct **TestStructFixed** equal to **STRUCT\_FIXED\_VALUE**. |
| Reference | DS\_SP\_0042, DS\_SP\_0047 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_41] Struct with dynamic length members

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of a struct parameter where some members are dynamic length datatypes (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRUCT\_DYNAMIC\_VALUE:** Dynamic struct value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoStructDynamic** with the following parameters:    * **value** is set to **STRUCT\_DYNAMIC\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0509 (**echoStructDynamic**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides a value of struct **TestStructDynamic** equal to **STRUCT\_DYNAMIC\_VALUE**. |
| Reference | DS\_SP\_0042, DS\_SP\_0047 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_42] Struct with struct members

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of a struct parameter where some members are nested struct datatypes (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRUCT\_NESTED\_VALUE:** Nested struct value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoStructNested** with the following parameters:    * **value** is set to **STRUCT\_NESTED\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x050A (**echoStructNested**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides a value of struct **TestStructNested** equal to **STRUCT\_NESTED\_VALUE**. |
| Reference | DS\_SP\_0042, DS\_SP\_0047 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_43] Struct with length field of 8 bits

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of a struct parameter with a length field of 8 bits (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRUCT\_LENGTH:** (UINT8)Length of the struct value used for the test.  **STRUCT\_FIXED\_VALUE:** Fixed struct value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoStructFixed8BitLength** with the following parameters:    * **value** is set to **STRUCT\_FIXED\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0506 (**echoStructFixed8BitLength**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **STRUCT\_LENGTH**.    5. Payload provides a value of struct **TestStructFixed** equal to **STRUCT\_FIXED\_VALUE**. |
| Reference | DS\_SP\_0043, DS\_SP\_0044 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_44] Struct with length field of 16 bits

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of a struct parameter with a length field of 16 bits (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRUCT\_LENGTH:** (UINT16)Length of the struct value used for the test.  **STRUCT\_FIXED\_VALUE:** Fixed struct value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoStructFixed16BitLength** with the following parameters:    * **value** is set to **STRUCT\_FIXED\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0507 (**echoStructFixed16BitLength**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT16 value equal to **STRUCT\_LENGTH**.    5. Payload provides a value of struct **TestStructFixed** equal to **STRUCT\_FIXED\_VALUE**. |
| Reference | DS\_SP\_0043, DS\_SP\_0044 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_45] Struct with length field of 32 bits

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks processing of a struct parameter with a length field of 32 bits (sending and receiving). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRUCT\_LENGTH:** (UINT32)Length of the struct value used for the test.  **STRUCT\_FIXED\_VALUE:** Fixed struct value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoStructFixed32BitLength** with the following parameters:    * **value** is set to **STRUCT\_FIXED\_VALUE**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0508 (**echoStructFixed32BitLength**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT32 value equal to **STRUCT\_LENGTH**.    5. Payload provides a value of struct **TestStructFixed** equal to **STRUCT\_FIXED\_VALUE**. |
| Reference | DS\_SP\_0043, DS\_SP\_0044 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_46] Struct is greater than expected

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to successfully deserialize the message, even when length of received struct value is greater than specified by data type definition. Only the known bytes shall be interpreted. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRUCT\_LENGTH:** 8-bit unsigned integer which specifies the regular length of struct value used for the test.  **STRUCT\_LENGTH\_MODIFIED:** 8-bit unsigned integer which specifies the modified length of the struct value used for the test. Must be greater than the regular length of data type TestStructFixed (10 bytes).  **STRUCT\_FIXED\_VALUE:** Fixed struct value used as arbitrary payload for the test.  **STRUCT\_FIXED\_VALUE\_MODIFIED:** Conforms to value of test variable given above but with additional bytes of arbitrary value appended. Length must be equal to value of test variable STRUCT\_LENGTH\_MODIFIED. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoStructFixed8BitLength** with the following parameters:    * **value** is set to **STRUCT\_FIXED\_VALUE\_MODIFIED**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0506 (**echoStructFixed8BitLength**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides a value of struct **TestStructFixed** equal to **STRUCT\_FIXED\_VALUE**. |
| Reference | DS\_SP\_0045 |
| Notes |  |

#### [SOMEIPGEN\_DATA\_47] Struct is shorter than expected

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to abort deserialization and treat the message as malformed when length of received struct value is shorter than specified by data type definition. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRUCT\_LENGTH\_MODIFIED:** 8-bit unsigned integer which specifies the modified length of the struct value used for the test. Must be shorter than the regular length of data type TestStructFixed (10 bytes).  **STRUCT\_FIXED\_VALUE\_MODIFIED:** Fixed struct value used as arbitrary payload for the test. Must be truncated according to value of test variable given above. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **echoStructFixed8BitLength** with the following parameters:    * **value** is set to **STRUCT\_FIXED\_VALUE\_MODIFIED**. 2. DUT: Sends the corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0506 (**echoStructFixed8BitLength**).    3. Return Code is equal to 0x09 (E\_MALFORMED\_MESSAGE). |
| Reference | DS\_SP\_0046 |
| Notes |  |

### Transport Protocol

#### [SOMEIPGEN\_TP\_01] SOME/IP-TP segments message which is too long for single UDP datagram

|  |  |
| --- | --- |
| Item | Description |
| Purpose | SOME/IP-TP segments message which is to long for single UDP datagram. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT shall support message segmentation via SOME/IP-TP. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_ARRAY**: Sequence of 5000 UINT8 values used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **setUINT8ArrayReliable** with the following parameters:    * **dataLength** is set to 0x00001388 (5000).    * **data** is set to **UINT8\_ARRAY**. 2. TESTER: Sends SOME/IP Request message for getter of field **TestFieldUINT8Array**. 3. DUT: Sends 1st segment of corresponding SOME/IP Response message. 4. TESTER: Verify received SOME/IP-TP segment. 5. DUT: Sends 2nd segment of corresponding SOME/IP Response message. 6. TESTER: Verify received SOME/IP-TP segment. 7. DUT: Sends 3rd segment of corresponding SOME/IP Response message. 8. TESTER: Verify received SOME/IP-TP segment. 9. DUT: Sends last segment of corresponding SOME/IP Response message. 10. TESTER: Verify received SOME/IP-TP segment. |
| Verification | 1. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**). 2. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**). 3. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**). 4. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**). |
| Reference | DS\_SP\_0122 |
| Notes |  |

#### [SOMEIPGEN\_TP\_02] SOME/IP-TP message use same header field values as the original message

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a SOME/IP-TP message use same header field values as the original message. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT shall support message segmentation via SOME/IP-TP. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_ARRAY**: Sequence of 5000 UINT8 values used as arbitrary payload for the test.  **REQUEST\_ID\_1**: Request ID of the original message SOME/IP service |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **setUINT8ArrayReliable** with the following parameters:    * **dataLength** is set to 0x00001388 (5000).    * **data** is set to **UINT8\_ARRAY**. 2. TESTER: Sends SOME/IP Request message for getter of field **TestFieldUINT8Array**. 3. DUT: Sends 1st segment of corresponding SOME/IP Response message. 4. TESTER: Verify received SOME/IP-TP segment. |
| Verification | 1. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. Request ID is equal to **REQUEST\_ID\_1.**    4. Protocol Version is equal to 0x01.    5. Interface Version is equal to **InterfaceVersion**.    6. Return Code is equal to 0x00 (E\_OK) |
| Reference | DS\_SP\_0156 |
| Notes |  |

#### [SOMEIPGEN\_TP\_03] SOME/IP-TP message length

|  |  |
| --- | --- |
| tem | Description |
| Purpose | Checks for proper length of SOME/IP message carrying the 1st SOME/IP-TP segment. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT shall support message segmentation via SOME/IP-TP. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_ARRAY**: Sequence of 5000 UINT8 values used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **setUINT8ArrayReliable** with the following parameters:    * **dataLength** is set to 0x00001388 (5000).    * **data** is set to **UINT8\_ARRAY**. 2. TESTER: Sends SOME/IP Request message for getter of field **TestFieldUINT8Array**. 3. DUT: Sends 1st segment of corresponding SOME/IP Response message. 4. TESTER: Verify received SOME/IP-TP segment. |
| Verification | 1. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. Length is equal to 0x0000057C (1404). |
| Reference | DS\_SP\_0153 |
| Notes |  |

#### [SOMEIPGEN\_TP\_04] SOME/IP-TP message shall carry a nonzero Session ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | SOME/IP-TP message shall carry a nonzero Session ID. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT shall support message segmentation via SOME/IP-TP. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_ARRAY**: Sequence of 5000 UINT8 values used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **setUINT8ArrayReliable** with the following parameters:    * **dataLength** is set to 0x00001388 (5000).    * **data** is set to **UINT8\_ARRAY**. 2. TESTER: Sends SOME/IP Request message for getter of field **TestFieldUINT8Array**. 3. DUT: Sends 1st segment of corresponding SOME/IP Response message. 4. TESTER: Verify received SOME/IP-TP segment. |
| Verification | 1. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. Session ID is equal to the value of the preceding request message. |
| Reference | DS\_SP\_0149 |
| Notes |  |

#### [SOMEIPGEN\_TP\_05] All SOME/IP-TP segments shall have same session ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | All SOME/IP-TP segments shall have same session ID. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT shall support message segmentation via SOME/IP-TP. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_ARRAY**: Sequence of 5000 UINT8 values used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **setUINT8ArrayReliable** with the following parameters:    * **dataLength** is set to 0x00001388 (5000).    * **data** is set to **UINT8\_ARRAY**. 2. TESTER: Sends SOME/IP Request message for getter of field **TestFieldUINT8Array**. 3. DUT: Sends 1st segment of corresponding SOME/IP Response message. 4. TESTER: Verify received SOME/IP-TP segment. 5. DUT: Sends 2nd segment of corresponding SOME/IP Response message. 6. TESTER: Verify received SOME/IP-TP segment. 7. DUT: Sends 3rd segment of corresponding SOME/IP Response message. 8. TESTER: Verify received SOME/IP-TP segment. 9. DUT: Sends last segment of corresponding SOME/IP Response message. 10. TESTER: Verify received SOME/IP-TP segment. |
| Verification | 1. SOME/IP-TP segment is received within **TimeoutResponse**.    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. Session ID is equal to the value of the preceding request message. 2. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. Session ID is same as the one of previous segment. 3. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. Session ID is same as the one of previous segment. 4. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. Session ID is same as the one of previous segment. |
| Reference | DS\_SP\_0150 |
| Notes |  |

#### [SOMEIPGEN\_TP\_06] SOME/IP-TP segments provides expected offset value

|  |  |
| --- | --- |
| Item | Description |
| Purpose | SOME/IP-TP segments provides expected offset value. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT shall support message segmentation via SOME/IP-TP. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_ARRAY**: Sequence of 5000 UINT8 values used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **setUINT8ArrayReliable** with the following parameters:    * **dataLength** is set to 0x00001388 (5000).    * **data** is set to **UINT8\_ARRAY**. 2. TESTER: Sends SOME/IP Request message for getter of field **TestFieldUINT8Array**. 3. DUT: Sends 1st segment of corresponding SOME/IP Response message. 4. TESTER: Verify received SOME/IP-TP segment. 5. DUT: Sends 2nd segment of corresponding SOME/IP Response message. 6. TESTER: Verify received SOME/IP-TP segment. 7. DUT: Sends 3rd segment of corresponding SOME/IP Response message. 8. TESTER: Verify received SOME/IP-TP segment. 9. DUT: Sends last segment of corresponding SOME/IP Response message. 10. TESTER: Verify received SOME/IP-TP segment. |
| Verification | 1. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. Offset is equal to 0 2. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. Offset is equal to 0x0000057 (87). 3. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. Offset is equal to 0x00000AE (174). 4. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. Offset is equal to 0x0000105 (261). |
| Reference | DS\_SP\_0151, DS\_SP\_0152 |
| Notes |  |

#### [SOMEIPGEN\_TP\_07] SOME/IP-TP segments sets More Segments Flag properly.

|  |  |
| --- | --- |
| Item | Description |
| Purpose | SOME/IP-TP segment set More Segments Flag set to 1 and the last Segment is set to 0. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT shall support message segmentation via SOME/IP-TP. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_ARRAY**: Sequence of 5000 UINT8 values used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **setUINT8ArrayReliable** with the following parameters:    * **dataLength** is set to 0x00001388 (5000).    * **data** is set to **UINT8\_ARRAY**. 2. TESTER: Sends SOME/IP Request message for getter of field **TestFieldUINT8Array**. 3. DUT: Sends 1st segment of corresponding SOME/IP Response message. 4. TESTER: Verify received SOME/IP-TP segment. 5. DUT: Sends 2nd segment of corresponding SOME/IP Response message. 6. TESTER: Verify received SOME/IP-TP segment. 7. DUT: Sends 3rd segment of corresponding SOME/IP Response message. 8. TESTER: Verify received SOME/IP-TP segment. 9. DUT: Sends last segment of corresponding SOME/IP Response message. 10. TESTER: Verify received SOME/IP-TP segment. |
| Verification | 1. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. More Segments Flag is set to 1. 2. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. More Segments Flag is set to 1. 3. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. More Segments Flag is set to 1. 4. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. More Segments Flag is set to 0. |
| Reference | DS\_SP\_0151 |
| Notes |  |

#### [SOMEIPGEN\_TP\_08] Length of SOME/IP-TP segments shall be multiple of 16 bytes

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Length of SOME/IP-TP segments shall be multiple of 16 bytes. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT shall support message segmentation via SOME/IP-TP. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_ARRAY**: Sequence of 5000 UINT8 values used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **setUINT8ArrayReliable** with the following parameters:    * **dataLength** is set to 0x00001388 (5000).    * **data** is set to **UINT8\_ARRAY**. 2. TESTER: Sends SOME/IP Request message for getter of field **TestFieldUINT8Array**. 3. DUT: Sends 1st segment of corresponding SOME/IP Response message. 4. TESTER: Verify received SOME/IP-TP segment. 5. DUT: Sends 2nd segment of corresponding SOME/IP Response message. 6. TESTER: Verify received SOME/IP-TP segment. 7. DUT: Sends 3rd segment of corresponding SOME/IP Response message. 8. TESTER: Verify received SOME/IP-TP segment. 9. DUT: Sends last segment of corresponding SOME/IP Response message. 10. TESTER: Verify received SOME/IP-TP segment. |
| Verification | 1. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. Length - 12 is a multiple of 16. 2. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. Length - 12 is a multiple of 16. 3. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. Length - 12 is a multiple of 16. 4. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**). |
| Reference | DS\_SP\_0154 |
| Notes |  |

#### [SOMEIPGEN\_TP\_09] Maximum SOME/IP-TP segment length

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks length of each SOME/IP-TP segment. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT shall support message segmentation via SOME/IP-TP. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_ARRAY**: Sequence of 5000 UINT8 values used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **setUINT8ArrayReliable** with the following parameters:    * **dataLength** is set to 0x00001388 (5000).    * **data** is set to **UINT8\_ARRAY**. 2. TESTER: Sends SOME/IP Request message for getter of field **TestFieldUINT8Array**. 3. DUT: Sends 1st segment of corresponding SOME/IP Response message. 4. TESTER: Verify received SOME/IP-TP segment. 5. DUT: Sends 2nd segment of corresponding SOME/IP Response message. 6. TESTER: Verify received SOME/IP-TP segment. 7. DUT: Sends 3rd segment of corresponding SOME/IP Response message. 8. TESTER: Verify received SOME/IP-TP segment. 9. DUT: Sends last segment of corresponding SOME/IP Response message. 10. TESTER: Verify received SOME/IP-TP segment. |
| Verification | 1. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. Length is equal to 0x0000057C (1404). 2. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. Length is equal to 0x0000057C (1404). 3. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. Length is equal to 0x0000057C (1404). 4. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. Length is equal to 0x00000348 (840). |
| Reference | DS\_SP\_0155 |
| Notes |  |

#### [SOMEIPGEN\_TP\_10] Receive segmented SOME/IP message

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Receive segmented SOME/IP message. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT shall support message segmentation via SOME/IP-TP. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_ARRAY\_DATA:** Sequence of 4000 UINT8 values used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Creates 3 SOME/IP-TP segments for a SOME/IP Request message of ETS method **echoUINT8Array** with the following parameters:    * **arrayLength** is set to 0x00000FA0 (4000).    * **array** is set to **UINT8\_ARRAY\_DATA**. 2. TESTER: Sends 1st segment of SOME/IP Request message. 3. TESTER: Sends 2nd segment of SOME/IP Request message. 4. TESTER: Sends last segment of SOME/IP Request message. 5. DUT: Sends 1st segment of corresponding SOME/IP Response message. 6. DUT: Sends 2nd segment of corresponding SOME/IP Response message. 7. DUT: Sends last segment of corresponding SOME/IP Response message. 8. TESTER: Reassembles SOME/IP Response message from received SOME/IP-TP segments. 9. TESTER: Verify reassembled SOME/IP Response message. |
| Verification | 1. All SOME/IP-TP segments are received within **TimeoutResponse**. 2. Reassembled SOME/IP Response message contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0009 (9).    3. Payload provides UINT8 array with a length of 0x00000FA0 (4000) and data equal to **UINT8\_ARRAY\_DATA**. |
| Reference | DS\_SP\_0122 |
| Notes |  |

#### [SOMEIPGEN\_TP\_11] Ignore reserved field of SOME/IP-TP message

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Ignore reserved field of SOME/IP-TP message. To ensure that there is no negative influence to the SOME/IP behavior |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT shall support message segmentation via SOME/IP-TP. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_ARRAY\_DATA:** Sequence of 4000 UINT8 values used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Creates 3 SOME/IP-TP segments for a SOME/IP Request message of ETS method **echoUINT8Array** with the following parameters:    * **arrayLength** is set to 0x00000FA0 (4000).    * **array** is set to **UINT8\_ARRAY\_DATA**. 2. TESTER: Sends 1st segment of SOME/IP Request message with Reserved Flags set to 1. 3. TESTER: Sends 2nd segment of SOME/IP Request message. 4. TESTER: Sends last segment of SOME/IP Request message with Reserved Flags set to 1. 5. DUT: Sends 1st segment of corresponding SOME/IP Response message. 6. DUT: Sends 2nd segment of corresponding SOME/IP Response message. 7. DUT: Sends last segment of corresponding SOME/IP Response message. 8. TESTER: Reassembles SOME/IP Response message from received SOME/IP-TP segments. 9. TESTER: Verify reassembled SOME/IP Response message. |
| Verification | 1. All SOME/IP-TP segments are received within **TimeoutResponse**. 2. Reassembled SOME/IP Response message contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0009 (9).    3. Payload provides UINT8 array with a length of 0x00000FA0 (4000) and data equal to **UINT8\_ARRAY\_DATA**. |
| Reference | DS\_SP\_0151 |
| Notes |  |

### Service Discovery Message Format

#### [SOMEIPGEN\_SD\_FORMAT\_01] Message length shall be smaller than 1400

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Message length shall be smaller than 1400. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends multicast SOME/IP-SD message. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Length is smaller or equal to 1400. |
| Reference | DS\_SP\_0038 |
| Notes | The test case expects to check the first message a SOME/IP server will send out, which should be actual a SOME/IP-SD message. Therefore appropriate values of a valid SOME/IP-SD message header are assumed for verification. |

#### [SOMEIPGEN\_SD\_FORMAT\_02] Unicast Flag

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a SOME/IP-SD message provides a bit flag which indicates if receiving using unicast is supported. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends multicast SOME/IP-SD message. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Unicast Flag is set to 1. |
| Reference | DS\_SD\_0019, DS\_SD\_0020, DS\_SD\_0028, DS\_SD\_0029 |
| Notes |  |

#### [SOMEIPGEN\_SD\_FORMAT\_03] Explicit Initial Data Control Flag

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a SOME/IP-SD message provides a bit flag which indicates if explicit initial data control is supported. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends multicast SOME/IP-SD message. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Explicit Initial Data Control Flag is set to 1. |
| Reference | DS\_SD\_0019, DS\_SD\_0020, DS\_SD\_0030, DS\_SD\_0031 |
| Notes |  |

#### [SOMEIPGEN\_SD\_FORMAT\_04] Undefined bits in the flag field

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that undefined bits in the flag field of a SOME/IP-SD message are statically set to zero. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends multicast SOME/IP-SD message. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Message ID is equal to 0xFFFF8100.    2. All undefined bits within flag field are set to 0. |
| Reference | DS\_SD\_0019, DS\_SD\_0020, DS\_SD\_0032 |
| Notes |  |

#### [SOMEIPGEN\_SD\_FORMAT\_05] Reserved bits

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that reserved bits of a SOME/IP-SD message are statically set to null. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends multicast SOME/IP-SD message. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Message ID is equal to 0xFFFF8100.    2. All bits within reserved bits field are set to 0. |
| Reference | DS\_SD\_0019, DS\_SD\_0033 |
| Notes |  |

#### [SOMEIPGEN\_SD\_FORMAT\_06] Entries Array

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a SOME/IP-SD message provides an Entries Array of dynamic length. The array starts with an UINT32 field counting the number of bytes of the following data. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends multicast SOME/IP-SD message. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Length of Entries Array is:       * equal to the remaining message length reduced by 4, In case of empty Options Array.       * smaller than the remaining message length reduced by 4, In case of non empty Options Array. |
| Reference | DS\_SD\_0019, DS\_SD\_0034, DS\_SD\_0037 |
| Notes |  |

#### [SOMEIPGEN\_SD\_FORMAT\_07] Options Array

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a SOME/IP-SD message provides an Options Array of dynamic length. The array starts with an UINT32 field counting the number of bytes of the following data. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends multicast SOME/IP-SD message. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Length of Options Array is equal to the remaining message length. |
| Reference | DS\_SD\_0019, DS\_SD\_0036, DS\_SD\_0037 |
| Notes |  |

#### [SOMEIPGEN\_SD\_FORMAT\_08] Supported Entry Types

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that an Entries Array of a SOME/IP-SD message provides supported entry types only. Every entry starts with an UINT8 field encoding the type of the entry. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends multicast SOME/IP-SD message. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Length of Entries Array is not equal to 0x00000000.    3. Type of all entries in Entries Array conforms to one of the following values:       * 0x00 (Find Service)       * 0x01 (Offer Service / Stop Offer Service)       * 0x06 (Subscribe Eventgroup / Stop Subscribe Eventgroup)       * 0x07 (Subscribe Eventgroup Ack / Nack) |
| Reference | DS\_SD\_0040, DS\_SD\_0041, DS\_SD\_0118 |
| Notes |  |

#### [SOMEIPGEN\_SD\_FORMAT\_09] Supported Option Types

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that an Options Array of a SOME/IP-SD message provides supported option types only. Every option starts with an UINT16 field followed by an UINT8 field. The first one specifies the size of the option in bytes and the second one encodes the type of the option. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends multicast SOME/IP-SD message. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Length of Options Array is not equal to 0x00000000.    3. Type of all options in Options Array conforms to one of the following values:       * 0x04 (IPv4 Endpoint Option)       * 0x14 (IPv4 Multicast Option)       * 0x24 (IPv4 SD Endpoint Option)       * 0x01 (Configuration Option) |
| Reference | DS\_SD\_0071, DS\_SD\_0072, DS\_SD\_0073, DS\_SD\_0103 |
| Notes |  |

### Service Discovery Communication Behavior

#### [SOMEIPGEN\_SD\_BEHAVIOR\_01] SD Message shall be sent over UDP

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a SOME/IP-SD Message is sent over UDP. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | **PORT\_NUMBER**: (Uint16) Number of the UDP port used for SOME/IP Service Discovery. |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends multicast SOME/IP-SD message. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received by port **PORT\_NUMBER** before **TimeoutStartupSD.** |
| Reference | DS\_SD\_0003 |
| Notes | The test case expects to check the first message a SOME/IP server will send out, which should be actual a SOME/IP-SD message. Therefore appropriate values of a valid SOME/IP-SD message header are assumed for verification. |

#### [SOMEIPGEN\_SD\_BEHAVIOR\_02] Session ID is incremented after each message

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that SOME/IP-SD uses incrementation of Session ID. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends multicast SOME/IP-SD message. 3. TESTER: Verify received SOME/IP-SD message. 4. Repeat preceding steps 2 to 3 until a total number of 0xFFFE (65534) SOME/IP-SD messages are processed. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Session ID is equal to 0x0001. 2. Each SOME/IP-SD message is received before its respective **TimeoutOfferService** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Session ID is incremented by one in compare to value of the preceding message. |
| Reference | DS\_SD\_0011, DS\_SD\_0013 |
| Notes | The test case expects to check the first message a SOME/IP server will send out, which should be actual a SOME/IP-SD message. Therefore appropriate values of a valid SOME/IP-SD message header are assumed for verification. |

#### [SOMEIPGEN\_SD\_BEHAVIOR\_03] Session ID is one after wrapping

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester observes the session id incrementation. After wrap-around the Session ID has to start from 1. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service.  In order to save test time, **setSessionID** may be called with sessionID close to wrap-around. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends multicast SOME/IP-SD message. 3. TESTER: Verify received SOME/IP-SD message. 4. Repeat preceding steps 2 to 3 until a total number of 0xFFFE (65534) SOME/IP-SD messages are processed. 5. DUT: Sends multicast SOME/IP-SD message. 6. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Session ID is equal to 0x0001. 2. Each SOME/IP-SD message is received before its respective **TimeoutOfferService** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Session ID is incremented by one in compare to value of the preceding message. 3. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Session ID is equal to 0x0001. |
| Reference | DS\_SD\_0012 |
| Notes |  |

#### [SOMEIPGEN\_SD\_BEHAVIOR\_04] Reboot Flag is set to one

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a SOME/IP-SD message provides a bit flag which is used for reboot detection of SOME/IP Service Discovery instances. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends multicast SOME/IP-SD message. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Message ID is equal to 0xFFFF8100 (Service Discovery).    2. Reboot Flag is set to 1. |
| Reference | DS\_SD\_0021, DS\_SD\_0023 |
| Notes |  |

#### [SOMEIPGEN\_SD\_BEHAVIOR\_05] Reboot Flag after Session ID wraps around 1st time

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the Reboot Flag of a SOME/IP-SD message toggles its state when value of Session ID is wrapping around for the first time. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service.  In order to save test time, **setSessionID** may be called with sessionID close to wrap-around. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends multicast SOME/IP-SD message. 3. TESTER: Verify received SOME/IP-SD message. 4. Repeat preceding steps 2 to 3 until a total number of 0xFFFE (65534) SOME/IP-SD messages are processed. 5. DUT: Sends multicast SOME/IP-SD message. 6. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Session ID is equal to 0x0001.    3. Reboot Flag is set to 1. 2. Each SOME/IP-SD message is received before its respective **TimeoutOfferService** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Reboot Flag is set to 1. 3. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Session ID is equal to 0x0001.    3. Reboot Flag is set to 0. |
| Reference | DS\_SD\_0021, DS\_SD\_0023, DS\_SD\_0189 |
| Notes |  |

#### [SOMEIPGEN\_SD\_BEHAVIOR\_06] Reboot Flag after Session ID wraps around 2nd time

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the Reboot Flag of a SOME/IP-SD message does not change its state when value of Session ID is wrapping around for the second time. |
| Preconditions | DUT shall provide or consume at least one SOME/IP service.  In order to save test time, **setSessionID** may be called with sessionID close to wrap-around. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | None |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends multicast SOME/IP-SD message. 3. TESTER: Verify received SOME/IP-SD message. 4. Repeat preceding steps 2 to 3 until a total number of 0xFFFE (65534) SOME/IP-SD messages are processed. 5. Repeat again preceding steps 2 to 3 until an additional number of 0xFFFE (65534) SOME/IP-SD messages are processed. 6. DUT: Sends multicast SOME/IP-SD message. 7. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before**TimeoutStartupSD** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Session ID is equal to 0x0001.    3. Reboot Flag is set to 1. 2. Each SOME/IP-SD message is received before its respective **TimeoutOfferService** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Reboot Flag is set to 1. 3. Each SOME/IP-SD message is received before its respective **TimeoutOfferService** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Reboot Flag is set to 0. 4. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Message ID is equal to 0xFFFF8100.    2. Session ID is equal to 0x0001.    3. Reboot Flag is set to 0. |
| Reference | DS\_SD\_0021, DS\_SD\_0023, DS\_SD\_0189 |
| Notes |  |

## SOME/IP Server

### RPC Protocol Specification

#### [SOMEIPSRV\_RPC\_01] Request is answered with a Response copy Message ID, Request ID and the Interface Version.

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expect that a received message of type REQUEST is answered with a message of type RESPONSE and uses the same message ID, Request ID and interface Version as the one used by the Request where to answer. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **PORT\_NUMBER**: (Uint16) Number of the UDP Port used by the simulated SOME/IP client. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request message from source Port **PORT\_NUMBER** for method **echoUINT8** with parameter **requestValue** set to **UINT8\_VALUE**. 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP Response message is received at destination port **PORT\_NUMBER** before **TimeoutResponse** andcontains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**).    3. Request ID is equal to the one of preceding Request message.    4. Interface Version is equal to **ETS\_MajorVersion\_0**.    5. Message Type is equal to 0x80 (RESPONSE). |
| Reference | DS\_SP\_0024, DS\_SP\_0025, DS\_SP\_0081, DS\_SD\_0234, DS\_SP\_0084, DS\_SP\_0011, DS\_SP\_0085 |
| Notes |  |

#### [SOMEIPSRV\_RPC\_02] Proper order of arguments in payload for Response

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expects that the DUT Response have the proper order of arguments in payload. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UINT16\_VALUE**: 16-bit unsigned integer used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request message for method **echoBitfields** with the following parameters:    * **value1** is set to **UINT8\_VALUE**.    * **value2** is set to **UINT16\_VALUE**.    * **value3** is set to **UINT32\_VALUE**. 4. DUT: Sends the corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within TimeoutOfferService and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0041 (**echoBitfields**).    3. Result Code is equal to 0x00 (E\_OK).    4. Payload provides the following data in the prescribed order:       * UINT8 value equal to **UINT8\_VALUE**.       * UINT16 value equal to **UINT16\_VALUE**.       * UINT32 value equal to **UINT32\_VALUE**. |
| Reference | DS\_SP\_0085 |
| Notes |  |

#### [SOMEIPSRV\_RPC\_03] Return Code for no error

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expects that the response message of the DUT returns the code for no error when the request was processed without any errors. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request message for method **echoUINT8** with parameter **requestValue** set to **UINT8\_VALUE**. 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0.**    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. Received SOME/IP Response message within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0.**    2. Return Code is equal to 0x00 (E\_OK). |
| Reference | DS\_SP\_0029, DS\_SP\_0030, DS\_SP\_0031, DS\_SP\_0102, DS\_SP\_0106 |
| Notes |  |

#### [SOMEIPSRV\_RPC\_04] Return Code for wrong Protocol Version error

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expects that the response message from the DUT returns an appropriate error code when a wrong SOME/IP Protocol Version arrives in a request. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **WRONG\_PROTOCOL\_VERSION** : (Uint8) SOME/IP Protocol Version number which is not supported by DUT |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends modified SOME/IP Request message for method **echoUINT8** with parameter **requestValue** set to **UINT8\_VALUE**, where SOME/IP Protocol Version is changed to **WRONG\_PROTOCOL\_VERSION**. 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0.**    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. Received SOME/IP Response message within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Return Code is equal to 0x07 (E\_WRONG\_PROTOCOL\_VERSION). |
| Reference | DS\_SP\_0031, DS\_SP\_0032, DS\_SP\_0106, DS\_SP\_0107 |
| Notes |  |

#### [SOMEIPSRV\_RPC\_05] Return Code for unknown Service error

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expects that a response message from the DUT returns an appropriate error code when an unknown Service ID is addressed in a request. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UNKNOWN\_SERVICE\_ID** : (Uint16) Service ID which is unknown to DUT |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends modified SOME/IP Request message for method **echoUINT8** with parameters **requestValue** set to **UINT8\_VALUE**, where Service ID is changed to **UNKNOWN\_SERVICE\_ID**. 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. Received SOME/IP Response message within **TimeoutResponse** and contains:    1. Service ID is equal to **UNKNOWN\_SERVICE\_ID**.    2. Return Code is equal to 0x02 (E\_UNKNOWN\_SERVICE). |
| Reference | DS\_SP\_0031, DS\_SP\_0032, DS\_SP\_0106, DS\_SP\_0107 |
| Notes |  |

#### [SOMEIPSRV\_RPC\_06] Return Code for wrong Interface Version error

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expects that a response message from the DUT returns an appropriate error code when a wrong interface version number is addressed in a request. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **WRONG\_INTERFACE\_VERSION** : (Uint8) Major version number of the respective SOME/IP service which is not supported by DUT |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends modified SOME/IP Request message for method **echoUINT8** with parameter **requestValue** set to **UINT8\_VALUE**, where Interface Version is changed to **WRONG\_INTERFACE\_VERSION**. 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0.**    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. Received SOME/IP Response message within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Return Code is equal to 0x08 (E\_WRONG\_INTERFACE \_VERSION). |
| Reference | DS\_SP\_0031, DS\_SP\_0032, DS\_SP\_0106, DS\_SP\_0107 |
| Notes |  |

#### [SOMEIPSRV\_RPC\_07] Return Code for unknown Method error

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expects that a response message from the DUT returns an appropriate error code when an unknown Method ID is addressed in a request. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UNKNOWN\_METHOD\_ID** : (Uint8) Method ID which is unknown to the respective SOME/IP service |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends modified SOME/IP Request message for method **echoUINT8** with parameter **requestValue** set to **UINT8\_VALUE**, where Method ID is changed to **UNKNOWN\_METHOD\_ID**. 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0.**    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. Received SOME/IP Response message within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Return Code is equal to 0x03 (E\_UNKNOWN\_METHOD). |
| Reference | DS\_SP\_0031, DS\_SP\_0032, DS\_SP\_0106, DS\_SP\_0107 |
| Notes |  |

#### [SOMEIPSRV\_RPC\_08] Return Code for wrong Message Type error

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expects that a response message from the DUT returns an appropriate error code when a wrong Message Type arrives in a request. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **WRONG\_MESSAGE\_TYPE** : (Uint8) Message Type different than 0x00 (REQUEST) (0x01 (Request-No-Return) recommended) |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends modified SOME/IP Request message for method **echoUINT8** with parameter **requestValue** set to **UINT8\_VALUE**, where Message Type is changed to **WRONG\_MESSAGE\_TYPE**. 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0.**    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. Received SOME/IP Response message within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Return Code is equal to 0x0A (E\_WRONG\_MESSAGE\_TYPE). |
| Reference | DS\_SP\_0031, DS\_SP\_0032, DS\_SP\_0106, DS\_SP\_0107 |
| Notes |  |

#### [SOMEIPSRV\_RPC\_09] Return Code for malformed message error

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expects that a response message from the DUT returns an appropriate error code when the request’s payload data is not parsable. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_ARRAY**: Sequence of 8-bit unsigned integer elements used as arbitrary payload for the test. Must be fewer elements as required by the ETS data type TestArrayUINT8Static. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request message for method **echoStaticUINT8Array** with parameter **requestValue** set to **UINT8\_ARRAY**. 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0.**    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. Received SOME/IP Response message within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Return Code is equal to 0x09 (E\_MALFORMED\_MESSAGE). |
| Reference | DS\_SP\_0031, DS\_SP\_0032, DS\_SP\_0106, DS\_SP\_0107 |
| Notes |  |

#### [SOMEIPSRV\_RPC\_10] Return Code for Length equals 0

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks handling of request message with an invalid message length of zero bytes. Minimum size of any SOME/IP message is 8 bytes. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends modified SOME/IP Request message for method **echoUINT8**  with parameter **requestValue** set to **UINT8\_VALUE**, where Length is changed to 0x00000000 (0 bytes). 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0.**    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. Received SOME/IP Response message within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Return Code is equal to 0x09 (E\_MALFORMED\_MESSAGE). |
| Reference | DS\_SP\_0031, DS\_SP\_0032, DS\_SP\_0106, DS\_SP\_0107 |
| Notes |  |

#### [SOMEIPSRV\_RPC\_11] Return Code for Length smaller than 8

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks handling of request message with an invalid message length smaller than 8 bytes. Minimum size of any SOME/IP message is 8 bytes. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends modified SOME/IP Request message for method **echoUINT8**  with parameter **requestValue** set to **UINT8\_VALUE**, where Length is changed to 0x00000007 (7 bytes). 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0.**    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. Received SOME/IP Response message within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Return Code is equal to 0x09 (E\_MALFORMED\_MESSAGE). |
| Reference | DS\_SP\_0031, DS\_SP\_0032, DS\_SP\_0106, DS\_SP\_0107 |
| Notes |  |

#### [SOMEIPSRV\_RPC\_12] Request ID may be reused if Response arrived

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that request message with a reused Request ID is processed and answered without any error. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request message for method **echoUINT8**  with parameter **requestValue** set to **UINT8\_VALUE.** 4. DUT: Sends response message. 5. TESTER: Verify received response message. 6. TESTER: Sends SOME/IP Request message for method **echoUINT8** with same Session ID value as in the preceding request message 7. DUT: Sends response message. 8. TESTER: Verify received response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0.**    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. 1st Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Request ID is equal to the one of preceding Request message.    3. Return Code is equal to 0x00 (E\_OK). 3. 2nd Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Request ID is equal to the one of preceding Request message.    3. Return Code is equal to 0x00 (E\_OK). |
| Reference | DS\_SP\_0012 |
| Notes |  |

#### [SOMEIPSRV\_RPC\_13] Fire & Forget Request is not answered with a Response

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a received message of type REQUEST\_NO\_RETURN is not answered with any response. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **ARRAY\_LENGTH**: Number of elements contained by the array.  **UINT8\_ARRAY**: Sequence of UINT8 elements used as arbitrary payload for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request-No-Return message for method **setUINT8Array**  with the following parameters:    * **dataLength** is set to **ARRAY\_LENGTH**.    * **data** is set to **UINT8\_ARRAY.** 4. TESTER: Observes further DUT communication until a period of **TimeoutResponse** has elapsed. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. No SOME/IP messages were observed which contain:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x50C (**setUINT8Array**).    3. Message Type is equal to 0x80 (RESPONSE). |
| Reference | DS\_SP\_0026, DS\_SP\_0088, DS\_SP\_0089 |
| Notes |  |

#### [SOMEIPSRV\_RPC\_14] Fire & Forget Request shall return no error

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expects that a received message of type REQUEST\_NO\_RETURN is not answered from the DUT with any response, even when it’s processing results in an error. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request-No-Return message for method **setUINT8Array** without any payload**.** 4. TESTER: Observes further DUT communication until a period of **TimeoutResponse** has elapsed. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0.**    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. No SOME/IP messages were observed which contain:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x50C (**setUINT8Array**).    3. Message Type is equal to 0x80 (RESPONSE) or 0x81 (ERROR).    4. Return Code is different than 0x00 (E\_OK). |
| Reference | DS\_SP\_0090, DS\_SP\_0104 |
| Notes |  |

#### [SOMEIPSRV\_RPC\_15] Getter returns value of a Field

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that response message of a request, representing the getter of a Field, returns the value of that field in payload. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request message for getter of field **TestFieldUINT8.** 4. DUT: Sends response message. 5. TESTER: Verify received response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0.**    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. Received SOME/IP Response message within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Return Code is equal to 0x00 (E\_OK).    3. Payload carries the expected field value. |
| Reference | DS\_SP\_0096, DS\_SP\_0098 |
| Notes |  |

#### [SOMEIPSRV\_RPC\_16] Setter changes value of a Field

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a setter changes value of a Field. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary field value. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request message for setter of field **TestFieldUINT8** with payload set to **UINT8\_VALUE**. 4. DUT: Sends the corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. 6. TESTER: Sends SOME/IP Request message for getter of field **TestFieldUINT8** of ETS instance **ETS\_InstanceID\_0**. 7. DUT: Sends the corresponding SOME/IP Response message. 8. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0027 (setter of **TestFieldUINT8**).    3. Result Code is equal to 0x00 (E\_OK). 3. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0026 (getter of **TestFieldUINT8**).    3. Result Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **UINT8\_VALUE**. |
| Reference | DS\_SP\_0096, DS\_SP\_0099 |
| Notes |  |

#### [SOMEIPSRV\_RPC\_17] Setter returns the value that was set to a Field

|  |  |
| --- | --- |
| Item | Description |
| Purpose | For values which lies within the Field’s accepted limits, the DUT is expected to send back the same value in the setter’s response which was received in the setter’s corresponding request before. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary field value. Must be in range 0x00 – 0x7F. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request message for setter of field **TestFieldUINT8** with payload set to **UINT8\_VALUE**. 4. DUT: Sends the corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0027 (setter of **TestFieldUINT8**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **UINT8\_VALUE**. |
| Reference | DS\_SP\_0096, DS\_SP\_0099 |
| Notes |  |

#### [SOMEIPSRV\_RPC\_19] Server sends Error msg instead of Response and copy header field values from Request

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expects that the DUT sends an Error msg instead of a Response and copy header field values from Request. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload. Must be in range 0x80 – 0xFF in order to trigger out of range error.  **UTF8\_STRING\_VALUE:** Arbitrary non-empty character string with UTF8 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 3 byte values: 0xEF, 0xBB and 0xBF. Must be terminated with character “\0”. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request message for method **echoUINT8Conditional** with the following parameters:  * requestValue is set to **UINT8\_VALUE.**  1. DUT: Sends SOME/IP Error message. 2. TESTER: Verify received SOME/IP Error message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. SOME/IP Error message is received at destination port **PORT\_NUMBER** within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x518 (**echoUINT8Conditional**).    3. Request ID is equal to the one of preceeding Request message.    4. Interface Version is equal to the one of preceeding Request message.    5. Message Type is equal to 0x81 (ERROR).    6. Return code is equal 0x20 (referred to ETS specification). |
| Reference | DS\_SP\_0024, DS\_SP\_0025, DS\_SP\_0103 DS\_SP\_0029, DS\_SP\_0030, DS\_SP\_0102, DS\_SP\_0105 |
| Notes |  |

#### [SOMEIPSRV\_RPC\_20] Server sends Error msg with specific payload layout

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Server forwards application error and sends Error msg with specific payload layout. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING\_VALUE:** Arbitrary non-empty character string with UTF8 encoding used for the test. Must start with a Byte Order Mark (BOM) consisting of the following 3 byte values: 0xEF, 0xBB and 0xBF. Must be terminated with character “\0”.  **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload. Must be in range 0x80 – 0xFF in order to trigger out of range error. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request message for setter of **TestFieldUTF8DynamicReliable** with payload set to **UTF8\_STRING\_VALUE.** 4. DUT: Sends the corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. 6. TESTER: Sends SOME/IP Request message for method **echoUINT8Conditional** with the following parameters:    * **requestValue** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Error message. 8. TESTER: Verify received SOME/IP Error message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x561(**TestFieldUTF8DynamicReliable**)    3. Message Type is equal to 0x80 (RESPONSE).    4. Return Code is equal to 0x00 (E\_OK). 3. SOME/IP Error message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x518 (**echoUINT8Conditional**).    3. Message Type is equal to 0x81 (ERROR).    4. Return Code is equal to 0x20 (referred to ETS specification).    5. Payload provides an UTF8 string equal to **UTF8\_STRING\_VALUE**. |
| Reference | DS\_SP\_0109, DS\_SP\_0112, DS\_SP\_0111 |
| Notes |  |

### Transport Protocol

#### [SOMEIPSRV\_TP\_01] UDP Binding really use UDP

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that an UDP binding is used for the communication, that some/IP really use the UDP protocol for an SOME/IP-SD message. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request message for method **echoUINT8** with parameter **value** set to **UINT8\_VALUE**. 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP Response message is received via UDP within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**). |
| Reference | DS\_SP\_0115, DS\_SP\_0122, DS\_SD\_0127 |
| Notes |  |

#### [SOMEIPSRV\_TP\_02] UDP Port maps Request to correct Service instance

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to forward an incoming Request over UDP to the correct service instance, when the DUT runs different instances of the same service. |
| Preconditions | DUT is running and offering the Enhanced Testability Service in two instances which only differs by its instance Id.  Method **serviceActivate** with instanceKey set to 0x02 (alternative instance) may be called for this test case. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SERVICE\_DOWN:** (Uint8)Duration in seconds where the service shall be deactivated. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **suspendInterface** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**.   Destination Port is set to **ETS\_PortUDP\_0**.   1. DUT: Sends multicast SOME/IP-SD message carrying a Stop Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request message for method **suspendInterface** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**.   Destination Port is set to **ETS\_PortUDP\_1**.   1. DUT: Sends multicast SOME/IP-SD message carrying a Stop Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStopOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Stop Offer Service Entry).    2. Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Service instance is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to 0x000000. 2. SOME/IP-SD message is received before **TimeoutStopOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Stop Offer Service Entry).    2. Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Service instance is equal to **ETS\_InstanceID\_1.**    4. Entry: TTL is equal to 0x000000. |
| Reference | DS\_SP\_0121, DS\_SP\_0143 |
| Notes |  |

#### [SOMEIPSRV\_TP\_03] Receive Multiple SOME/IP messages in one UDP package

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The tester Sending two SOMEIP messages in one UDP package. DUT hast to reply on both SOMEIP messages and send the correct response. |
| Preconditions | Please check section Preconditions |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE\_1**: Unsigned integer value used as arbitrary payload for the test.  **UINT8\_VALUE\_2**: Unsigned integer value used as arbitrary payload for the test. Shall be different than **UINT8\_VALUE\_1.** |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends one UDP datagram carrying the following SOME/IP messages in the prescribed order:    * Request for ETS method **echoUINT8** with parameter **value** set to **UINT8\_VALUE\_1**.    * Request for ETS method **echoUINT8** with parameter **value** set to **UINT8\_VALUE\_2**. 4. DUT: Sends the corresponding SOME/IP Responses messages. 5. TESTER: Verify received SOME/IP Responses message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. Two SOME/IP Response messages are received within **TimeoutResponse** and contain:    1. Both: Service ID is equal to **ETS\_ServiceID\_0**.    2. Both: Method ID is equal to 0x0008 (**echoUINT8**).    3. Both: Return Code is equal to 0x00 (E\_OK).    4. 1st message: Payload provides an UINT8 value equal to **UINT8\_VALUE\_1**.    5. 2nd message: Payload provides an UINT8 value equal to **UINT8\_VALUE\_2**. |
| Reference | DS\_SP\_0118, DS\_SP\_0124, DS\_SP\_0125 |
| Notes |  |

#### [SOMEIPSRV\_TP\_04] Notification UDP datagram via Unicast

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Check if the DUT is able to send unicast Some/IP Notification messages over UDP. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message with an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with an Subscribe Service Entry for Eventgroup ID 0x0005. 4. DUT: Sends unicast SOME/IP-SD message carrying corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. DUT: Sends unicast SOME/IP Notification message for field **TestFieldUINT8**. 7. TESTER: Verify received SOME/IP Notification message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0005.    4. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received via unicast UDP within **TimeoutInitialEvents** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8006 (**TestFieldUINT8**).    3. Message Type is equal to 0x02 (NOTIFICATION). |
| Reference | DS\_SP\_0127 |
| Notes |  |

#### [SOMEIPSRV\_TP\_05] Notification UDP datagram via Multicast

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Check if the DUT is able to send multicast Some/IP Notification messages over UDP. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message with an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with an Subscribe Service Entry for Eventgroup ID 0x0006. 4. DUT: Sends unicast SOME/IP-SD message carrying corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for method **triggerEventUINT8Multicast** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds)    * **duration** is set to 0x00000001 (1 second).    * **debounceTime** is set to 0x00000001 (1 second). 7. DUT: Sends multicast SOME/IP Notification message for ETS event **TestEventUINT8Multicast**. 8. TESTER: Verify received SOME/IP Notification message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0006.    4. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received over UDP within **duration** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x800B (**TestEventUINT8Multicast**).    3. Message Type is equal to 0x02 (NOTIFICATION). |
| Reference | DS\_SP\_0127 |
| Notes |  |

#### [SOMEIPSRV\_TP\_06] TCP Binding really use TCP

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that when an TCP binding is used for the communication, that some/IP really use the TCP protocol for an SOME/IP message |
| Preconditions | The ECU shall provide at least SOME/IP one service |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Try to establish TCP reconnect by sending SYN. 4. DUT: Accepts TCP connection by sending SYN + ACK. 5. TESTER: Accepts TCP connection by sending ACK. 6. TESTER: Sends SOME/IP Request message for method **echoUINT8RELIABLE** with parameter **value** set to **UINT8\_VALUE**. 7. DUT: Sends corresponding SOME/IP Response message. 8. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. TCP SYN+ACK is received within **TimeoutCloseTcpConnection**. 3. SOME/IP Response message is received via TCP within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x000A (**echoUINT8RELIABLE**). |
| Reference | DS\_SP\_0115, DS\_SD\_0127 |
| Notes |  |

#### [SOMEIPSRV\_TP\_07] TCP Binding transports message with a length larger than 1400

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is sending a SOME/IP messages over TCP Binding transports Message with a length larger than 1400 |
| Preconditions | The ECU shall provide at least SOME/IP one service |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE\_ARRAY**: Unsigned integer array value used as arbitrary payload for the test.  **ARRAY\_LENGTH\_1:** Array size must be larger than 1400 byte. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message with an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Try to establish TCP reconnect by sending SYN. 4. DUT: Accepts TCP connection by sending SYN + ACK. 5. TESTER: Accepts TCP connection by sending ACK. 6. TESTER: Sends unicast SOME/IP-SD message with an Subscribe Service Entry for Eventgroup ID 0x0002. 7. DUT: Sends unicast SOME/IP-SD message with an Subscribe ACK Entry for the ETS. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends SOME/IP Request message for method **echoUINT8ArrayReliable** with the following parameters:    * **dataLength** is set to **ARRAY\_LENGTH\_1**.    * **data** is set to **UINT8\_VALUE\_ARRAY.** 10. DUT: Sends corresponding SOME/IP Response message. 11. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. TCP SYN+ACK is received within **TimeoutCloseTcpConnection**. 3. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is not equal to 0x0000. 4. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x504 (**echoUINT8ArrayReliable**).    3. Both: Return Code is equal to 0x00 (E\_OK).    4. Payload provides UINT8 array equal to **UINT8\_VALUE\_ARRAY.** |
| Reference | DS\_SP\_0035, DS\_SP\_0128 |
| Notes |  |

#### [SOMEIPSRV\_TP\_08] TCP Port maps Request to correct Service instance

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to forward an incoming Request over TCP to the correct service instance, when the DUT runs different instances of the same service. |
| Preconditions | DUT is running and offering the Enhanced Testability Service in two instances which only differs by its instance ID version.  Method **serviceActivate** with instanceKey set to 0x02 (alternative instance) may be called for this test case. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SERVICE\_DOWN:** (Uint8)Duration in seconds where the service shall be deactivated. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **suspendInterfaceReliable** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**.   Destination Port is set to **ETS\_PortTCP\_0.**   1. DUT: Sends multicast SOME/IP-SD message carrying a Stop Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request message for method **suspendInterfaceReliable** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**.   Destination Port is set to **ETS\_PortTCP\_1.**   1. DUT: Sends multicast SOME/IP-SD message carrying a Stop Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStopOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Stop Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: TTL is equal to 0x000000 2. SOME/IP-SD message is received before **TimeoutStopOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Stop Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_1**.    4. Entry: TTL is equal to 0x000000 |
| Reference | DS\_SP\_0121, DS\_SP\_0143 |
| Notes | Method suspend interface is used as Request by the test, so the correct mapping could be easily verified by checking the corresponding stop offer service of the DUT. The stop offer service provides the information which service instance is stopped. |

#### [SOMEIPSRV\_TP\_09] Receive multiple SOME/IP messages in one TCP segment

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The tester sending two SOMEIP messages in one TCP segment. DUT has to reply to all two SOMEIP messages and send the correct response.  Check the handling when more SOMEIP messages are in one transport protocol frame. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE\_1**: Unsigned integer value used as arbitrary payload for the test.  **UINT8\_VALUE\_2**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends one TCP segment carrying the following SOME/IP messages in the prescribed order:    * Request for ETS method **echoUINT8RELIABLE** with parameter **value** set to **UINT8\_VALUE\_1**.    * Request for ETS method **echoUINT8RELIABLE** with parameter **value** set to **UINT8\_VALUE\_2**. 4. DUT: Sends the corresponding SOME/IP Responses messages. 5. TESTER: Verify received SOME/IP Responses message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. Two SOME/IP Response messages are received within **TimeoutResponse** and contain:    1. Both: Service ID is equal to **ETS\_ServiceID\_0**.    2. Both: Method ID is equal to 0x000A (**echoUINT8RELIABLE**).    3. Both: Return Code is equal to 0x00 (E\_OK).    4. 1st message: Payload provides an UINT8 value equal to **UINT8\_VALUE\_1**.    5. 2nd message: Payload provides an UINT8 value equal to **UINT8\_VALUE\_2**. |
| Reference | DS\_SP\_0118 |
| Notes |  |

#### [SOMEIPSRV\_TP\_10] Server don’t close TCP connection when service is deavtivated

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Server don’t close TCP connection when service is deactivated. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SERVICE\_DOWN:** Duration in seconds where the service shall be deactivated.  **PORT\_NUMBER:** (Uint16) Number of TCP related port use for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Establish TCP connection with local port **PORT\_NUMBER.** 4. TESTER: Sends unicast SOME/IP-SD message with an Subscribe Service Entry for Eventgroup ID 0x0002. 5. DUT: Sends unicast SOME/IP-SD message carrying corresponding Subscribe Eventgroup Acknowledgement Entry. 6. TESTER: Verify received SOME/IP-SD message. 7. TESTER: Sends SOME/IP Request-No-Return message for method **suspendInterface** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to **T\_SERVICE\_DOWN**. 8. TESTER: Observes further DUT communication on local port **PORT\_NUMBER** for a period of **T\_SERVICE\_DOWN**. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Eventgroup ID is equal to 0x0002.    3. Entry: TTL is not equal to 0x0000. 3. No attempts to close TCP connection were observed (FIN not received) |
| Reference | DS\_SP\_0138 |
| Notes |  |

#### [SOMEIPSRV\_TP\_11] Different instances of same Service use different Ports for UDP bindings

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to simultaneously host two instances of same service where each of them is addressable by its respective Transport Protocol related port number. |
| Preconditions | DUT is running and offering the Enhanced Testability Service in two instances: **ETS\_InstanceID\_0** and **ETS\_InstanceID\_1**.  Method **serviceActivate** with instanceKey set to 0x02 (alternative instance) may be called for this test case. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying Offer Service Entries. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request message for method **echoUINT8** with the following parameters:    * **value** is set to **UINT8\_VALUE**.   Destination Port is set to **ETS\_PortUDP\_0**.   1. DUT: Sends corresponding SOME/IP Response message. 2. TESTER: Verify received SOME/IP Response message. 3. TESTER: Sends SOME/IP Request message for method **echoUINT8** with the following parameters:    * **value** is set to **UINT8\_VALUE**.   Destination Port is set to **ETS\_PortUDP\_1**.   1. DUT: Sends corresponding SOME/IP Response message. 2. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides two entries of type 0x01 (Offer Service Entry).    2. Both Entries: Service ID is equal to **ETS\_ServiceID\_0.**    3. One Entry: Service instance is equal to **ETS\_InstanceID\_0.**    4. One Entry: References IPV4 Endpoint Option for **ETS\_PortUDP\_0.**    5. Other Entry: Service instance is equal to **ETS\_InstanceID\_1.**    6. Other Entry: References IPV4 Endpoint Option for **ETS\_PortUDP\_1.**    7. Both entries: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP Response message is received within **TimeoutResponse** which contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **UINT8\_VALUE**.    5. Source Port is equal to **ETS\_PortUDP\_0.** 3. SOME/IP Response message is received within **TimeoutResponse** which contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008(**echoUINT8**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **UINT8\_VALUE.**    5. Source Port is equal to **ETS\_PortUDP\_1.** |
| Reference | DS\_SP\_0142, DS\_SP\_0143 |
| Notes |  |

#### [SOMEIPSRV\_TP\_12] UDP and TCP in one Service

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to provide UDP and TCP binding in one Service. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 address of the simulated client used for the test.  **PORT\_UDP**: (Uint16) Number of the client’s UDP port used for the test.  **PORT\_TCP**: (Uint16) Number of the client’s TCP port used for the test.  **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds [sec]. Must not be zero. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0002 with TTL set to **T\_SUBSCRIPTION**. The following options are referenced:  * IPv4 Endpoint Option for **IP\_ADDRESS** and **PORT\_UDP**. * IPv4 Endpoint Option for **IP\_ADDRESS** and **PORT\_TCP**.  1. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request-No-Return message for method **triggerEventUINT8** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds)    * **duration** is set to **T\_SUBSCRIPTION**.    * **debounceTime** is set to 0x00000001 (1 second). 4. DUT: Sends Unicast SOME/IP Notification messages for **TestEventUINT8**. 5. TESTER: Verify received SOME/IP message. 6. TESTER: Sends SOME/IP Request-No-Return message for method **triggerEventUINT8RELIABLE** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds)    * **duration** is set to **T\_SUBSCRIPTION**.    * **debounceTime** is set to 0x00000001 (1 second). 7. DUT: Sends Unicast SOME/IP Notification messages for **TestEventUINT8RELIABLE**. 8. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Eventgroup ID is equal to 0x0002.    3. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    4. Entry: TTL is equal to **T\_SUBSCRIPTION**. 3. SOME/IP Notification message is received via UDP within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**).    3. message Type is equal to 0x02 (NOTIFICATION)    4. UDP Destination IP is equal to **IP\_ADDRESS** and Destination Port Number is equal to **PORT\_UDP**. 4. SOME/IP Notification message is received via TCP within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8003 (**TestEventUINT8RELIABLE**).    3. message Type is equal to 0x02 (NOTIFICATION)    4. TCP Destination IP is equal to **IP\_ADDRESS** and Destination Port Number is equal to **PORT\_TCP**. |
| Reference | DS\_SP\_0116, DS\_SD\_0232 |
| Notes |  |

#### [SOMEIPSRV\_TP\_13] Magic Cookie Message Format Server

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks the Format of the SOME/IP Magic Cookie Message |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT related settings for period threshold of sending Magic Cookies must be 100 milliseconds [msec]. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **PORT\_NUMBER**: (Uint16) Number of the TCP port where the Enhanced Testability Service is reachable.  **UTF8\_STRING:** Null terminated character string used as arbitrary payload. String length (Uint32) Length of the character string. Must be at least 30000. |
| Test Procedure | 1. TESTER: Establish TCP connection with local port **PORT\_NUMBER**. 2. TESTER: Sends SOME/IP Request message for setter of ETS field **TestFieldUTF8DynamicReliable** with the following parameters:    * UTF8 string equal to **UTF8\_STRING**. 3. DUT: Sends SOME/IP Magic Cookie message 4. TESTER: Verify received Magic Cookie message. |
| Verification | 1. SOME/IP message is received within **TimeoutResponse** and contains:    1. Message ID is equal to 0xFFFF 8000.    2. Length is equal to 0x0000 0008    3. Request ID is equal to =0xDEAD BEEF    4. Protocol Version is equal to 0x01.    5. Interface version is equal to 0x01.    6. Message Type is equal to 0x02 (NOTIFICATION).    7. Return Code is equal to 0x00 (E\_OK). |
| Reference | DS\_SP\_0144, DS\_SP\_0146 |
| Notes | Magic Cookie message is triggered by forcing the DUT to transmit a large response message over TCP. |

#### [SOMEIPSRV\_TP\_14] Magic Cookie Cyclic sending Server

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send Magic Cookie messages in a cyclic manner. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT related settings for period threshold of sending Magic Cookies must be 100 milliseconds [msec]. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **PORT\_NUMBER**: (Uint16) Number of the TCP port where the Enhanced Testability Service is reachable.  **UTF8\_STRING:** Null terminated character string used as arbitrary payload.  **N\_REPETITIONS**: (Uint8) Number of interactions the cyclic delay of Magic Cookie shall be measured. Must be at least 3.  **MAGIC\_COOKIE\_CYCLIC\_DELAY**: (Uint8) Interval of sending Magic Cookie messages used by the DUT. |
| Test Procedure | 1. TESTER: Establish TCP connection with local port **PORT\_NUMBER**. 2. TESTER: Sends SOME/IP Request message for setter of ETS field **TestFieldUTF8DynamicReliable** with the following parameters:    * UTF8 string equal to **UTF8\_STRING**. 3. DUT: Sends SOME/IP Magic Cookie message 4. TESTER: Verify received Magic Cookie message. 5. Repeat preceding steps 2 and 3 until a total number of **N\_REPETITIONS** are processed. 6. TESTER: Verify the measured cyclic delay of the SOME/IP Magic Cookie messages received in step 5. |
| Verification | 1. SOME/IP message is received within **TimeoutResponse** and contains:    1. Message ID is equal to 0xFFFF 8000.    2. Request ID is equal to 0xDEAD BEEF 2. Measured delay value between two consecutive SOME/IP Magic Cookie messages is equal to **MAGIC\_COOKIE\_CYCLIC\_DELAY** +/-**TimingTolerance.** |
| Reference | DS\_SP\_0148 |
| Notes |  |

#### [SOMEIPSRV\_TP\_15] SOME/IP-TP message type set to TP\_Notification

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT sends a SOME/IP message larger as 1400 bytes and the Tester checks that the received message is a SOME/IP-TP message where the message type is set to TP\_Notification. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT shall support message segmentation via SOME/IP-TP. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Lifetime of Eventgroup subscription used for the test. Must be at least 2 seconds.  **ARRAY\_LENGTH**: Size of message payload in bytes. Must be larger than 1400.  **UINT8\_ARRAY**: Array of UINT8 values used as arbitrary payload for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0005 and TTL set to **T\_SUBSCRIPTION**. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for method **setUINT8ArrayReliable** with the following parameters:    * **dataLength** is set to **ARRAY\_LENGTH**.    * **data** is set to **UINT8\_ARRAY**. 7. DUT: Sends 1st segment of SOME/IP Notification message for event **TestFieldUINT8Array**. 8. TESTER: Verify received SOME/IP-TP segment. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Eventgroup ID is equal to 0x0005.    3. Service ID is equal to **ETS\_ServiceID\_0**.    4. Entry: TTL is equal to **T\_SUBSCRIPTION**.   8. SOME/IP-TP segment is received within 1 second and contains:   * 1. Service ID is equal to **ETS\_ServiceID\_0**.   2. Method ID is equal to 0x8007 (**TestFieldUINT8Array**).   3. Message Type is equal to 0x22 (TP\_NOTIFICATION). |
|  | DS\_SP\_0028, DS\_SP\_0157 |
| Notes |  |

#### [SOMEIPSRV\_TP\_16] SOME/IP-TP message type set to TP\_Response

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT sends a SOME/IP message larger as 1400 bytes and the Tester checks that the received message is a SOME/IP-TP message where the message type is set to TP\_Response. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT shall support message segmentation via SOME/IP-TP. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **ARRAY\_LENGTH**: Size of message payload in bytes. Must be larger than 1400  **UINT8\_ARRAY**: Array of UINT8 values used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **setUINT8ArrayReliable** with the following parameters:    * **dataLength** is set to **ARRAY\_LENGTH**.    * **data** is set to **UINT8\_ARRAY**. 2. TESTER: Sends SOME/IP Request message for getter of ETS field **TestFieldUINT8Array**. 3. DUT: Sends 1st segment of corresponding SOME/IP Response message. 4. TESTER: Verify received SOME/IP-TP segment. |
| Verification | 1. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0028 (getter of **TestFieldUINT8Array**).    3. message Type is equal to 0xA0 (TP\_RESPONSE). |
| Reference | DS\_SP\_0028, DS\_SP\_0157 |
| Notes |  |

#### [SOMEIPSRV\_TP\_17] SOME/IP-TP message type set to TP\_Error

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT sends a SOME/IP message larger as 1400 bytes and the Tester checks that the received message is a SOME/IP-TP message where the message type is set to TP Error. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT shall support message segmentation via SOME/IP-TP. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRING\_LENGTH**: (Uint32) Length in number of characters. Must be larger than 1400.  **UTF8\_STRING**: Character string used as arbitrary payload for the test.  **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload. Must be in range 0x80 – 0xFF in order to trigger out of range error. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **setUTF8DYNAMICReliable** with the following parameters:    * **dataLength** is set to **STRING\_LENGTH**.    * **data** is set to **UTF8\_STRING**. 2. TESTER: Sends SOME/IP Request message for method **echoUINT8Conditional** with the following parameters:    * **requestValue** is set to **UINT8\_VALUE**. 3. DUT: Sends 1st segment of corresponding SOME/IP Error message. 4. TESTER: Verify received SOME/IP-TP segment. |
| Verification | 1. SOME/IP-TP segment is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x518 (**echoUINT8Conditional**).    3. Message Type is equal to 0xA1 (TP\_ERROR).    4. Return Code is equal to 0x20 (referred to ETS specification).    5. Payload provides an UTF8 string equal to **UTF8\_STRING\_VALUE**. |
| Reference | DS\_SP\_0028, DS\_SP\_0157 |
| Notes |  |

#### [SOMEIPSRV\_TP\_18] Different instances of same Service use different Ports for TCP bindings

|  |  |
| --- | --- |
| Item | Description |
| Purpose | When having more than one instance of a service a TCP connection per services instance is needed. |
| Preconditions | DUT is running and offering the Enhanced Testability Service in two instances which only differs by its Major version.  Method **serviceActivate** with **instanceKey** set to 0x02 (alternative instance) may be called for this test case. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying Offer Service Entries. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Opens TCP connection to the DUT at Destination port number **ETS\_PortTCP\_0.** 4. TESTER: Sends SOME/IP Request message for method **echoUINT8RELIABLE** with the following parameters:    * **value** is set to **UINT8\_VALUE**. 5. DUT: Sends corresponding SOME/IP Response message. 6. TESTER: Verify received SOME/IP Response message. 7. TESTER: Opens TCP connection to the DUT at Destination port number **ETS\_PortTCP\_1.** 8. TESTER: Sends SOME/IP Request message for method **echoUINT8RELIABLE** with the following parameters:    * **value** is set to **UINT8\_VALUE**. 9. DUT: Sends corresponding SOME/IP Response message. 10. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides two entries of type 0x01 (Offer Service Entry).    2. Both Entries: Service ID is equal to **ETS\_ServiceID\_0**    3. Both entries: TTL is equal to **ETS\_Lifetime**.    4. One Entry: References IPV4 Endpoint Option for **ETS\_PortTCP\_0.**    5. Other Entry: References IPV4 Endpoint Option for **ETS\_PortTCP\_1.** 2. SOME/IP Response message is received within **TimeoutResponse** which contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x000A (**echoUINT8RELIABLE**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **UINT8\_VALUE**. 3. SOME/IP Response message is received within **TimeoutResponse** which contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x000A(**echoUINT8RELIABLE**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **UINT8\_VALUE.** |
| Reference | DS\_SP\_0130 |
| Notes |  |

#### [SOMEIPSRV\_TP\_19] Different services can use same ports for UDP bindings

|  |  |
| --- | --- |
| Item | Description |
| Purpose | While different Services shall be able to share the same port number of the transport layer protocol used, multiple Service-Instances of the same service on one single ECU shall listen on different ports per Service-Instance. |
| Preconditions | DUT is running and offering the Enhanced Testability Service twice with different Service IDs.  Method **serviceActivate** with **instanceKey** set to 0x01 (alternative service) may be called for this test case. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying Offer Service Entries. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request message for method **echoUINT8** of service **ETS\_ServiceID\_0** with the following parameters:    * **value** is set to **UINT8\_VALUE**. 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. 6. TESTER: Sends SOME/IP Request message for method **echoUINT8** of service **ETS\_ServiceID\_1** with the following parameters:    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends corresponding SOME/IP Response message. 8. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides two entries of type 0x01 (Offer Service Entry).    2. One Entry: Service ID is equal to **ETS\_ServiceID\_0.**    3. One Entry: References IPV4 Endpoint Option for **ETS\_PortUDP\_0.**    4. Other Entry: Service ID is equal to **ETS\_ServiceID\_1.**    5. Other Entry: References IPV4 Endpoint Option for **ETS\_PortUDP\_0.**    6. Both entries: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP Response message is received within **TimeoutResponse** which contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **UINT8\_VALUE**. 3. SOME/IP Response message is received within **TimeoutResponse** which contains:    1. Service ID is equal to **ETS\_ServiceID\_1**.    2. Method ID is equal to 0x0008(**echoUINT8**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **UINT8\_VALUE.** |
| Reference | DS\_SP\_0143 |
| Notes |  |

#### [SOMEIPSRV\_TP\_20] Different services can use same ports for TCP bindings

|  |  |
| --- | --- |
| Item | Description |
| Purpose | While different Services shall be able to share the same port number of the transport layer protocol used, multiple Service-Instances of the same service on one single ECU shall listen on different ports per Service-Instance. |
| Preconditions | DUT is running and offering the Enhanced Testability Service twice with different Service IDs.  Method **serviceActivate** with **instanceKey** set to 0x01 (alternative service) may be called for this test case. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying Offer Service Entries. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request message for method **echoUINT8RELIABLE** of service **ETS\_ServiceID\_0** with the following parameters:    * **value** is set to **UINT8\_VALUE**. 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. 6. TESTER: Sends SOME/IP Request message for method **echoUINT8RELIABLE** of service **ETS\_ServiceID\_1** with the following parameters:    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends corresponding SOME/IP Response message. 8. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides two entries of type 0x01 (Offer Service Entry).    2. One Entry: Service ID is equal to **ETS\_ServiceID\_0.**    3. One Entry: References IPV4 Endpoint Option for **ETS\_PortTCP\_0.**    4. Other Entry: Service ID is equal to **ETS\_ServiceID\_1.**    5. Other Entry: References IPV4 Endpoint Option for **ETS\_PortTCP\_0.**    6. Both entries: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP Response message is received within **TimeoutResponse** which contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x000A (**echoUINT8RELIABLE**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **UINT8\_VALUE**. 3. SOME/IP Response message is received within **TimeoutResponse** which contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x000A(**echoUINT8RELIABLE**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **UINT8\_VALUE.** |
| Reference | DS\_SP\_0143 |
| Notes |  |

### Service Discovery Message Format

#### [SOMEIPSRV\_SD\_FORMAT\_01] Offer Service Entry

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send a well-formed Offer Service Entry which conforms to the on-wire format specified for SOME/IP-SD related Service Entries. Value of each entry field will be checked.  Respective options for the ETS specific TCP- and UDP binding are expected to be referenced in the entry. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying a Service Entry of type 0x01. 2. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides one entry.    2. Length of Entries Array is equal to 0x00000010 (16 bytes).    3. Entry: Type is equal to 0x01 (Offer Service Entry).    4. Entry: Index of 1st Option Run is equal to 0x00 (0).    5. Entry: Index of 2nd Option Run is equal to 0x00 (0).    6. Entry: Number of Option 1 is equal to 0x1 (1).    7. Entry: Number of Option 2 is equal to 0x0 (0).    8. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    9. Entry: Instance ID is equal to **ETS\_InstanceID\_0**.    10. Entry: Major Version is equal to **ETS\_MajorVersion\_0**.    11. Entry: TTL is equal to **ETS\_Lifetime**.    12. Entry: Minor Version is equal to **ETS\_MinorVersion\_0**. |
| Reference | DS\_ID\_0001, DS\_ID\_0002, DS\_ID\_0006, DS\_ID\_0007,DS\_SD\_0040, DS\_SD\_0041, DS\_SD\_0042, DS\_SD\_0043, DS\_SD\_0044, DS\_SD\_0045, DS\_SD\_0046, DS\_SD\_0047, DS\_SD\_0048, DS\_SD\_0049, DS\_SD\_0050, DS\_SD\_0051,DS\_SD\_0052, DS\_SD\_104, DS\_SD\_0105, DS\_SD\_0106, DS\_SD\_0107, DS\_SD\_0108, DS\_SD\_0124, |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_02] Stop Offer Service Entry

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send a well-formed Stop Offer Service Entry which conforms to the on-wire format specified for SOME/IP-SD related Service Entries. Value of each entry field will be checked.  Referenced options are not expected for this test. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for ETS method **suspendInterface** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **duration** is set to **TimeoutOfferService**. 2. DUT: Sends multicast SOME/IP-SD message carrying a Service Entry of type 0x01. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides one entry.    2. Entry: Type is equal to 0x01 (Offer Service Entry).    3. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0**.    5. Entry: Major Version is equal to **ETS\_MajorVersion\_0**.    6. Entry: Minor Version is equal to **ETS\_MinorVersion\_0**.    7. Entry: TTL is equal to 0x000000 (0 seconds). |
| Reference | DS\_SD\_0129 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_03] Subscribe Eventgroup Acknowledgement Entry.

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send a well-formed Subscribe Eventgroup Acknowledgement Entry which conforms to the on-wire format specified for SOME/IP-SD related Eventgroup Entries. Value of each entry field will be checked.  For Subscribe Eventgroup Acknowledgement Entries is expected that all fields, except the one for the entry type and the fields used for referencing options, shall provide the same values as in the Subscribe Eventgroup Entry that is being answered.  Referenced options are not expected for this test. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for the ETS. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0005. 4. DUT: Sends unicast SOME/IP-SD message carrying an Eventgroup Entry of type 0x07. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received before **TimeoutSubscribeAck** and contains:    1. Entries Array provides one entry.    2. Length of Entries Array is equal to 0x00000010 (16).    3. Entry: Type is equal to 0x07 (Subscribe Eventgroup Acknowledgement).    4. Entry: Index of 1st Option Run is equal to 0x00 (0).    5. Entry: Index of 2nd Option Run is equal to 0x0 (0).    6. Entry: Number of Option 1 is equal to 0x0 (0).    7. Entry: Number of Option 2 is equal to 0x0 (0).    8. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    9. Entry: Instance ID is equal to **ETS\_InstanceID\_0**.    10. Entry: Major Version is equal to **ETS\_MajorVersion\_0**.    11. Entry: TTL is equal to value of preceding Subscribe Eventgroup Entry.    12. Entry: All reserved bits are set to 0.    13. Entry: Initial Data Requested Flag is equal to value of preceding Subscribe Eventgroup Entry.    14. Entry: Counter is equal to value of preceding Subscribe Eventgroup Entry.    15. Entry: Eventgroup ID is equal to 0x0005. |
| Reference | DS\_ID\_0001, DS\_ID\_0002, DS\_ID\_0006, DS\_ID\_0007, DS\_ID\_0013, DS\_ID\_0014,DS\_SD\_0040, DS\_SD\_0053, DS\_SD\_0054, DS\_SD\_0055, DS\_SD\_0056, DS\_SD\_0057, DS\_SD\_0058, DS\_SD\_0059, DS\_SD\_0060, DS\_SD\_0061, DS\_SD\_0062, DS\_SD\_0063, DS\_SD\_0064, DS\_SD\_0065, DS\_SD\_0066, DS\_SD\_0067, DS, DS\_SD\_0068,DS\_SD\_104, DS\_SD\_0105, DS\_SD\_0106, DS\_SD\_0108, DS\_SD\_0137 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_04] Subscribe Eventgroup NACK

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send a well-formed Subscribe Eventgroup Negative Acknowledgement Entry which conforms to the on-wire format specified for SOME/IP-SD related Eventgroup Entries. Value of each entry field will be checked.  For Subscribe Eventgroup Negative Acknowledgement Entries is expected that all fields, except the both for entry type and TTL as well as the fields used for referencing options, shall provide the same values as in the Subscribe Eventgroup Entry that is being answered.  Referenced options are not expected for this test. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UNKNOWN\_EVENTGROUP\_ID**: Eventgroup ID which is unknown to the ETS. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message with an Offer Service Entry for the ETS. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for **UNKNOWN\_EVENTGROUP\_ID**. 4. DUT: Sends unicast SOME/IP-SD message with Eventgroup Entry of type 0x07. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck**  and contains:    1. Entries Array provides one entry.    2. Entry: Type is equal to 0x07 (Subscribe Eventgroup Acknowledgement).    3. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0**.    5. Entry: Major Version is equal to **ETS\_MajorVersion\_0**.    6. Entry: TTL is equal to 0x000000 (0).    7. Entry: Initial Data Requested Flag is equal to value of preceding Subscribe Eventgroup Entry.    8. Entry: Counter is equal to value of preceding Subscribe Eventgroup Entry.    9. Entry: Eventgroup ID is equal to **UNKNOWN\_EVENTGROUP\_ID**. |
| Reference | DS\_SD\_0140 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_05] IPv4 Endpoint Option for TCP

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks format of a TCP related IPv4 Endpoint Option. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 address where the Enhanced Testability Service is reachable.  **PORT\_NUMBER**: (Uint16) Number of the TCP port where the Enhanced Testability Service is reachable. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provides an option of type 0x04 (IPv4 Endpoint Option).    2. Option: Length is equal to 0x0009.    3. Discardable Flag is equal to 0.    4. Option: All reserved bits are set to 0.    5. Option: IPv4-Address is equal to **IP\_ADDRESS**.    6. Option: L4-Proto is equal to 0x06 (TCP).    7. Option: Port Number is equal to **PORT\_NUMBER**. |
| Reference | DS\_SD\_0070, DS\_SD\_0075 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_06] IPv4 Endpoint Option for UDP

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks format of an UDP related IPv4 Endpoint Option. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 address where the Enhanced Testability Service is reachable.  **PORT\_NUMBER**: (Uint16) Number of the UDP port where the Enhanced Testability Service is reachable. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provides an option of type 0x04 (IPv4 Endpoint Option).    2. Option: Length is equal to 0x0009.    3. Discardable Flag is equal to 0.    4. Option: All reserved bits are set to 0.    5. Option: IPv4-Address is equal to **IP\_ADDRESS**.    6. Option: L4-Proto is equal to 0x11 (UDP).    7. Option: Port Number is equal to **PORT\_NUMBER**. |
| Reference | DS\_SD\_0070, DS\_SD\_0075 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_07] IPv4 Multicast Option

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks format of an IPv4 Multicast Option. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  For the DUT a IPv4MulticastOption is configured (see point 1.1.2.3.5). |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 multicast address where the Enhanced Testability Service shall send multicast events and multicast notification events to.  **PORT\_NUMBER**: (Uint16) Number of the UDP port where the Enhanced Testability Service shall send multicast events and multicast notification events to. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0006. 4. DUT: Sends unicast SOME/IP-SD message carrying corresponding Subscribe Eventgroup Acknowledgement Entry with. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received before **TimeoutSubscribeAck** and contains:    1. Options Array provides an option of type 0x14 (IPv4 Multicast Option).    2. Option: Length is equal to 0x0009.    3. Discardable Flag is equal to 0.    4. Option: All reserved bits are set to 0.    5. Option: IPv4-Address is equal to **IP\_ADDRESS**.    6. Option: L4-Proto is equal to 0x11 (UDP).    7. Option: Port Number is equal to **PORT\_NUMBER**. |
| Reference | DS\_SD\_0080, DS\_SD\_0081, DS\_SD\_0242 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_08] IPv4 SD Endpoint Option

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks the format of an IPv4 SD Endpoint Option. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 address used in the SD Endpoint Option for the test.  **PORT\_NUMBER**: (Uint16) Number of the UDP port used in the SD Endpoint Option for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for ETS method **provideSDEndpointOption** with the following parameters:    * **duration** is set to **TimeoutOfferService**.    * **ip\_address** is set to **IP\_ADDRESS**.    * **portNumber** is set to **PORT\_NUMBER**. 2. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provides an option of type 0x24 (IPv4 SD Endpoint Option).    2. Option: Length is equal to 0x0009.    3. Option: Discardable Flag is equal to 0.    4. Option: All reserved bits are set to 0.    5. Option: IPv4-Address is equal to **IP\_ADDRESS**.    6. Option: L4-Proto is equal to 0x11 (UDP).    7. Option: Port Number is equal to **PORT\_NUMBER**. |
| Reference | DS\_SD\_0089 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_09] IPv4 SD Endpoint Option is included up to one time only

|  |  |
| --- | --- |
| Item | Description |
| Purpose | IPv4 SD Endpoint Option is included up to one time only. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 address used in the SD Endpoint Option for the test.  **PORT\_NUMBER**: (Uint16) Number of the UDP port used in the SD Endpoint Option for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for ETS method **provideSDEndpointOption** with the following parameters:    * **duration** is set to **TimeoutOfferService**.    * **ip\_address** is set to **IP\_ADDRESS**.    * **portNumber** is set to **PORT\_NUMBER**. 2. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provide one option of type 0x24 (IPv4 SD Endpoint Option).    2. Option: Length is equal to 0x0009.    3. Option: Discardable Flag is equal to 0.    4. Option: IPv4-Address is equal to **IP\_ADDRESS**.    5. Option: L4-Proto is equal to 0x11 (UDP).    6. Option: Port Number is equal to **PORT\_NUMBER**.    7. Option Array doesn’t provide another Option of Type 0x24. |
| Reference | DS\_SD\_0083 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_10] IPv4 SD Endpoint Option shall be the first Option in array if included

|  |  |
| --- | --- |
| Item | Description |
| Purpose | IPv4 SD Endpoint Option shall be the first Option in array if included. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 address used in the SD Endpoint Option for the test.  **PORT\_NUMBER**: (Uint16) Number of the UDP port used in the SD Endpoint Option for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for ETS method **provideSDEndpointOption** with the following parameters:    * **duration** is set to **TimeoutOfferService**.    * **ip\_address** is set to **IP\_ADDRESS**.    * **portNumber** is set to **PORT\_NUMBER**. 2. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Type of first option in Options Array is equal to 0x24 (IPv4 SD Endpoint Option)    2. Option: Length is equal to 0x0009.    3. Option: Discardable Flag is equal to 0.    4. Option: IPv4-Address is equal to **IP\_ADDRESS**.    5. Option: L4-Proto is equal to 0x11 (UDP).    6. Option: Port Number is equal to **PORT\_NUMBER**. |
| Reference | DS\_SD\_0085 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_11] IPv4 SD Endpoint Option shall not be referenced by any entry

|  |  |
| --- | --- |
| Item | Description |
| Purpose | IPv4 SD Endpoint Option shall not be referenced by any entry. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 address used in the SD Endpoint Option for the test.  **PORT\_NUMBER**: (Uint16) Number of the UDP port used in the SD Endpoint Option for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for ETS method **provideSDEndpointOption** with the following parameters:    * **duration** is set to **TimeoutOfferService**.    * **ip\_address** is set to **IP\_ADDRESS**.    * **portNumber** is set to **PORT\_NUMBER**. 2. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provide one option of type 0x24 (IPv4 SD Endpoint Option).    2. Option: Length is equal to 0x0009.    3. Option: Discardable Flag is equal to 0.    4. Option: IPv4-Address is equal to **IP\_ADDRESS**.    5. Option: L4-Proto is equal to 0x11 (UDP).    6. Option: Port Number is equal to **PORT\_NUMBER**.    7. Option of Type 0x24 (IPv4 SD Endpoint Option) is not referenced by any entry in Entries Array. |
| Reference | DS\_SD\_0086 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_12] Configuration Option: Length and Type

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the received Configuration Option have the proper of the Length and Type. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING\_1**: Arbitrary name part of a configuration string related item.  **UTF8\_STRING\_2**: Arbitrary value part of a configuration string related item.  **STRING\_LENGTH\_1**: Length of name part in number of characters.  **STRING\_LENGTH\_2**: Length of value part in number of characters. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for ETS method **provideConfigurationOption** with the following parameters:    * **duration** is set to **TimeoutOfferService**.    * **keyLength** is set to **STRING\_LENGTH\_1**.    * **key** is set to **UTF8\_STRING\_1**.    * **valueLength** is set to **STRING\_LENGTH\_2**.    * **value** is set to **UTF8\_STRING\_2**.    * **key2Length** is set to 0x00.    * **key2** is set to 0x00 (empty string). 2. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provides an option of type 0x01 (Configuration Option).    2. Option: Length is equal to sum of **STRING\_LENGTH\_1, STRING\_LENGTH\_2** and 4.    3. Option: All reserved bits are set to 0. |
| Reference | DS\_SD\_0090 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_13] Configuration Option: String contains character sequence separated with length field

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the received Configuration Option String contains a character sequence separated with length field. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING\_1**: Arbitrary name part of a configuration string related item.  **UTF8\_STRING\_2**: Arbitrary value part of a configuration string related item.  **STRING\_LENGTH\_1**: Length of name part in number of characters.  **STRING\_LENGTH\_2**: Length of value part in number of characters. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for ETS method **provideConfigurationOption** with the following parameters:    * **duration** is set to **TimeoutOfferService**.    * **keyLength** is set to **STRING\_LENGTH\_1**.    * **key** is set to **UTF8\_STRING\_1**.    * **valueLength** is set to **STRING\_LENGTH\_2**.    * **value** is set to **UTF8\_STRING\_2**.    * **key2Length** is set to 0x00.    * **key2** is set to 0x00 (empty string). 2. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provides an option of type 0x01 (Configuration Option).    2. Option: Value of 1st byte in Configuration String is equal to the sum of **STRING\_LENGTH\_1, STRING\_LENGTH\_2** and 1. |
| Reference | DS\_SD\_0092, DS\_SD\_0093 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_14] Configuration Option: String contains no character sequence after length field set to zero

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the received Configuration Option String contains no character sequence after length field set to zero. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING\_1**: Arbitrary name part of a configuration string related item.  **UTF8\_STRING\_2**: Arbitrary value part of a configuration string related item.  **STRING\_LENGTH\_1**: Length of name part in number of characters.  **STRING\_LENGTH\_2**: Length of value part in number of characters. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for ETS method **provideConfigurationOption** with the following parameters:    * **duration** is set to **TimeoutOfferService**.    * **keyLength** is set to **STRING\_LENGTH\_1**.    * **key** is set to **UTF8\_STRING\_1**.    * **valueLength** is set to **STRING\_LENGTH\_2**.    * **value** is set to **UTF8\_STRING\_2**.    * **key2Length** is set to 0x00.    * **key2** is set to 0x00 (empty string). 2. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provides an option of type 0x01 (Configuration Option).    2. Option: Configuration String contains only one byte with a value equal to 0x00 (“\0”).    3. Option: Value of last byte in Configuration String is equal to 0x00 (“\0”). |
| Reference | DS\_SD\_0094 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_15] Configuration Option: String contains key with value

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the received Configuration Option String contains a key with value. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING\_1**: Arbitrary name part of a configuration string related item.  **UTF8\_STRING\_2**: Arbitrary value part of a configuration string related item.  **STRING\_LENGTH\_1**: Length of name part in number of characters.  **STRING\_LENGTH\_2**: Length of value part in number of characters. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for ETS method **provideConfigurationOption** with the following parameters:    * **duration** is set to **TimeoutOfferService**.    * **keyLength** is set to **STRING\_LENGTH\_1**.    * **key** is set to **UTF8\_STRING\_1**.    * **valueLength** is set to **STRING\_LENGTH\_2**.    * **value** is set to **UTF8\_STRING\_2**.    * **key2Length** is set to 0x00.    * **key2** is set to 0x00 (empty string). 2. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provides an option of type 0x01 (Configuration Option).    2. Option: Configuration String consists of the following items in the preceding order:       * Single byte with a value equal to the sum of **STRING\_LENGTH\_1**, **STRING\_LENGTH\_2** and 1.       * Substring equal to **UTF8\_STRING\_1**.       * Single byte with a value equal to 0x3D (“=”).       * Substring equal to **UTF8\_STRING\_2**.       * Single byte with a value equal to 0x00 (“\0”). |
| Reference | DS\_SD\_0091, DS\_SD\_0095, DS\_SD\_0096, DS\_SD\_0097, DS\_SD\_0098 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_16] Configuration Option: String contains key with empty value

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the received Configuration Option String contains key with empty value. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING**: Arbitrary name part of a configuration string related item.  **STRING\_LENGTH**: (Uint32) Length of name part in number of characters. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for ETS method **provideConfigurationOption** with the following parameters:    * **duration** is set to **TimeoutOfferService**.    * **keyLength** is set to **STRING\_LENGTH**.    * **key** is set to **UTF8\_STRING**.    * **valueLength** is set to 0x00.    * **value** is set to 0x00 (empty string).    * **key2Length** is set to 0x00.    * **key2** is set to 0x00 (empty string). 2. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provides an option of type 0x01 (Configuration Option).    2. Option: Configuration String consists of the following items in the preceding order:       * Single byte with a value equal to **STRING\_LENGTH** + 1.       * Substring equal to **UTF8\_STRING**.       * Single byte with a value equal to 0x3D (“=”).       * Single byte with a value equal to 0x00 (“\0”). |
| Reference | DS\_SD\_0091, DS\_SD\_0095, DS\_SD\_0096, DS\_SD\_0097, DS\_SD\_0098, DS\_SD\_0100 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_17] Configuration Option: String contains key without value

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the received Configuration Option String contains a key without any value. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING\_1**: 1st arbitrary name part of a configuration string related item.  **UTF8\_STRING\_2**: 2nd arbitrary name part of a configuration string related item.  **STRING\_LENGTH\_1**: Length of 1st name part in number of characters.  **STRING\_LENGTH\_2**: Length of 2nd name part in number of characters. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for ETS method **provideConfigurationOption** with the following parameters:    * **duration** is set to **TimeoutOfferService**.    * **keyLength** is set to **STRING\_LENGTH\_1**.    * **key** is set to **UTF8\_STRING\_1**.    * **valueLength** is set to 0x00.    * **value** is set to 0x00 (empty string).    * **key2Length** is set to **STRING\_LENGTH\_2**.    * **key2** is set to **UTF8\_STRING\_2**. 2. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provides an option of type 0x01 (Configuration Option).    2. Option: Configuration String consists of the following items in the preceding order:       * Single byte with a value equal to **STRING\_LENGTH\_1** + 1.       * Substring equal to **UTF8\_STRING\_1**.       * Single byte with a value equal to 0x3D (“=”).       * Single byte with a value equal to **STRING\_LENGTH\_2** + 1.       * Substring equal to **UTF8\_STRING\_2**.       * Single byte with a value equal to 0x00 (“\0”). |
| Reference | DS\_SD\_0091, DS\_SD\_0095, DS\_SD\_0097, DS\_SD\_0098, DS\_SD\_0099 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_18] Configuration Option: String contains multiple key/value pairs of same key

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the received Configuration Option String contains multiple key/value pairs of same key. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING\_1**: Arbitrary name part of a configuration string related item.  **UTF8\_STRING\_2**: Arbitrary value part of a configuration string related item.  **STRING\_LENGTH\_1**: Length of name part in number of characters.  **STRING\_LENGTH\_2**: Length of value part in number of characters. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for ETS method **provideConfigurationOption** with the following parameters:    * **duration** is set to **TimeoutOfferService**.    * **keyLength** is set to **STRING\_LENGTH\_1**.    * **key** is set to **UTF8\_STRING\_1**.    * **valueLength** is set to **STRING\_LENGTH\_2**.    * **value** is set to **UTF8\_STRING\_2**.    * **key2Length** is set to **STRING\_LENGTH\_1**.    * **key2** is set to **UTF8\_STRING\_1**. 2. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provides an option of type 0x01 (Configuration Option).    2. Option: Configuration String consists of the following items in the preceding order:       * Single byte with a value equal to the sum of **STRING\_LENGTH\_1**, **STRING\_LENGTH\_2** and 1.       * Substring equal to **UTF8\_STRING\_1**.       * Single byte with a value equal to 0x3D (“=”).       * Substring equal to **UTF8\_STRING\_2**.       * Single byte with a value equal to **STRING\_LENGTH\_1** + 1.       * Substring equal to **UTF8\_STRING\_1**.       * Single byte with a value equal to 0x00 (“\0”). |
| Reference | DS\_SD\_0091, DS\_SD\_0101 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_19] Length covers message length from first byte after the field

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the received message Length covers message length. The first byte after the length field. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:   Entries Array provides one Offer Service Entry   * 1. Length is equal to the sum of Length of Entries Array, Length of Options Array and 20 |
| Reference | DS\_SD\_0008 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_20] Offer Service Entry shall reference an IPv4 Endpoint Option

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send Offer Service entries which reference at least one IPv4 Endpoint Option. |
| Preconditions | DUT shall provide at least one SOME/IP service. |
| Test Setup | Blackbox Test (see chapter 4.1.1) |
| Test variables | none |
| Test Procedure | 1. DUT: Activates service and enters Service Discovery start-up sequence. 2. DUT: Sends multicast SOME/IP-SD message. 3. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides one or more entries of type 0x01 (Offer Service).    2. Each Offer Service Entry references at least one option of type 0x04 (IPv4 Endpoint Option). |
| Reference | DS\_SD\_0076, DS\_SD\_0125 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_21] Offer Service Entry shall reference an IPv4 Endpoint Option per Transport Protocol Binding

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the send Offer Service Entry shall reference an IPv4 Endpoint Option per Transport Protocol Binding. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for the ETS. 2. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides one Offer Service Entry.    2. Offer Service Entry references an option of type 0x04 (IPv4 Endpoint Option) with L4-Proto set to 0x06 (TCP) and Discardable Flag is equal to 0.    3. Offer Service Entry references an option of type 0x04 (IPv4 Endpoint Option) with L4-Proto set to 0x11 (UDP) and Discardable Flag is equal to 0. |
| Reference | DS\_SD\_0126 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_22] Subscribe Eventgroup ACK can reference an IPv4 Multicast Option

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the send Subscribe Eventgroup ACK can reference an IPv4 Multicast Option. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  For the DUT a IPv4MulticastOption is configured (see point 1.1.2.3.5). |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for the ETS. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0006. 4. DUT: Sends unicast SOME/IP-SD message carrying corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService**. 2. SOME/IP-SD message is received before **TimeoutSubscribeAck** and contains:    1. Entries Array provides one Subscribe Eventgroup Ack Entry.    2. Subscribe Eventgroup Ack Entry references an option of type 0x14 (IPv4 Multicast Option). |
| Reference | DS\_SD\_0079, DS\_SD\_0082 |
| Notes |  |

#### [SOMEIPSRV\_SD\_FORMAT\_23] Support of multiple Entries per message

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the send SOME/IP-SD message Support of multiple Entries per message. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for the ETS. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying the following entries:    * Subscribe Eventgroup Entry for Eventgroup ID 0x0005.    * Subscribe Eventgroup Entry for Eventgroup ID 0x0006. 4. DUT: Sends unicast SOME/IP-SD message carrying multiple entries. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received before **TimeoutSubscribeAck** and contains:    1. Entries Array provides two entries.    2. 1st Entry: Type is equal to 0x07 (Subscribe Eventgroup Acknowledgement Entry).    3. 2nd Entry: Type is equal to 0x07 (Subscribe Eventgroup Acknowledgement Entry). |
| Reference | DS\_SD\_0039 |
| Notes |  |

### Service Discovery Communication Behavior

#### [SOMEIPSRV\_SD\_BEHAVIOR\_01] Initial Wait Phase: Enter when service is available

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that Initial Wait Phase of the Service Discovery start-up sequence is entered when a new service instance becomes available. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT related settings for INITIAL\_DELAY must not define any delay. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SERVICE\_DOWN**: (Uint8) Duration in seconds [sec] the ETS shall be suspended. Must be at least as long as the configured CYCLIC\_OFFER\_DELAY.  **T\_RAW\_STARTUP**: (Uint16) Duration in milliseconds [msec] until DUT is expected to send the first Offer Service Entry of an activated SOME/IP service. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **suspendInterface** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**. 2. DUT: Stops offering the ETS. 3. TESTER: Observes further DUT communication for a period of **T\_SERVICE\_DOWN**. 4. DUT: Activates the ETS and enters Service Discovery start-up sequence. 5. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 6. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **T\_RAW\_STARTUP** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. |
| Reference | DS\_SD\_0123, DS\_SD\_0143, DS\_SD\_0145, DS\_SD\_0199 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_02] Initial Wait Phase: Delay sending of 1st Offer Service

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks delay before sending the very first Offer Service entry of a specific service instance during Initial Wait Phase of the Service Discovery start-up sequence. The phase shall use dedicated timing settings. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT related settings for INITIAL\_DELAY must define a minimum delay of at least 100 milliseconds [msec]. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SERVICE\_DOWN**: (Uint8) Duration in seconds [sec] the ETS shall be suspended. Must be at least as long as the configured CYCLIC\_OFFER\_DELAY.  **T\_INITIAL\_DELAY\_MIN**: (Uint8) The minimal expected delay before sending out the first Offer Service Entry for an activated SOME/IP service.  **T\_INITIAL\_DELAY\_MAX**: (Uint8) The maximal expected delay before sending out the first Offer Service Entry for a searched SOME/IP service. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **suspendInterface** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**. 2. DUT: Stops offering the ETS. 3. TESTER: Observes further DUT communication for a period of **T\_SERVICE\_DOWN**. 4. DUT: Activates the ETS and enters Service Discovery start-up sequence. 5. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 6. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received after **T\_INITIAL\_DELAY\_MIN** but before **T\_INITIAL\_DELAY\_MAX** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. |
| Reference | DS\_SD\_0143, DS\_SD\_0146 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_03] Initial Wait Phase: Delay of 1st Offer Service is random value

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the used delay during Initial Wait Phase is determined by choosing a random value from a predefined range. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT related settings for INITIAL\_DELAY must define different values for lower and upper limit. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SERVICE\_DOWN**: (Uint8) Duration in seconds [sec] the ETS shall be suspended. Must be at least as long as the configured CYCLIC\_OFFER\_DELAY.  **N\_ITERATIONS**: (Uint8) Number of iterations the initial delay shall be measured. Must not be less than a number of 3. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **suspendInterface** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**. 2. DUT: Stops offering the ETS. 3. TESTER: Observes further DUT communication for a period of **T\_SERVICE\_DOWN**. 4. DUT: Activates the ETS and enters Service Discovery start-up sequence. 5. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 6. TESTER: Verify received SOME/IP-SD message. 7. Repeat preceding steps 1 to 6 until a total number of **N\_ITERATIONS** are processed. 8. TESTER: Compares the measured initial delay for SOME/IP-SD messages received in steps 1 to 7. |
| Verification | 1. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 3. Measured delay values differ in at least 1 millisecond [msec] +/- **TimingToleranceMillisec** |
| Reference | DS\_SD\_0143, DS\_SD\_0147, DS\_SD\_0148 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_04] Repetition Phase: Delay sending of 2nd Offer Service

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks delay before sending the first Offer Service entry of a specific service instance during Repetition Phase of the Service Discovery start-up sequence. The phase shall use dedicated timing settings. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP parameter REPETITIONS\_MAX must not be zero.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SERVICE\_DOWN**: (Uint8) Duration in seconds [sec] the ETS shall be suspended. Must be at least as long as the configured CYCLIC\_OFFER\_DELAY.  **T\_REPETITION\_BASE\_DELAY**: (Uint16) Base delay value in milliseconds [msec] configured for the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **suspendInterface** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**. 2. DUT: Stops offering the ETS. 3. TESTER: Observes further DUT communication for a period of **T\_SERVICE\_DOWN**. 4. DUT: Activates the ETS and enters Service Discovery start-up sequence. 5. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 6. TESTER: Verify received SOME/IP-SD message. 7. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 8. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 3. SOME/IP-SD message is received immediately after **T\_REPETITION\_BASE\_DELAY** +/- **TimingToleranceMillisec** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. |
| Reference | DS\_SD\_0143, DS\_SD\_0144, DS\_SD\_0153, DS\_SD\_0154 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_05] Repetition Phase: Number of sent Offer Service Entries

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks the number of Offer Service entries which were sent for a specific service instance during Repetition Phase of the Service Discovery start-up sequence. The maximum number of repetitions shall correspond to a predefined value. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP Parameter REPETITIONS\_MAX must be a value greater than 1.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SERVICE\_DOWN**: (Uint8) Duration in seconds [sec] the ETS shall be suspended. Must be at least as long as the configured CYCLIC\_OFFER\_DELAY.  **N\_REPETITIONS** (Uint8): Number of message repetitions configured for the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **suspendInterface** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**. 2. DUT: Stops offering the ETS. 3. TESTER: Observes further DUT communication for a period of **T\_SERVICE\_DOWN**. 4. DUT: Activates the ETS and enters Service Discovery start-up sequence. 5. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 6. TESTER: Verify received SOME/IP-SD message. 7. Repeat preceding steps 5 and 6 for a total number of **N\_REPETITIONS**. 8. TESTER: Verify SOME/IP-SD messages received in step 7. |
| Verification | 1. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 3. Measured delay value between two consecutive SOME/IP-SD messages is doubled by each following message, considering a tolerance of+/- **TimingToleranceMillisec**. |
| Reference | DS\_SD\_0143, DS\_SD\_0156 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_06] Repetition Phase: Delay is doubled after each following Offer Service

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks delay between Offer Service entries of a specific service instance, which were sent during Repetition Phase of the Service Discovery start-up sequence. After each following entry the delay shall be doubled. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP parameter REPETITIONS\_MAX must be a value greater than 1.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SERVICE\_DOWN**: (Uint8) Duration in seconds [sec] the ETS shall be suspended. Must be at least as long as the configured CYCLIC\_OFFER\_DELAY.  **N\_REPETITIONS**: (Uint8) Number of message repetitions configured for the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **suspendInterface** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**. 2. DUT: Stops offering the ETS. 3. TESTER: Observes further DUT communication for a period of **T\_SERVICE\_DOWN**. 4. DUT: Activates the ETS and enters Service Discovery start-up sequence. 5. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 6. TESTER: Verify received SOME/IP-SD message. 7. Repeat preceding steps 5 and 6 for a total number of **N\_REPETITIONS**. 8. TESTER: Compares the measured repetition delay for SOME/IP-SD messages received in step 7. |
| Verification | 1. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 3. Measured delay value between two consecutive SOME/IP-SD messages is doubled by each following message, considering a tolerance of+/- **TimingToleranceMillisec**. |
| Reference | DS\_SD\_0143, DS\_SD\_0155 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_07] Repetition Phase: Same settings for all services within an ECU

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to use the same Repetition Phase related settings for all of its service instances. |
| Preconditions | DUT is running and offering the Enhanced Testability Service in two instances which only differs by its Major version.  SOME/IP Parameter REPETITIONS\_MAX must be a value greater than 1.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero.  Method **serviceActivate** with **instanceKey** set to 0x03 (alternative major version) may be called for this test case. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SERVICE\_DOWN**: (Uint8) Duration in seconds [sec] the ETS shall be suspended. Must be at least as long as the configured CYCLIC\_OFFER\_DELAY.  **N\_REPETITIONS**: (Uint8) Number of message repetitions configured for the SOME/IP-SD related Repetition Phase of the DUT.  **T\_REPETITION\_PHASE**: (Uint16) The expected overall duration in milliseconds [msec] of the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **suspendInterface** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**.    * Interface Version is set to **ETS\_MajorVersion\_0**.   ETS service for **ETS\_MajorVersion\_0** is addressed.   1. TESTER: Sends SOME/IP Request-No-Return message for method **suspendInterface** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**.    * Interface Version is set to **ETS\_MajorVersion\_1**.   ETS service for **ETS\_MajorVersion\_1** is addressed.   1. DUT: Stops offering the ETS. 2. TESTER: Observes further DUT communication for a period of **T\_SERVICE\_DOWN**. 3. DUT: Activates the ETS and enters Service Discovery start-up sequence. 4. DUT: Sends multicast SOME/IP-SD message carrying Offer Service Entries. 5. TESTER: Verify received SOME/IP-SD message. 6. Repeat preceding steps 5 and 6 for a total number of **N\_REPETITIONS**. 7. TESTER: Verify SOME/IP-SD messages received in step 7. |
| Verification | 1. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides two entries of type 0x01 (Offer Service Entry).    2. Both Entries: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Both Entries: TTL is equal to **ETS\_Lifetime**.    5. One Entry: Major version is equal to **ETS\_MajorVersion\_0**.    6. One Entry: Major version is equal to **ETS\_MajorVersion\_1**. 3. All SOME/IP-SD messages were received before **T\_REPETITION\_PHASE** is elapsed. |
| Reference | DS\_SD\_0143, DS\_SD\_0161 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_08] Repetition Phase: After Link-loss/Link-up jump to Initial Wait Phase

|  |  |
| --- | --- |
| Item | Description |
| Purpose | At the Repetition Phase jump to the Initial Wait Phase after Link-loss/Link-up happens. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP Parameter REPETITIONS\_MAX must be a value greater than 1.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero.  (For example, **suspendEthernetInterface** method may be called for this test) |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SERVICE\_DOWN**: (Uint8) Duration in seconds [sec] the ETS shall be suspended. Must be at least as long as the configured CYCLIC\_OFFER\_DELAY.  **T\_REPETITION\_BASE\_DELAY**: (Uint16) Base delay value in milliseconds [msec] configured for the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **suspendInterface** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**. 2. DUT: Stops offering the ETS. 3. TESTER: Observes further DUT communication for a period of **T\_SERVICE\_DOWN**. 4. DUT: Activates the ETS and enters Service Discovery start-up sequence. 5. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 6. TESTER: Verify received SOME/IP-SD message. 7. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Forces link-down on the DUT’s network interface until a period of 2 x **T\_REPETITION\_BASE\_DELAY** is elapsed. 10. DUT: After Wake up, Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 11. TESTER: Verify received SOME/IP-SD message. 12. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 13. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 3. SOME/IP-SD message is received immediately after **T\_REPETITION\_BASE\_DELAY** +/- **TimingToleranceMillisec** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 4. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 5. SOME/IP-SD message is received immediately after **T\_REPETITION\_BASE\_DELAY** +/- **TimingToleranceMillisec** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. |
| Reference | DS\_SD\_0143, DS\_SD\_0144, DS\_SD\_0182 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_09] Main Phase: Delay sending of next Offer Service

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks delay before sending the first Offer Service entries of a specific service instance during Main Phase of the Service Discovery. The phase shall use dedicated timing settings. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SERVICE\_DOWN**: (Uint8) Duration in seconds [sec] the ETS shall be suspended. Must be at least as long as the configured CYCLIC\_OFFER\_DELAY.  **N\_REPETITIONS**: (Uint8) Number of message repetitions configured for the SOME/IP-SD related Repetition Phase of the DUT.  **T\_CYCLIC\_OFFER\_DELAY**: (Uint8) The value of SOME/IP parameter CYCLIC\_OFFER\_DELAY which is used by the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **suspendInterface** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**. 2. DUT: Stops offering the ETS. 3. TESTER: Observes further DUT communication for a period of **T\_SERVICE\_DOWN**. 4. DUT: Activates the ETS and enters Service Discovery start-up sequence. 5. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 6. TESTER: Verify received SOME/IP-SD message. 7. Repeat preceding steps 5 and 6 for a total number of **N\_REPETITIONS**. 8. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 9. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 3. SOME/IP-SD message is received immediately after **T\_CYCLIC\_OFFER\_DELAY** +/- **TimingTolerance** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. |
| Reference | DS\_SD\_0143, DS\_SD\_0144, DS\_SD\_0163, DS\_SD\_0164 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_10] Main Phase: Offer Service is sent cyclically

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Check that Offer Service entries of a specific service instance are sent with correct timing during Main Phase of the Service Discovery. Entries shall be sent cyclically. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **N\_REPETITIONS** (Uint8): Number of Offer Services to observe for the test. Should be at least 3 times.  **T\_CYCLIC\_OFFER\_DELAY** : (Uint8) The value of SOME/IP parameter CYCLIC\_OFFER\_DELAY which is used by the DUT. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. Repeat preceding steps 1 and 2 until a total number of **N\_REPETITIONS** are processed. 4. TESTER: Verify the measured cyclic offer delay of SOME/IP-SD messages received in step 3. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. Measured delay value between two consecutive SOME/IP-SD messages is equal to **T\_CYCLIC\_OFFER\_DELAY** +/- **TimingTolerance**. |
| Reference | DS\_SD\_0143, DS\_SD\_0165, DS\_SD\_0169 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_11] Main Phase: After Link-loss/Link-up jump to Initial Wait Phase

|  |  |
| --- | --- |
| Item | Description |
| Purpose | After a Link-loss/Link-up in Main Phase jump to Initial Wait Phase. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP Parameter REPETITIONS\_MAX must not be zero.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero.  (For example, **suspendEthernetInterface** method may be called for this test) |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_REPETITION\_BASE\_DELAY**: (Uint16) Base delay value in milliseconds [msec] configured for the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Forces link-down on the DUT’s network interface until a period of **TimeoutOfferService** is elapsed. 4. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 7. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 3. SOME/IP-SD message is received immediately after **T\_REPETITION\_BASE\_DELAY** +/- **TimingToleranceMillisec** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. |
| Reference | DS\_SD\_0143, DS\_SD\_0144, DS\_SD\_0182, DS\_SD\_0224 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_12] Skip Repetition Phase if max repetitions is set to zero

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that Repetition Phase of the Service Discovery start-up sequence is skipped when predefined value of repetitions is set to zero. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP parameter REPETITIONS\_MAX must be zero.  (For example, **setRepetitionMaxCount** method may be called for this test). |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SERVICE\_DOWN**: (Uint8) Duration in seconds [sec] the ETS shall be suspended. Must be at least as long as the configured CYCLIC\_OFFER\_DELAY.  **T\_CYCLIC\_OFFER\_DELAY**: (Uint8) The value of SOME/IP parameter CYCLIC\_OFFER\_DELAY which is used by the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **suspendInterface** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**. 2. DUT: Stops offering the ETS. 3. TESTER: Observes further DUT communication for a period of **T\_SERVICE\_DOWN**. 4. DUT: Activates the ETS and enters Service Discovery start-up sequence. 5. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 6. TESTER: Verify received SOME/IP-SD message. 7. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 8. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 3. SOME/IP-SD message is received immediately after **T\_CYCLIC\_OFFER\_DELAY** +/- **TimingTolerance** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. |
| Reference | DS\_SD\_0143, DS\_SD\_0144, DS\_SD\_0162 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_13] Request Response Delay for Offer Service as answer of multicast Find Service

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Request Response Delay for Offer Service as answer of multicast Find Service. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT related settings for REQUEST\_RESPONSE\_DELAY must define a minimum delay of at least 1 second [sec]. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_RESPONSE\_DELAY\_MIN**: (Uint8) The minimal expected delay for answering to Service Discovery entries when DUT related setting of REQUEST\_RESPONSE\_DELAY is got involved.  **T\_RESPONSE\_DELAY\_MAX**: (Uint8) The maximal expected delay for answering to Service Discovery entries when DUT related setting of REQUEST\_RESPONSE\_DELAY is got involved.  **T\_CYCLIC\_OFFER\_DELAY**: (Uint8) The value of SOME/IP parameter CYCLIC\_OFFER\_DELAY which is used by the DUT. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s ETS instance **ETS\_InstanceID\_0.** Shall be send before ½ T\_CYCLIC\_OFFER\_DELAY is elapsed since step 1. 4. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received after **T\_RESPONSE\_DELAY\_MIN** but before **T\_RESPONSE\_DELAY\_MAX** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Destination IP of received SD message is equal to **TESTER\_IPv4Address**. |
| Reference | DS\_SD\_0172 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_15] Request Response Delay shall not be applied if unicast message is answered

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Request Response Delay shall not be applied if unicast message is answered. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT related settings for REQUEST\_RESPONSE\_DELAY must define a minimum delay of at least 1 second [sec]. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_RESPONSE\_DELAY\_MIN**: (Uint8) The minimal expected delay for answering to Service Discovery entries when DUT related setting of REQUEST\_RESPONSE\_DELAY is got involved. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s ETS instance. 4. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received before **T\_RESPONSE\_DELAY\_MIN** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Detination IP of received SD message is equal to **TESTER\_IPv4Address**. |
| Reference | DS\_SD\_0173 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_16] Request Response Delay is randomly chosen between min and max

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Request Response Delay is randomly chosen between min and max. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT related settings for REQUEST\_RESPONSE\_DELAY must define different values for lower and upper limit. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_RESPONSE\_DELAY\_MIN**: (Uint8) The minimal expected delay for answering to Service Discovery entries when DUT related setting of REQUEST\_RESPONSE\_DELAY is got involved.  **T\_RESPONSE\_DELAY\_MAX**: (Uint8) The maximal expected delay for answering to Service Discovery entries when DUT related setting of REQUEST\_RESPONSE\_DELAY is got involved. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s ETS instance **ETS\_InstanceID\_0**. 4. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s ETS instance **ETS\_InstanceID\_0**. 7. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Compares the measured response delay for SOME/IP-SD messages received in step 5 and 8. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received after **T\_RESPONSE\_DELAY\_MIN** but before **T\_RESPONSE\_DELAY\_MAX** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry**:** Destination IP of received SD message is equal to **TESTER\_IPv4Address.** 3. SOME/IP-SD message is received after **T\_RESPONSE\_DELAY\_MIN** but before **T\_RESPONSE\_DELAY\_MAX** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Destination IP of received SD message is equal to **TESTER\_IPv4Address**. 4. Measured delay values differ in at least 1 millisecond [msec]. |
| Reference | DS\_SD\_0174, DS\_SD\_0175 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_17] Same Request Response Delay within an ECU

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to use same Request Response Delay for all hosted service instances. |
| Preconditions | DUT is running and offering the Enhanced Testability Service in two instances which only differs by its Major version.  DUT related settings for REQUEST\_RESPONSE\_DELAY must define a fix delay of at least 1 second [sec] (minimum delay = maximum delay ).  Method **serviceActivate** with **instanceKey** set to 0x03 (alternative major version) may be called for this test case. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_RESPONSE\_DELAY\_MIN**: (Uint8) The minimal expected delay for answering to Service Discovery entries when DUT related setting of REQUEST\_RESPONSE\_DELAY is got involved.  **T\_RESPONSE\_DELAY\_MAX**: (Uint8) The maximal expected delay for answering to Service Discovery entries when DUT related setting of REQUEST\_RESPONSE\_DELAY is got involved. |
| Test Procedure | 1. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for ETS instance **ETS\_InstanceID\_0** of version **ETS\_MajorVersion\_0**. 2. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 3. TESTER: Verify received SOME/IP-SD message. 4. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for ETS instance of version **ETS\_MajorVersion\_1**. 5. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 6. TESTER: Verify received SOME/IP-SD message. 7. TESTER: Verify measured Request Response Delay of step 3 and 6. |
| Verification | 1. SOME/IP-SD message is received after **T\_RESPONSE\_DELAY\_MIN** but before **T\_RESPONSE\_DELAY\_MAX** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Major Version is equal to **ETS\_MajorVersion\_0**. 2. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Major Version is equal to **ETS\_MajorVersion\_1**.    5. Entry: Destination IP of received SD message is equal to **TESTER\_IPv4Address**. 3. Measured delay values are equal to **T\_RESPONSE\_DELAY** +/ **TimingTolerance**. |
| Reference | DS\_SD\_0178 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_18] Offer Service is sent as unicast to answer Find Service received in Repetition Phase

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT has to answer the Testers Multicast Find Service with a Unicast Offer Service, stating all IPs and Ports needed to fulfill all sort of possible Communications with it. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP Parameter REPETITIONS\_MAX must be a value greater than 1.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SERVICE\_DOWN**: (Uint8) Duration in seconds [sec] the ETS shall be suspended. Must be at least as long as the configured CYCLIC\_OFFER\_DELAY.  **IP\_ADDRESS**: IPv4 unicast address used as source address of the simulated SOME/IP client for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **suspendInterface** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**. 2. DUT: Stops offering the ETS. 3. TESTER: Observes further DUT communication for a period of **T\_SERVICE\_DOWN**. 4. DUT: Activates the ETS and enters Service Discovery start-up sequence. 5. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 6. TESTER: Verify received SOME/IP-SD message. 7. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s ETS instance **ETS\_InstanceID\_0**. IP Source Address is set to **IP\_ADDRESS**. 8. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 9. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 3. SOME/IP-SD message is received before **TimeoutResponse** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: TTL is equal to **ETS\_Lifetime**.    4. IP Destination Address is equal to **IP\_ADDRESS**. |
| Reference | DS\_SD\_0179 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_19] Offer Service is sent as unicast to answer Find Service received in Main Phase less than ½ cycle

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Offer Service of the DUT is sent as unicast to answer the Find Service, which is received in Main Phase less than ½ cycle. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP parameter CYCLIC\_OFFER\_DELAY must be set to at least 2 seconds. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 unicast address used as source address of the simulated SOME/IP client for the test.  **T\_CYCLIC\_OFFER\_DELAY**: (Uint8) The value of SOME/IP parameter CYCLIC\_OFFER\_DELAY which is used by the DUT. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s ETS instance **ETS\_InstanceID\_0**. Shall be send before ½ T\_CYCLIC\_OFFER\_DELAY is elapsed since step 1. IP Source Address is set to **IP\_ADDRESS**. 4. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutResponse** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**.    5. IP Destination Address is equal to **IP\_ADDRESS**. |
| Reference | DS\_SD\_0180 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_20] Offer Service is sent as multicast to answer Find Service received in Main Phase equal or more than ½ cycle

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Offer Service of the DUT is sent as multicast to answer the Find Service which is received in Main Phase equal or more than ½ cycle. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP parameter CYCLIC\_OFFER\_DELAY must be set to at least 2 seconds. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 multicast address used by DUT as destination address for SOME/IP-SD related multicast messages.  **T\_CYCLIC\_OFFER\_DELAY**: (Uint8) The value of SOME/IP parameter CYCLIC\_OFFER\_DELAY which is used by the DUT. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Waits until ½ **T\_CYCLIC\_OFFER\_DELAY** is elapsed. 4. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s ETS instance **ETS\_InstanceID\_0**. 5. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 6. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutResponse** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**.    5. IP Destination Address is equal to **IP\_ADDRESS**. |
| Reference | DS\_SD\_0180 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_21] Sending Unicast and Multicast use different Session ID counters

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to use different unicast and the Multicast Session ID counters for sending Some/IP SD messages.  After more than the initial offer Service the session Id of this Offer Service from the DUT, which is send as multicast is greater than 1. The following Acknowledgement of the DUT, which is send as unicast have a Session ID of 1. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT related Session ID counter must be < 1. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying an Subscribe Eventgroup Entry for ID 0x0005. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Session ID is greater than 0x0001. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Eventgroup ID is equal to 0x0005.    5. Entry: Session ID is equal to 0x0001. |
| Reference | DS\_SD\_0250, DS\_SD\_0251 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_22] Each Unicast relation use different Session ID counters for sending

|  |  |
| --- | --- |
| Item | Description |
| Purpose | 2 Clients with different Endpoint options, communicate with the DUT. The session ID counter of each unicast message is different, cause both clients are sending the Subscription for the DUT for the first time. Both session ID counters runs independently and the sending Session ID doesn’t increment. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. Two ETS Services Instances running at the ECU. Use a fresh unicast relation, so the session id counter start with 1. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS\_1**: IPv4 unicast address used as source address of the simulated SOME/IP client for the test.  **IP\_ADDRESS\_2**: Arbitrary IPv4 unicast address used for the test. Must be different than value of test variable IP\_ADDRESS\_1. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message from IP Endpoint address **IP\_ADDRESS\_1**, carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0005. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends unicast SOME/IP-SD message from IP Endpoint address **IP\_ADDRESS\_2**, carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0005. 7. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 8. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received at Endpoint address **IP\_ADDRESS\_1** within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Session ID is equal to 0x0001.    5. Entry: TTL is equal to **ETS\_Lifetime**. 3. SOME/IP-SD message is received at Endpoint address **IP\_ADDRESS\_2** within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Session ID is equal to 0x0001.    5. Entry: TTL is equal to **ETS\_Lifetime**. |
| Reference | DS\_SD\_0250, DS\_SD\_0251 |
| Notes | So its a different Counter ID cause both are sending a subscription to the client the first time, but the Session id of both Instances is 1 and not count higher. |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_23] Different Session ID counters for received Unicast and Multicast messages of same ECU

|  |  |
| --- | --- |
| Item | Description |
| Purpose | When using different Session ID counters for unicast and multicast messages it is expected that a subscription on the DUT is not affected when receiving multicast messages and unicast messages with the same Session ID. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **SESSION\_ID\_1**: (Uint16) First session ID uses for the test.  **IP\_ADDRESS\_1**: IPv4 unicast address used as source address of the simulated SOME/IP client for the test.  **T\_SUBSCRIPTION**: (Uint8) Lifetime of Eventgroup subscription used for the test. Must be at least 3 seconds. |
| Test Procedure | 1. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s ETS instance with session ID = **SESSION\_ID\_1** and reboot Flag = 1. 2. DUT: Sends Unicast SOME/IP-SD message carrying an Offer Service Entry. 3. TESTER: Verify received SOME/IP-SD messages. 4. TESTER: Sends unicast SOME/IP-SD message from IP Endpoint address **IP\_ADDRESS\_1,** carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0005.   The session ID is set to **SESSION\_ID\_1** + 1 and reboot flag=1**.**   1. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request-No-Return message for method **triggerEventUINT8** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds)    * **duration** is set to **T\_SUBSCRIPTION**.    * **debounceTime** is set to 0x00000001 (1 second). 4. DUT: Sends SOME/IP Notification messages for **TestEventUINT8.** 5. TESTER: Verify received SOME/IP message. 6. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s ETS instance with session ID = **SESSION\_ID\_1** + 1and reboot Flag = 1. 7. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 8. TESTER: Verify received SOME/IP-SD messages. 9. DUT: Sends SOME/IP Notification messages for **TestEventUINT8.** 10. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Session ID is equal to 0x0001.    4. Instance ID is equal to **ETS\_InstanceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: Eventgroup ID is equal to 0x0005. 3. SOME/IP Notification message is received within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**). 4. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to **ETS\_InstanceID\_0**. 5. SOME/IP Notification message is received within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**). |
| Reference | DS\_SD\_0189, DS\_SD\_0250, DS\_SD\_0251 |
| Notes | One Client communicate with the DUT. The Client send a Find Service to the DUT with the session ID of 1. The session ID of the following Subscription of the Client is greater than 1 and the Server send events to the client. Then the Client send a session ID of 2. The events of the DUT works Farther correct, even though the session Id of the client changed. |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_24] Different Session ID counters per ECU for received Multicast messages

|  |  |
| --- | --- |
| Item | Description |
| Purpose | It is expected that a subscription on the DUT is not affected when receiving multicast message with a session ID which is smaller than one of the last received multicast message from another ECU. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **SESSION\_ID\_1:** (Uint16) First session ID uses for the test.  **SESSION\_ID\_2:** (Uint16) Second session ID uses for the test (must be greater than SESSION\_ID\_1)  **IP\_ADDRESS\_1**: IPv4 unicast address used as source address of the simulated SOME/IP client for the test.  **IP\_ADDRESS\_2**: Arbitrary IPv4 unicast address used for the test (must be different than value of test variable IP\_ADDRESS\_1) |
| Test Procedure | 1. TESTER: Sends multicast SOME/IP-SD message from IP Endpoint address **IP\_ADDRESS\_1**, carrying a Find Service Entry for the DUT’s ETS instance with the session ID = **SESSION\_ID\_1** and reboot flag= 1. 2. DUT: Sends SOME/IP-SD Unicast message carrying an Offer Service Entry. 3. TESTER: Verify received SOME/IP-SD messages. 4. TESTER: Sends multicast SOME/IP-SD message from IP Endpoint address **IP\_ADDRESS\_2**, carrying a Find Service Entry for the DUT’s ETS instance. The session ID is set to **SESSION\_ID\_2** and reboot flag= 1. 5. DUT: Sends SOME/IP-SD Unicast message carrying an Offer Service Entry. 6. TESTER: Verify received SOME/IP-SD messages. 7. TESTER: Sends unicast SOME/IP-SD message carrying the following entries:    * Subscribe Eventgroup Entry for Eventgroup ID 0x0005 from IP Endpoint address **IP\_ADDRESS\_1**. 8. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 9. TESTER: Verify received SOME/IP-SD message. 10. TESTER: Sends unicast SOME/IP-SD message carrying the following entries:     * Subscribe Eventgroup Entry for Eventgroup ID 0x0005 from IP Endpoint address **IP\_ADDRESS\_2**. 11. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 12. TESTER: Verify received SOME/IP-SD message. 13. TESTER: Sends SOME/IP Request-No-Return message for method **triggerEventUINT8** ETS instance **ETS\_InstanceID\_0** with the following parameters:     * **delay** is set to 0x00000000 (0 seconds)     * **duration** is set to **T\_SUBSCRIPTION**.     * **debounceTime** is set to 0x00000001 (1 second). 14. DUT: Sends Unicast SOME/IP Notification messages for **TestEventUINT8.** 15. TESTER: Verify received SOME/IP message. 16. DUT: Sends Unicast SOME/IP Notification messages for **TestEventUINT8.** 17. TESTER: Verify received SOME/IP message. 18. TESTER: Sends multicast SOME/IP-SD message from IP Endpoint address **IP\_ADDRESS\_1**, carrying a Find Service Entry for the DUT’s ETS instance. The session ID is set to **SESSION\_ID\_1**  and reboot flag=1. 19. DUT: Sends SOME/IP-SD Unicast message carrying an Offer Service Entry. 20. TESTER: Verify received SOME/IP-SD message. 21. DUT: Sends Unicast SOME/IP Notification messages for **TestEventUINT8.** 22. TESTER: Verify received SOME/IP message. 23. TESTER: Observes further DUT communication for a period of **T\_SUBSCRIPTION** from IP endpoint address **IP\_ADDRESS\_1**. |
| Verification | 1. SOME/IP-SD message is received at IP Endpoint Address **IP\_ADDRESS\_1** within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Session ID is equal to 0x0001.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. SOME/IP-SD message is received at IP Endpoint Address **IP\_ADDRESS\_2** within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to ETS\_**ServiceID\_0**.    3. Entry: Session ID is equal to 0x0001.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 3. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: Eventgroup ID is equal to 0x0005. 4. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: Eventgroup ID is equal to 0x0005. 5. SOME/IP Notification message is received at IP Endpoint Address **IP\_ADDRESS\_1** within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**). 6. SOME/IP Notification message is received at IP Endpoint Address **IP\_ADDRESS\_2** within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**). 7. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Instance ID is equal to **ETS\_InstanceID\_0**.    3. Entry: Session ID is equal to 0x0003(When Session Handling is not applied to the notification message) or 0x0004(When Session Handling is applied to the notification message).    4. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 8. SOME/IP Notification message is received at IP Endpoint Address **IP\_ADDRESS\_2** within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**). 9. No SOME/IP Notification message were were observed at IP Endpoint Address **IP\_ADDRESS\_1** which contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**). |
| Reference | DS\_SD\_0189, DS\_SD\_0250, DS\_SD\_0251 |
| Notes | Two Clients with different Endpoint options, communicate with the DUT.  Both clients Sends a multicast Find service short behind, where the session ID of the first client is 1 and from the second client greater than 1. Both send events. After the next cycle the find service of the first client is 1 again and of the second it is N + 1. It means, that the first client had have a reboot and deleted all subscriptions. But the second client run farther without a failure or a reboot. It shows that there are different Session ID counters, because the second client still sends continuously events to the DUT, without a reboot was detected. |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_25] Different Session ID counters per ECU for received Unicast messages

|  |  |
| --- | --- |
| Item | Description |
| Purpose | It is expected that a subscription on the DUT is not affected when receiving unicast message with a session ID which is smaller than one of the last received unicast message from another ECU. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **SESSION\_ID\_1**: (Uint16) First session ID uses for the test.  **SESSION\_ID\_2:** (Uint16) Second session ID uses for the test (must be greater than SESSION\_ID\_1)  **IP\_ADDRESS\_1**: IPv4 unicast address used as source address of the simulated SOME/IP client for the test.  **IP\_ADDRESS\_2**: Arbitrary IPv4 unicast address used for the test (must be different than value of test variable IP\_ADDRESS\_1)  **T\_SUBSCRIPTION**: (Uint8) Lifetime of Eventgroup subscription used for the test. Must be at least 3 seconds. |
| Test Procedure | 1. TESTER: Sends multicast SOME/IP-SD message from IP Endpoint address **IP\_ADDRESS\_1**, carrying a Find Service Entry for the DUT’s ETS instance. 2. DUT: Sends SOME/IP-SD Unicast message carrying an Offer Service Entry. 3. TESTER: Verify received SOME/IP-SD messages. 4. TESTER: Sends multicast SOME/IP-SD message from IP Endpoint address **IP\_ADDRESS\_2**, carrying a Find Service Entry for the DUT’s ETS instance. 5. DUT: Sends SOME/IP-SD Unicast message carrying an Offer Service Entry. 6. TESTER: Verify received SOME/IP-SD messages. 7. TESTER: Sends unicast SOME/IP-SD message from IP Endpoint address **IP\_ADDRESS\_1,** carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0005. The session ID is set to **SESSION\_ID\_2** and the reboot Flag = 1**.** 8. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 9. TESTER: Verify received SOME/IP-SD message. 10. TESTER: Sends SOME/IP Request-No-Return message for method **triggerEventUINT8** ETS instance **ETS\_InstanceID\_0** with the following parameters:     * **delay** is set to 0x00000000 (0 seconds)     * **duration** is set to **T\_SUBSCRIPTION**.     * **debounceTime** is set to 0x00000001 (1 second). 11. DUT: Sends Unicast SOME/IP Notification messages for **TestEventUINT8.** 12. TESTER: Verify received SOME/IP message. 13. TESTER: Sends unicast SOME/IP-SD message from IP Endpoint address **IP\_ADDRESS\_2,** carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0005. The session ID is set to **SESSION\_ID\_2** and the reboot Flag = 1**.** 14. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 15. TESTER: Verify received SOME/IP-SD message. 16. DUT: Sends Unicast SOME/IP Notification messages for **TestEventUINT8.** 17. TESTER: Verify received SOME/IP message. 18. DUT: Sends Unicast SOME/IP Notification messages for **TestEventUINT8.** 19. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP-SD message is received at IP Endpoint Address **IP\_ADDRESS\_1** within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received at IP Endpoint Address **IP\_ADDRESS\_2** within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 3. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: Eventgroup ID is equal to 0x0005. 4. SOME/IP Notification message is received at IP Endpoint Address **IP\_ADDRESS\_1** within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**). 5. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: Eventgroup ID is equal to 0x0005. 6. SOME/IP Notification message is received at IP Endpoint Address **IP\_ADDRESS\_2** within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**). 7. SOME/IP Notification message is received at IP Endpoint Address **IP\_ADDRESS\_1** within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**). |
| Reference | DS\_SD\_0189, DS\_SD\_0250, DS\_SD\_0251 |
| Notes | Two Clients with different Endpoint options, communicate with the DUT. The subscription, which is send via unicast, of the first client have a session ID greater than 1. From the second client the Subscription received later, when the first client receives events yet. It shows that there are different Session ID counters, because the DUT still sends continuously events to the first client, without a reboot was detected. |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_26] Ignore undefined bits in SD message header

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT Ignore undefined bits in the flag and the reserved bits in SD message header |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 unicast address used as source address of the simulated SOME/IP client for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends multicast SOME/IP-SD message with all undefined bits in Flag and Reserved fields changed to 1, carrying a Find Service Entry for the DUT’s ETS instance **ETS\_InstanceID\_0**. IP Source Address is set to **IP\_ADDRESS**. 4. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutResponse** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: TTL is equal to **ETS\_Lifetime**.    5. IP Destination Address is equal to **IP\_ADDRESS**. |
| Reference | DS\_SD\_0032, DS\_SD\_0033 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_27] Ignore not required options in Find Service

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT Ignore not required options in Find Service. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS\_1**: IPv4 unicast address used as source address of the simulated SOME/IP client for the test.  **IP\_ADDRESS\_2**: Arbitrary IPv4 unicast address used for the test. Must be different than value of test variable IP\_ADDRESS\_1.  **SD\_PortNumber**: Port used for SOME/IP-SD |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s ETS instance **ETS\_InstanceID\_0** which references the following option in addition:    * Type is set to 0x04 (IPv4 Endpoint Option).    * Discardable Flag is set to 0.    * IPv4 Address is set to **IP\_ADDRESS\_2**.    * L4-Proto is set to 0x11 (UDP).    * Port Number is set to **SD\_PortNumber**. 4. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to ETS\_InstanceID\_0.    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutResponse** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to ETS\_InstanceID\_0.    4. Entry: TTL is equal to **ETS\_Lifetime**.    5. IP Destination Address is equal to **IP\_ADDRESS\_1**.    6. Discardable Flag is equal to 0.    7. UDP Destination Port Number is equal to **SD\_PortNumber**. |
| Reference | DS\_SD\_0114, DS\_SD\_0122 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_28] Answer Find Service with setup from IPv4 SD Endpoint Option

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT Answer a Find Service with a setup from IPv4 SD Endpoint Option. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 address used in the SD Endpoint Option for the test.  **PORT\_NUMBER**: (Uint16) Number of the UDP port used in the SD Endpoint Option for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s instance **ETS\_InstanceID\_0**. Options Array includes the following option in addition:    * Type is set to 0x24 (IPv4 SD Endpoint Option).    * Discardable Flag is set to 0.    * IPv4 Address is set to **IP\_ADDRESS**.    * L4-Proto is set to 0x11 (UDP).    * Port Number is set to **PORT\_NUMBER**. 4. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to ETS\_InstanceID\_0.    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutResponse** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to ETS\_InstanceID\_0.    4. Entry: TTL is equal to **ETS\_Lifetime**.    5. IP Destination Address is equal to **IP\_ADDRESS**.    6. Discardable Flag is equal to 0.    7. UDP Destination Port Number is equal to **PORT\_NUMBER**. |
| Reference | DS\_SD\_0087, DS\_SD\_0088 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_29] Stop Offer Service shall announce that service is no longer offered

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a SOME/IP Service Discovery instance sents Stop Offer Service Entries for all of the services provided by this instance. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SERVICE\_DOWN**: (Uint8) Duration in seconds [sec] the ETS shall be suspended. Must be at least as long as the configured CYCLIC\_OFFER\_DELAY. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **suspendInterface** instance **ETS\_InstanceID\_0** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**. 2. DUT: Stops offering the ETS. 3. DUT: Sends multicast SOME/IP-SD message carrying a Stop Offer Service Entry. 4. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStopOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to ETS\_InstanceID\_0.    4. Entry: TTL is equal to 0x000000 (0 seconds). |
| Reference | DS\_SD\_0128, DS\_SD\_0181, DS\_SD\_0184 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_30] Stop Offer Service is sent for all services if ECU is being shut down

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Besides the default Instance, an alternative instance at the same ECU are running. Before the DUT is shutdown, it will send a Stop Offer Service for all service instances. |
| Preconditions | DUT is running and offering the Enhanced Testability Service in two instances which only differs by its Major version.  Method **serviceActivate** with **instanceKey** set to 0x03 (alternative major version) may be called for this test case. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s ETS with **ETS\_MajorVersion\_0.** 2. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 3. TESTER: Verify received SOME/IP-SD messages. 4. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s ETS instance **ETS\_InstanceID\_0** with **ETS\_MajorVersion\_1.** 5. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 6. TESTER: Verify received SOME/IP-SD messages. 7. TESTER: Sends SOME/IP Request-No-Return message for method **shutdownECU** of ETS instance **ETS\_InstanceID\_0** for Instance Awith the following parameters:    * **delay** is set to 0x02 (2seconds). 8. DUT: Sends multicast SOME/IP-SD message carrying a Stop Offer Service Entry for the DUT’s ETS instance **ETS\_InstanceID\_0** with **ETS\_MajorVersion\_0**. 9. TESTER: Verify received SOME/IP-SD message. 10. DUT: Sends multicast SOME/IP-SD message carrying a Stop Offer Service Entry for the DUT’s ETS instance **ETS\_InstanceID\_0** with **ETS\_MajorVersion\_1**. 11. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Major version is equal to **ETS\_MajorVersion\_0.**    5. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: TTL is equal to **ETS\_Lifetime**.    5. Entry: Major version is equal to **ETS\_MajorVersion\_1.** 3. SOME/IP-SD message for is received before **TimeoutStopOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: Major version is equal to **ETS\_MajorVersion\_0.**    5. Entry: TTL is equal to 0x000000 (0 seconds). 4. SOME/IP-SD message for is received before **TimeoutStopOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: Major version is equal to **ETS\_MajorVersion\_1.**    5. Entry: TTL is equal to 0x000000 (0 seconds). |
| Reference | DS\_SD\_0188 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_31] Ignore Find Service with unknown Service ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT ignore a Find Service with unknown Service ID. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UNKNOWN\_SERVICE\_ID**: (Uint16) Service ID which is unknown to DUT.  **IP\_ADDRESS**: IPv4 unicast address used as source address of the simulated SOME/IP client for the test.  **T\_CYCLIC\_OFFER\_DELAY**: (Uint8) The value of SOME/IP parameter CYCLIC\_OFFER\_DELAY which is used by the DUT. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s instance **ETS\_InstanceID\_0** with Service ID changed to **UNKNOWN\_SERVICE\_ID**. IP Source Address is set to **IP\_ADDRESS**. 4. TESTER: Observes further DUT communication until **T\_CYCLIC\_OFFER\_DELAY** is elapsed. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: TTL is equal to **ETS\_Lifetime**.    5. IP Destination Address is equal to **IP\_ADDRESS**. |
| Reference | DS\_SD\_0194, DS\_SD\_0195, DS\_SD\_0196 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_32] Ignore Find Service with unknown Instance ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT ignore a Find Service with an unknown Instance ID |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UNKNOWN\_INSTANCE\_ID**: (Uint8) An ETS Instance ID which is unknown to DUT.  **IP\_ADDRESS**: IPv4 unicast address used as source address of the simulated SOME/IP client for the test.  **T\_CYCLIC\_OFFER\_DELAY**: (Uint8) The value of SOME/IP parameter CYCLIC\_OFFER\_DELAY which is used by the DUT. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s instance **ETS\_InstanceID\_0** with Instance ID changed to **UNKNOWN\_INSTANCE\_ID**. IP Source Address is set to **IP\_ADDRESS**. 4. TESTER: Observes further DUT communication until **T\_CYCLIC\_OFFER\_DELAY** is elapsed. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: TTL is equal to **ETS\_Lifetime**.    5. IP Destination Address is equal to **IP\_ADDRESS**. |
| Reference | DS\_SD\_0194, DS\_SD\_0195, DS\_SD\_0196 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_33] Ignore Find Service with wrong Major Version

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT ignore a Find Service with a wrong Major Version. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UNKNOWN\_MAJOR\_VERSION**: (Uint8) An ETS major version which is unknown to DUT.  **IP\_ADDRESS**: IPv4 unicast address used as source address of the simulated SOME/IP client for the test.  **T\_CYCLIC\_OFFER\_DELAY**: (Uint8) The value of SOME/IP parameter CYCLIC\_OFFER\_DELAY which is used by the DUT. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s ETS instance with Major Version changed to **UNKNOWN\_MAJOR\_VERSION**. IP Source Address is set to **IP\_ADDRESS**. 4. TESTER: Observes further DUT communication until **T\_CYCLIC\_OFFER\_DELAY** is elapsed. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: TTL is equal to **ETS\_Lifetime**.    5. IP Destination Address is equal to **IP\_ADDRESS**. |
| Reference | DS\_SD\_0194, DS\_SD\_0195, DS\_SD\_0196 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_34] Ignore Find Service when Session ID is set to zero

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT ignore a Find Service when the Session ID is set to zero. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 unicast address used as source address of the simulated SOME/IP client for the test.  **T\_CYCLIC\_OFFER\_DELAY**: (Uint8) The value of SOME/IP parameter CYCLIC\_OFFER\_DELAY which is used by the DUT. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends multicast SOME/IP-SD message with Session ID changed to 0x0000 and carrying a Find Service Entry for the DUT’s instance **ETS\_InstanceID\_0**. 4. TESTER: Observes further DUT communication until **T\_CYCLIC\_OFFER\_DELAY** is elapsed. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: TTL is equal to **ETS\_Lifetime**.    5. IP Destination Address is equal to **IP\_ADDRESS**. |
| Reference | DS\_SD\_0013 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_35] Ignore Find Service if Unicast flag is set to zero

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT ignore a Find Service if an Unicast flag is set to zero. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 unicast address used as source address of the simulated SOME/IP client for the test.  **T\_CYCLIC\_OFFER\_DELAY**: (Uint8) The value of SOME/IP parameter CYCLIC\_OFFER\_DELAY which is used by the DUT. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends multicast SOME/IP-SD message with Unicast Flag changed to 0 and carrying a Find Service Entry for the DUT’s instance **ETS\_InstanceID\_0**. 4. TESTER: Observes further DUT communication until **T\_CYCLIC\_OFFER\_DELAY** is elapsed. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: TTL is equal to **ETS\_Lifetime**.    5. IP Destination Address is equal to **IP\_ADDRESS**. |
| Reference | DS\_SD\_0029 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_36] Service is offered once per Major Version on same ECU

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Besides the default Instances, an alternative instance at the same ECU are running. For every instance of the same service, which differs only  in its major version, the DUT is expected to send an Offer Service Entry. |
| Preconditions | DUT is running and offering the Enhanced Testability Service in two instances which only differs by its Major version.  Method **serviceActivate** with **instanceKey** set to 0x03 (alternative major version) may be called for this test case. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s ETS instance **ETS\_InstanceID\_0** with **ETS\_MajorVersion\_0.** 2. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 3. TESTER: Verify received SOME/IP-SD messages. 4. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s ETS instance **ETS\_InstanceID\_0** with **ETS\_MajorVersion\_1.** 5. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 6. TESTER: Verify received SOME/IP-SD messages. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0.**    3. Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Major version is equal to **ETS\_MajorVersion\_0.**    5. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0.**    3. Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: Major version is equal to **ETS\_MajorVersion\_1.**    5. Entry: TTL is equal to **ETS\_Lifetime**. |
| Reference | DS\_SD\_0253 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_37] Server SD Entries has same Service and Instance IDs but different Major Versions

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Besides the default Instance an alternative instance at the same ECU are running. For every Service Instance the server SD Entries after stop and reoffer the service the Entries has the same Service and Instance ID, but a different Major Version. |
| Preconditions | DUT is running and offering the Enhanced Testability Service in two instances which only differs by its Major version.  Method **serviceActivate** with **instanceKey** set to 0x03 (alternative major version) may be called for this test case. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SERVICE\_DOWN**: (Uint8) Duration in seconds [sec] the ETS shall be suspended. Must be at least as long as the configured CYCLIC\_OFFER\_DELAY. |
| Test Procedure | 1. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s ETS with **ETS\_MajorVersion\_0.** 2. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 3. TESTER: Verify received SOME/IP-SD messages. 4. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for the DUT’s ETS with **ETS\_MajorVersion\_1.** 5. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 6. TESTER: Verify received SOME/IP-SD messages. 7. TESTER: Sends SOME/IP Request-No-Return message for method **suspendInterface** ofinstance **ETS\_InstanceID\_0** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**.   ETS service for **ETS\_MajorVersion\_0** is addressed.   1. DUT: Sends multicast SOME/IP-SD message carrying a Stop Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD messages. 3. TESTER: Sends SOME/IP Request-No-Return message for method **suspendInterface** ofinstance **ETS\_InstanceID\_0** withthe following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**.   ETS service for **ETS\_MajorVersion\_1** is addressed.   1. DUT: DUT: Sends multicast SOME/IP-SD message carrying a Stop Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Major version is equal to **ETS\_MajorVersion\_0.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Major version is equal to **ETS\_MajorVersion\_1.**    4. Entry: TTL is equal to **ETS\_Lifetime**. 3. SOME/IP-SD message for is received before **TimeoutStopOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**    4. Entry: Major version is equal to **ETS\_MajorVersion\_0.**    5. Entry: TTL is equal to 0x000000 (0 seconds). 4. SOME/IP-SD message for is received before **TimeoutStopOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**    4. Entry: Major version is equal to **ETS\_MajorVersion\_1.**    5. Entry: TTL is equal to 0x000000 (0 seconds). |
| Reference | DS\_SD\_0248 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_38] Server forward Request to correct Service instance

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to forward a SOME/IP request to the correct Service instance, which is addressed by the client. Therefore the DUT runs two instances of the ETS which differs only in its Major version. The Request is sends to only one of the instances. To verify that the Request is processed by correct service instance, value of field Interface version returned by the corresponding Response is evaluated. The field interface version always provides the major version of service instance where it is located. |
| Preconditions | DUT is running and offering the Enhanced Testability Service in two instances which only differs by its Major version.  Method **serviceActivate** with **instanceKey** set to 0x03 (alternative major version) may be called for this test case. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for getter of ETS field **InterfaceVersion** of ETS instance **ETS\_InstanceID\_0** instance with **ETS\_MajorVersion\_0.** 2. DUT: Sends corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP response message. 4. TESTER: Sends SOME/IP Request message for getter of ETS field **InterfaceVersion** of ETS instance **ETS\_InstanceID\_1** instance with **ETS\_MajorVersion\_1.** 5. DUT: Sends corresponding SOME/IP Response message. 6. TESTER: Verify received SOME/IP response message. |
| Verification | 1. SOME/IP response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Instance ID is equal to **ETS\_InstanceID\_0**.    3. Method ID is equal to0x25 (Field Interfaceversion).    4. Return Code is equal to 0x00 (E\_OK).    5. Payload provides an Uint8 value equal to **ETS\_MajorVersion\_0.** 2. SOME/IP response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Instance ID is equal to **ETS\_InstanceID\_0**.    3. Method ID is equal to0x25 (Field Interfaceversion).    4. Return Code is equal to 0x00 (E\_OK).    5. Payload provides an Uint8 value equal to **ETS\_MajorVersion\_1.** |
| Reference | DS\_SD\_0249 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_39] Server ignores request if service lifetime has been expired

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to ignore incoming Requests if lifetime of addressed Service has been expired. No Response or error message shall be sent in reaction to the Request. |
| Preconditions | DUT is running and offering the Enhanced Testability. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **T\_SERVICE\_DOWN**: (Uint8) Duration in seconds [sec] the ETS shall be suspended. Must be at least as long as 2 times **ETS\_Lifetime.** |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends SOME/IP Request-No-Return message for method **suspendInterface** instance **ETS\_InstanceID\_0** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SERVICE\_DOWN**. 4. DUT: Stops offering the ETS. 5. TESTER: Observes further DUT communication for a period of **ETS\_Lifetime**. 6. TESTER: Sends Some/IP Request for method **echoUINT8** with the following parameters:    * **value** is set to **UINT8\_VALUE**. 7. TESTER: Observes further DUT communication for a period of **TimeoutResponse**. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: TTL is equal to **ETS\_Lifetime.** 2. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: TTL is equal to **ETS\_Lifetime.** 3. No SOME/IP messages were observed which contain:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x08 (**echoUINT8**) |
| Reference | DS\_SD \_0124 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_40] Offer multiple service instances if client asks for any Major Version

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to answer a Find Service Entry, which is looking for a service without addressing any specific major version, by sending a SOME/IP-SD message with Offer Service Entries for all hosted major versions of the addressed service. |
| Preconditions | DUT is running and offering the Enhanced Testability Service in two instances which only differs by its Major Version.  Method **serviceActivate** with **instanceKey** set to 0x03 (alternative major version) may be called for this test case. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | none |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying Offer Service Entries. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for Service ID **ETS\_ServiceID\_0** with Major Version is set to 0xFF (any version). 4. DUT: Sends unicast SOME/IP-SD message carrying Offer Service Entries. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides two entries of type 0x01 (Offer Service Entry).    2. All entries: Service ID is equal to **ETS\_ServiceID\_0**.    3. All entries: Instance ID is equal to **ETS\_InstanceID\_0**.    4. All entries: TTL is equal to **ETS\_Lifetime**.    5. One entry: Major Version is equal to **ETS\_MajorVersion\_0**.    6. One entry: Major Version is equal to **ETS\_MajorVersion\_1**. 2. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides two entries of type 0x01 (Offer Service Entry).    2. All entries: Service ID is equal to **ETS\_ServiceID\_0**.    3. All entries: Instance ID is equal to **ETS\_InstanceID\_0**.    4. All entries: TTL is equal to **ETS\_Lifetime**.    5. One entry: Major Version is equal to **ETS\_MajorVersion\_0**.    6. One entry: Major Version is equal to **ETS\_MajorVersion\_1**. |
| Reference | DS\_SD\_0120 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_41] Offer single service instance if client asks for specific Major Version

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to answer a Find Service Entry, which is looking for a service of a specific major version, by sending a SOME/IP-SD message with a single Offer Service Entry for the addressed service version only. |
| Preconditions | DUT is running and offering the Enhanced Testability Service in two instances which only differs by its Major version.  Method **serviceActivate** with **instanceKey** set to 0x03 (alternative major version) may be called for this test case. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | none |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying Offer Service Entries. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for Service ID **ETS\_ServiceID\_0** with Major Version set to **ETS\_MajorVersion\_1**. 4. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides two entries of type 0x01 (Offer Service Entry).    2. All entries: Service ID is equal to **ETS\_ServiceID\_0**.    3. All entries: Instance ID is equal to **ETS\_InstanceID\_0**.    4. All entries: TTL is equal to **ETS\_Lifetime**.    5. One entry: Major Version is equal to **ETS\_MajorVersion\_0**.    6. One entry: Major Version is equal to **ETS\_MajorVersion\_1**. 2. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides only one entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: TTL is equal to **ETS\_Lifetime**.    5. Entry: Major Version is equal to **ETS\_MajorVersion\_1**. |
| Reference | DS\_SD\_0120 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_42] Offer multiple service instances if client asks for any Minor Version

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to answer a Find Service Entry, which is looking for a service without addressing any specific minor version, by sending a SOME/IP-SD message with Offer Service Entries for all hosted minor versions of the addressed service. |
| Preconditions | DUT is running and offering the Enhanced Testability Service in two instances which only differs by its Minor Version.  Method **serviceActivate** with **instanceKey** set to 0x04 (alternative minor version) may be called for this test case. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | none |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying Offer Service Entries. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for Service ID **ETS\_ServiceID\_0** with Minor Version set to 0xFFFF FFFF (any version). 4. DUT: Sends unicast SOME/IP-SD message carrying Offer Service Entries. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides two entries of type 0x01 (Offer Service Entry).    2. All entries: Service ID is equal to **ETS\_ServiceID\_0**.    3. All entries: Instance ID is equal to **ETS\_InstanceID\_0**.    4. All entries: Major Version is equal to **ETS\_MajorVersion\_0**.    5. All entries: TTL is equal to **ETS\_Lifetime**.    6. One entry: Minor Version is equal to **ETS\_MinorVersion\_0**.    7. One entry: Minor Version is equal to **ETS\_MinorVersion\_1**. 2. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides two entries of type 0x01 (Offer Service Entry).    2. All entries: Service ID is equal to **ETS\_ServiceID\_0**.    3. All entries: Instance ID is equal to **ETS\_InstanceID\_0**.    4. All entries: Major Version is equal to **ETS\_MajorVersion\_0**.    5. All entries: TTL is equal to **ETS\_Lifetime**.    6. One entry: Minor Version is equal to **ETS\_MinorVersion\_0**.    7. One entry: Minor Version is equal to **ETS\_MinorVersion\_1**. |
| Reference | DS\_SD\_0120 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_43] Offer single service instance if client asks for specific Minor Version

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to answer a Find Service Entry, which is looking for a service of a specific minor version, by sending a SOME/IP-SD message with a single Offer Service Entry for the addressed service version only. |
| Preconditions | DUT is running and offering the Enhanced Testability Service in two instances which only differs by its Minor Version.  Method **serviceActivate** with **instanceKey** set to 0x04 (alternative minor version) may be called for this test case. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | none |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying Offer Service Entries. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for Service ID **ETS\_ServiceID\_0** with Minor Version set to **ETS\_MinorVersion\_1**. 4. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides two entries of type 0x01 (Offer Service Entry).    2. All entries: Service ID is equal to **ETS\_ServiceID\_0**.    3. All entries: Instance ID is equal to **ETS\_InstanceID\_0**.    4. All entries: TTL is equal to **ETS\_Lifetime**.    5. One entry: Minor Version is equal to **ETS\_MinorVersion\_0**.    6. One entry: Minor Version is equal to **ETS\_MinorVersion\_1**. 2. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides only one entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: TTL is equal to **ETS\_Lifetime**.    5. Entry: Minor Version is equal to **ETS\_MinorVersion\_1**. |
| Reference | DS\_SD\_0120 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_44] Offer multiple service instances if client asks for any Instance ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to answer a Find Service Entry, which is looking for a service without addressing any specific Instance ID, by sending a SOME/IP-SD message with Offer Service Entries for all hosted instances of the addressed service. |
| Preconditions | DUT is running and offering the Enhanced Testability Service in two instances which only differs by its Instance ID.  Method **serviceActivate** with **instanceKey** set to 0x02 (alternative instance) may be called for this test case. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | none |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying Offer Service Entries. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for Service ID **ETS\_ServiceID\_0** with Instance ID set to 0xFFFF (any version). 4. DUT: Sends unicast SOME/IP-SD message carrying Offer Service Entries. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides two entries of type 0x01 (Offer Service Entry).    2. All entries: Service ID is equal to **ETS\_ServiceID\_0**.    3. All entries: TTL is equal to **ETS\_Lifetime**.    4. One entry: Instance ID is equal to **ETS\_InstanceID\_0**.    5. One entry: Instance ID is equal to **ETS\_InstanceID\_1**. 2. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides two entries of type 0x01 (Offer Service Entry).    2. All entries: Service ID is equal to **ETS\_ServiceID\_0**.    3. All entries: TTL is equal to **ETS\_Lifetime**.    4. One entry: Instance ID is equal to **ETS\_InstanceID\_0**.    5. One entry: Instance ID is equal to **ETS\_InstanceID\_1**. |
| Reference | DS\_SD\_0120 |
| Notes |  |

#### [SOMEIPSRV\_SD\_BEHAVIOR\_45] Offer single service instance if client asks for specific Instance ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to answer a Find Service Entry, which is looking for a service with a specific Instance ID, by sending a SOME/IP-SD message with a single Offer Service Entry for the addressed service instance only. |
| Preconditions | DUT is running and offering the Enhanced Testability Service in two instances which only differs by its Instance ID.  Method **serviceActivate** with **instanceKey** set to 0x02 (alternative instance) may be called for this test case. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | none |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying Offer Service Entries. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends multicast SOME/IP-SD message carrying a Find Service Entry for Service ID **ETS\_ServiceID\_0** with Instance ID set to **ETS\_InstanceID\_1**. 4. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides two entries of type 0x01 (Offer Service Entry).    2. All entries: Service ID is equal to **ETS\_ServiceID\_0**.    3. All entries: TTL is equal to **ETS\_Lifetime**.    4. One entry: Instance ID is equal to **ETS\_InstanceID\_0**.    5. One entry: Instance ID is equal to **ETS\_InstanceID\_1**. 2. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides only one entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_1**.    4. Entry: TTL is equal to **ETS\_Lifetime**. |
| Reference | DS\_SD\_0120 |
| Notes |  |

### Service Discovery Publish/Subscribe Communication

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_01] ACK is sent for accepted Subscribe Eventgroup

|  |  |
| --- | --- |
| tem | Description |
| Purpose | Checks that a that a valid Subscribe Eventgroup Entry is answered with an appropriate Subscribe Eventgroup Acknowledgement. Else a Subscribe Eventgroup Acknowledgement Entry uses the same values as the one used by the Subscribe Eventgroup Entry that is answered. The following fields are affected:   * Service ID * Instance ID * Major Version * TTL * Eventgroup ID * Counter * Initial Data Requested Flag |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0005. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received via unicast UDP within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Service ID is equal to **ETS\_ServiceID\_0**.    3. Instance ID is same as in the preceding Subscribe Eventgroup Entry.    4. Major Version is same as in the preceding Subscribe Eventgroup Entry.    5. TTL is same as in the preceding Subscribe Eventgroup Entry.    6. Eventgroup ID is equal to Eventgroup ID 0x0005.    7. Counter is same as in the preceding Subscribe Eventgroup Entry.    8. Initial Data Requested Flag is same as in the preceding Subscribe Eventgroup Entry. |
| Reference | DS\_SD\_0136, DS\_SD\_0137 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_02] ACK reference Multicast Option when subscribed event is transported via multicast

|  |  |
| --- | --- |
| tem | Description |
| Purpose | Checks that a Subscribe Eventgroup Acknowledgement Entry references an IPv4 Multicast Option when notifications of the subscribed event are transported via multicast. The Multicast Option provides information where the notifications will be sent to. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for the ETS. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0006. 4. DUT: Sends unicast SOME/IP-SD message carrying corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received via unicast within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0006.    4. Subscribe Eventgroup Ack Entry references one option of type 0x14 (IPv4 Multicast Option). |
| Reference | DS\_SD\_0138, DS\_SD\_0241 |
| Notes | For ETS event TestEventUINT8Multicast the DUT provides a respective IPv4 Multicast Option. |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_03] ACK shall be transported by unicast

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the SOME/IP-SD message, which is carrying a Subscribe Eventgroup Acknowledgement Entry, is transported by unicast. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0005. 4. DUT: Sends unicast SOME/IP-SD message carrying corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received via unicast UDP within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0005.    4. Entry: TTL is same as in the preceding Subscribe Eventgroup Entry.    5. Destination IP of received SD message is equal to **TESTER\_IPv4Address**. |
| Reference | DS\_SD\_0203 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_04] NACK is sent for Subscribe Eventgroup with unknown Service ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a Subscribe Eventgroup Entry with an unknown Service ID is answered with an appropriate Subscribe Eventgroup Negative Acknowledgement Entry. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UNKNOWN\_SERVICE\_ID**: (Uint16) Service ID which is unknown to DUT. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0005 with a modified Instance ID set to **UNKNOWN\_SERVICE\_ID**. 4. DUT: Sends unicast SOME/IP-SD message carrying a corresponding Subscribe Eventgroup Negative Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **UNKNOWN\_SERVICE\_ID**.    3. Entry: Eventgroup ID is equal to 0x0005.    4. Entry: TTL is equal to 0x000000. |
| Reference | DS\_SD\_0139, DS\_SD\_0141, DS\_SD\_0196 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_05] NACK is sent for Subscribe Eventgroup with unknown Eventgroup

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a Subscribe Eventgroup Entry with an unknown Eventgroup ID is answered with an appropriate Subscribe Eventgroup Negative Acknowledgement Entry. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UNKNOWN\_EVENTGROUP\_ID**: (Uint16) Eventgroup ID which is unknown to the respective SOME/IP service. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry with a modified eventgroup ID set to **UNKNOWN\_EVENTGROUP\_ID**. 4. DUT: Sends unicast SOME/IP-SD message carrying a corresponding Subscribe Eventgroup Negative Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to **UNKNOWN\_EVENTGROUP\_ID**.    4. Entry: TTL is equal to 0x000000. |
| Reference | DS\_SD\_0139, DS\_SD\_0141, DS\_SD\_0196 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_06] NACK is sent for Subscribe Eventgroup with unknown Instance ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a Subscribe Eventgroup Entry with an unknown Instance ID is answered with an appropriate Subscribe Eventgroup Negative Acknowledgement Entry. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UNKNOWN\_INSTANCE\_ID**: (Uint16) Instance ID of the respective SOME/IP service which is unknown to DUT. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0005 with a modified Instance ID set to **UNKNOWN\_INSTANCE\_ID**. 4. DUT: Sends unicast SOME/IP-SD message carrying a corresponding Subscribe Eventgroup Negative Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within TimeoutOfferService and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **UNKNOWN\_INSTANCE\_ID**.    4. Entry: TTL is equal to 0x000000. |
| Reference | DS\_SD\_0139, DS\_SD\_0141, DS\_SD\_0196 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_07] NACK is sent for Subscribe Eventgroup with unknown Major Version

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a Subscribe Eventgroup Entry with an unknown Major Version is answered with an appropriate Subscribe Eventgroup Negative Acknowledgement Entry. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UNKNOWN\_MAJOR\_VERSION** : (Uint8) Major Version of the respective SOME/IP service which is unknown to DUT. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0005 with a modified Instance ID set to **UNKNOWN\_MAJOR\_VERSION**. 4. DUT: Sends unicast SOME/IP-SD message carrying a corresponding Subscribe Eventgroup Negative Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Service ID is equal to **ETS\_ServiceID\_0**.    3. Major Version is equal to **UNKNOWN\_MAJOR\_VERSION**.    4. TTL is equal to 0x000000. |
| Reference | DS\_SD\_0139, DS\_SD\_0141, DS\_SD\_0196 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_08] NACK is sent for Subscribe Eventgroup without required TCP connection

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a Subscribe Eventgroup Entry is answered with an appropriate Subscribe Eventgroup Negative Acknowledgement Entry when a required TCP connection was not opened by client. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 address where the SOME/IP client is reachable.  **PORT\_NUMBER**: (Uint16) Number of the TCP port where the SOME/IP client is listening for events of subscribed Eventgroup. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry which references an IPV4 endpoint option with following parameters:    * **ip\_address** is set to **IP\_ADDRESS**.    * **portNumber** is set to **PORT\_NUMBER**.    * Discardable Flag is set to 0 2. TESTER: Receive SOME/IP-SD message 3. TESTER: Does not open the required TCP connection on port number **PORT\_NUMBER** announced by the received Offer Service Entry. 4. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 5. DUT: Sends unicast SOME/IP-SD message carrying a corresponding Subscribe Eventgroup Negative Acknowledgement Entry. 6. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is equal to 0x000000. |
| Reference | DS\_SD\_0139, DS\_SD\_0141, DS\_SD\_0196 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_09] NACK is sent for Subscribe Eventgroup without any Endpoint Options

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a Subscribe Eventgroup Entry without any IPv4 Endpoint Options is answered with an appropriate Subscribe Eventgroup Negative Acknowledgement Entry. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0005 without any referenced options. 4. DUT: Sends unicast SOME/IP-SD message carrying a corresponding Subscribe Eventgroup Negative Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Service ID is equal to **ETS\_ServiceID\_0**.    3. Eventgroup ID is equal to Eventgroup ID 0x0005.    4. TTL is equal to 0x000000. |
| Reference | DS\_SD\_0117, DS\_SD\_0139, DS\_SD\_0141, DS\_SD\_0196, DS\_SD\_0197 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_10] NACK is sent for Subscribe Eventgroup with invalid Endpoint Option

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a Subscribe Eventgroup Entry with an invalid IPv4 Endpoint Option (unknown value for Layer 4 Protocol) is answered with an appropriate Subscribe Eventgroup Negative Acknowledgement Entry. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UNKNOWN\_L4\_PROTO**: (Uint8) IANA/IETF number of a transport protocol which is not supported by SOME/IP. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0005 which references the following option in addition:    * Endpoint Option: L4-Proto is set to **UNKNOWN\_L4\_PROTO**. 4. DUT: Sends unicast SOME/IP-SD message carrying a corresponding Subscribe Eventgroup Negative Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Service ID is equal to **ETS\_ServiceID\_0**.    3. Eventgroup ID is equal to ID 0x0005.    4. TTL is equal to 0x000000. |
| Reference | DS\_SD\_0139, DS\_SD\_0141, DS\_SD\_0196, DS\_SD\_0197 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_11] NACK is sent for Subscribe Eventgroup with conflict in Endpoint Options

|  |  |
| --- | --- |
| Item | Description |
| Purpose | NACK is sent for Subscribe Eventgroup with conflict in Endpoint Options. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **WRONG\_IP\_ADDRESS**: IPv4 Address different than TesterDefaultIPv4Address.  **WRONG\_PORT\_NUMBER**: (Uint8) UDP port number different than TesterDefaultPortUDP. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0005 that references the following option in addition:    * Type is set to 0x04 (IPv4 Endpoint Option).    * Discardable Flag is set to 0.    * L4-Proto is set to 0x11 (UDP).    * IPv4 Address is set to **WRONG\_IP\_ADDRESS**.    * Port Number is set to **WRONG\_PORT\_NUMBER**. 4. DUT: Sends unicast SOME/IP-SD message carrying an Eventgroup Entry of type 0x07. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0005.    4. Entry: TTL is equal to 0x0000. |
| Reference | DS\_SD\_0116, DS\_SD\_0197 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_12] NACK is sent for Subscribe Eventgroup with unknown Option

|  |  |
| --- | --- |
| Item | Description |
| Purpose | NACK is sent for Subscribe Eventgroup with unknown Option. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UNKNOWN\_OPTION\_TYPE**: (Uint8) SOME/IP option type which is unknown to DUT. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0005 that references the following option in addition:    * Type is set to **UNKNOWN\_OPTION\_TYPE**.    * Length is set to 0x0005.    * Discardable Flag is set to 0.    * All remaining bits are set to 0. 4. DUT: Sends unicast SOME/IP-SD message carrying an Eventgroup Entry of type 0x07. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0005.    4. Entry: TTL is equal to 0x0000. |
| Reference | DS\_SD\_0113 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_13] NACK shall be transported by unicast

|  |  |
| --- | --- |
| Item | Description |
| Purpose | NACK shall be transported by unicast. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UNKNOWN\_SERVICE\_ID**: (Uint16)Service ID which is unknown to DUT. |
| Test Procedure | 1. DUT: Sends Unicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0005 and Service ID is set to **UNKNOWN\_SERVICE\_ID**. 4. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Negative Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received via unicast UDP within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **UNKNOWN\_SERVICE\_ID**.    3. Entry: Eventgroup ID is equal to 0x0005.    4. Entry: TTL is equal to 0x0000.    5. Destination IP of received SD message is equal to **TESTER\_IPv4Address**. |
| Reference | DS\_SD\_0203 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_14] Ignore unknown option in Subscribe Eventgroup

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Ignore unknown option in Subscribe Eventgroup. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UNKNOWN\_OPTION\_TYPE**: (Uint8) Type of option which is unknown to the DUT. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message with an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0006. In addition, the entry shall reference the following option:    * Type is set to **UNKNOWN\_OPTION\_TYPE**.    * Length is set to 0x0005.    * Discardable Flag is set to 1.    * All remaining bits are set to 0. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Eventgroup ID is equal to 0x0006.    3. Entry: TTL is not equal to 0x0000 |
| Reference | DS\_SD\_0074, DS\_SD\_0113 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_15] Ignore redundant options of Subscribe Eventgroup

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Ignore redundant options of Subscribe Eventgroup. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message with an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0006. In addition, the entry shall reference the following option:  * Entry: Shall reference a second IPv4 Endpoint Option which is a duplicate of the first one.  1. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 2. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Eventgroup ID is equal to 0x0006.    3. Entry: TTL is not equal to 0x0000. |
| Reference | DS\_SD\_0115 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_16] Process Configuration Option referenced by Subscribe Eventgroup

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to deserialize content of Configuration Options and forward it to the application. In this test, the ETS stores the content for further verification into a field. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING**: Zero-terminated character string which represents a valid SOME/IP-SD Configuration Option string with arbitrary key/value-pairs.  **STRING\_LENGTH**: (Uint32) Length of configuration string. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0006. In addition the entry shall reference the following option:    * Type is set to 0x01 (Configuration Option).    * Length is set to **STRING\_LENGTH** + 1.    * Discardable Flag is set to 0.    * Reserved bits are set to 0.    * Configuration String is set to **UTF8\_STRING**. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Eventgroup ID is equal to 0x0006.    3. Entry: TTL is not equal to 0x0000.    4. Payload provides an UTF8 string equal to **UTF8\_STRING**. |
| Reference | DS\_SD\_0102 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_17] Process Subscribe entries from same message in order they arrived

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Process Subscribe entries from same message in order they arrived. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **EVENT\_TIME\_SLOT:** (Uint16) Period of time until then the DUT shall send out an event notification. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message with an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with the following entries in the prescribed order:    * Stop Subscribe Eventgroup Entry for Eventgroup ID 0x0006.    * Subscribe Eventgroup Entry for Eventgroup ID 0x0006. 4. DUT: Sends unicast SOME/IP-SD message with the corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8Multicast** with the following parameters**:**    * **delay** is set to **0x00000000.**    * **duration** is setto **EVENT\_TIME\_SLOT.**    * **debounceTime** is set to **EVENT\_TIME\_SLOT.** 7. DUT: Sends SOME/IP Notification message for Event **TestEventUINT8Multicast.** 8. TESTER: Verify SOME/IP Notification message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Eventgroup ID is equal to 0x0006.    3. Entry: TTL is not equal to 0x000000. 3. SOME/IP message is received within **EVENT\_TIME\_SLOT** and contains:    1. Service ID is equal to to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x800B (**TestEventUINT8Multicast**). |
| Reference | DS\_SD\_0035 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_18] Server send Event Notifications if a client has subscribed

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT sends Event Notifications if a client has subscribed. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **EVENT\_TIME\_SLOT**: (Uint16) Period of time until then the DUT shall send out an event notification. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to **EVENT\_TIME\_SLOT**.    * **debounceTime** is set to **EVENT\_TIME\_SLOT**. 7. DUT: Sends SOME/IP message with Message Type set to 0x02. 8. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Eventgroup ID is equal to 0x0002.    3. Entry: TTL is not equal to 0x0000. 3. SOME/IP message is received within **EVENT\_TIME\_SLOT** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Message Type is equal to 0x02 (NOTIFICATION). |
| Reference | DS\_SP\_0027, DS\_SP\_0091, DS\_SD\_0201 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_19] Server create proper header for Notification (Message ID / Request ID)

|  |  |
| --- | --- |
| Item | Description |
| Purpose | DUT is expected to create proper header for a SOME/IP Notification message. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **EVENT\_TIME\_SLOT**: (Uint16) Period of time until then the DUT shall send out an event notification. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying with Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to **EVENT\_TIME\_SLOT**.    * **debounceTime** is set to **EVENT\_TIME\_SLOT**. 7. DUT: Sends SOME/IP message with Message Type set to 0x02. 8. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Eventgroup ID is equal to 0x0002.    3. Entry: TTL is not equal to 0x0000. 3. SOME/IP message is received within **EVENT\_TIME\_SLOT** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**).    3. Client ID is equal to 0x0000.    4. Interface Version is equal to **ETS\_MajorVersion\_0**.    5. Message Type is equal to 0x02 (NOTIFICATION).    6. Return Code is equal to 0x00 (E\_OK). |
| Reference | DS\_ID\_0011, DS\_ID\_0012, DS\_SP\_0017, DS\_SP\_0024, DS\_SP\_0025, DS\_SP\_0029, DS\_SP\_0030, DS\_SP\_0092 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_20] Server send Event Notification to proper client address

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT send Event Notification to proper client address. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **EVENT\_TIME\_SLOT**: (Uint16) Period of time until then the DUT shall send out an event notification. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to **EVENT\_TIME\_SLOT**.    * **debounceTime** is set to **EVENT\_TIME\_SLOT**. 7. DUT: Sends SOME/IP Notification message for **TestEventUINT8**. 8. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Eventgroup ID is equal to 0x0002.    3. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received within **EVENT\_TIME\_SLOT** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**).    3. IP Destination Address is equal to **Tester\_IPv4Address**.    4. TP Destination Port is equal to **Tester\_PortUDP**. |
| Reference | DS\_SD\_0109, DS\_SD\_0111 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_21] Server send Event Notification from announced TCP unicast Endpoint

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT send Event Notification from announced TCP unicast Endpoint. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 address where the SOME/IP client is reachable.  **PORT\_NUMBER**: (Uint16) Number of the TCP port where the SOME/IP client is listening for events of subscribed Eventgroup.  **EVENT\_TIME\_SLOT**: (Uint16) Period of time until then the DUT shall send out an event notification. |
| Test Procedure | 1. DUT: Sends unicast SOME/IP-SD message carrying an Offer Service Entry which reference an IPv4 Endpoint Option for **IP\_ADDRESS** and **PORT\_NUMBER.** 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8Reliable** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to **EVENT\_TIME\_SLOT**.    * **debounceTime** is set to **EVENT\_TIME\_SLOT**. 7. DUT: Sends SOME/IP Notification message for **TestEventUINT8Reliable**. 8. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Eventgroup ID is equal to 0x0002.    3. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received from source address **IP\_ADDRESS** andport **PORT\_NUMBER** within **EVENT\_TIME\_SLOT** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8003 (**TestEventUINT8Reliable**).    3. IP Source Address is equal to the one which is announced by preceding Offer Service Entry.    4. TCP Source Port is equal to the one which is announced by preceding Offer Service Entry. |
| Reference | DS\_SD\_0229, DS\_SD\_0231, DS\_SD\_0235, DS\_SD\_0236 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_22] Server send Event Notification from announced UDP Endpoint

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT send Event Notification from announced UDP Endpoint. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **EVENT\_TIME\_SLOT**: (Uint16) Period of time until then the DUT shall send out an event notification. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0005. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to **EVENT\_TIME\_SLOT**.    * **debounceTime** is set to **EVENT\_TIME\_SLOT**. 7. DUT: Sends SOME/IP Notification message for **TestEventUINT8**. 8. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Eventgroup ID is equal to 0x0005.    3. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received within **EVENT\_TIME\_SLOT** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**).    3. IP Source Address is equal to the one which is announced by preceeding Offer Service Entry.    4. UDP Source Port is equal to the one which is announced by preceeding Offer Service Entry. |
| Reference | DS\_SD\_0077, DS\_SD\_0229, DS\_SD\_0230, DS\_SD\_0235, DS\_SD\_0236 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_23] Server send Event Notifications in a cyclic manner

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT sends Event Notifications in a cyclic manner. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT provides a build-in mechanism for cyclic sending of SOME/IP Event Notifications. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **EVENT\_CYCLE\_TIME**: Period of time afterwards the DUT shall send out an event notification. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerCyclicEventuINT8** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to **EVENT\_CYCLE\_TIME**    * **debounceTime** is set to **EVENT\_CYCLE\_TIME**. 7. DUT: Sends SOME/IP Notification message for **TestEventUINT8Cyclic**. 8. TESTER: Verify received SOME/IP message. 9. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerCyclicEventuINT8** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to **EVENT\_CYCLE\_TIME**.    * **debounceTime** is set to **EVENT\_CYCLE\_TIME**. 10. DUT: Sends SOME/IP Notification message for **TestEventUINT8Cyclic**. 11. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Eventgroup ID is equal to 0x0002.    3. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received within **EVENT\_CYCLE\_TIME** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8502 (**TestEventUINT8Cyclic**). 4. SOME/IP Notification message is received after **EVENT\_CYCLE\_TIME** +/- **TimingTolerance** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8502 (**TestEventUINT8Cyclic**). |
| Reference | DS\_SP\_0095 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_24] Server send Event Notifications on change

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT sends Event Notifications on change. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT provides a build-in mechanism for sending SOME/IP Event Notifications on change only. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **EVENT\_TIME\_SLOT**: (Uint16) Period of time until then the DUT shall send out an event notification.  **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for ETS method **setEventUINT8Change** with the following parameters:    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Notification message for **TestEventUINT8Change**. 8. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Eventgroup ID is equal to 0x0002.    3. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received within **EVENT\_TIME\_SLOT** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8503 (**TestEventUINT8Change**). |
| Reference | DS\_SP\_0095 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_25] Server shall not send initial value of event when client has subscribed

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT shall not send initial value of event when client has subscribed. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **OBSERVATION\_TIME**: (Uint16) Period of time where the DUT shall not send any event notifications. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Observes further DUT communication for a period of **OBSERVATION\_TIME**. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Eventgroup ID is equal to 0x0002.    3. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification messages with Method ID equal to 0x8001. (**TestEventUINT8**) are not received. |
| Reference | DS\_SD\_0215 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_26] Server sends only one Event notification to multiple Clients with same IP-Address

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT sends only one Event notification to multiple Clients with same IP-Address. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Lifetime of Eventgroup subscription used for the test. Shall be at least 3 seconds. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying the following entries:    * Subscribe Eventgroup Entry for Eventgroup ID 0x0006 with TTL set to **T\_SUBSCRIPTION** and Counter field set to 0x1.    * Subscribe Eventgroup Entry for Eventgroup ID 0x0006 with TTL set to **T\_SUBSCRIPTION** and Counter field set to 0x2. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entries. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8Multicast** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to 0x00000001.    * **debounceTime** is set to 0x00000001. 7. DUT: Sends SOME/IP Notification message for **TestEventUINT8Multicast**. 8. TESTER: Verify received SOME/IP Notification message. 9. TESTER: Observes further DUT communication for a period of 1 second. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0006.    4. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x800B (**TestEventUINT8Multicast**). 4. No SOME/IP Notification messages were observed which contain:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x800B (**TestEventUINT8Multicast**). |
| Reference | DS\_SP\_0093 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_27] Server shall not send duplicated events to same client

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to avoid sending of a duplicated event notification to same client when two eventgroups has been subscribed at the same time which includes the identical event. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for the ETS. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 4. DUT: Sends unicast SOME/IP-SD message carrying the corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0005. 7. DUT: Sends unicast SOME/IP-SD message carrying the corresponding Subscribe Eventgroup Acknowledgement Entry. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to 0x00000001 (1 second).    * **debounceTime** is set to 0x00000002 (2 seconds). 10. DUT: Sends SOME/IP Notification message. 11. TESTER: Verify received SOME/IP Notification message. 12. TESTER: Observes further DUT communication until a period of 1 second is elapsed. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is not equal to 0x0000. 3. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0005.    4. Entry: TTL is not equal to 0x0000. 4. SOME/IP Notification message is received within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**).    3. Message Type is equal to 0x02 (NOTIFICATION). 5. No SOME/IP Notification messages were observed which contain:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**).    3. Message Type is equal to 0x02 (NOTIFICATION). |
| Reference | DS\_SP\_0094, DS\_SD\_0200 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_28] Server send initial Field Value if client has subscribed

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT has to answer the Testers SubscribeEvengroup Message with a SubscribeEventgroupAck and immediately after send out the corresponding Initial Fields for the Subscribed Eventgroup. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT must support Explicit Initial Data Control functionality. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. DUT: Sends SOME/IP Notification message for field **TestFieldUINT8**. 7. TESTER: Verify received SOME/IP Notification message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received within **TimeoutInitialEvents** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8006 (**TestFieldUINT8**).    3. Message Type is equal to 0x02 (NOTIFICATION). |
| Reference | DS\_SP\_0096, DS\_SP\_0100, DS\_SD\_0216 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_29] Server send initial Field Value after Subscribe Eventgroup ACK

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT send initial Field Value after Subscribe Eventgroup ACK. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT must support Explicit Initial Data Control functionality. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0002 with Initial Data Requested Flag set to 1. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. DUT: Sends SOME/IP Notification message for field **TestFieldUINT8**. 7. TESTER: Verify received SOME/IP Notification message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received within **TimeoutInitialEvents** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8006 (**TestFieldUINT8**).    3. Message Type is equal to 0x02 (NOTIFICATION). |
| Reference | DS\_SD\_0212, DS\_SD\_0218 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_30] Initial Event is sent as unicast

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT Initial Event is sent as unicast. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT must support Explicit Initial Data Control functionality. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. DUT: Sends unicast SOME/IP Notification message for field **TestFieldUINT8**. 7. TESTER: Verify received SOME/IP Notification message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received within **TimeoutInitialEvents** and contains:    1. IP Destination Address is not a multicast address.    2. Service ID is equal to **ETS\_ServiceID\_0**.    3. Method ID is equal to 0x8006 (**TestFieldUINT8**).    4. Message Type is equal to 0x02 (NOTIFICATION). |
| Reference | DS\_SD\_0240 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_31] Initial Event is sent only once

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT Initial Event is send only once, else in this Testcase the TTL should not be expired. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT must support Explicit Initial Data Control functionality. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. DUT: Sends SOME/IP Notification message for field **TestFieldUINT8**. 7. TESTER: Verify received SOME/IP Notification message. 8. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 9. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 10. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 11. TESTER: Verify received SOME/IP-SD message. 12. TESTER: Observes further DUT communication for a period of **TimeoutInitialEvents**. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received within **TimeoutInitialEvents** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8006 (**TestFieldUINT8**). 4. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is not equal to 0x0000. 5. SOME/IP Notification messages with Method ID equal to 0x8006 (**TestFieldUINT8**) are not received. |
| Reference | DS\_SD\_0217 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_32] Initial Event is sent for each subscription of same Client

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT Initial Event is send for each subscription of same Client. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT must support Explicit Initial Data Control functionality. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. DUT: Sends SOME/IP Notification message for field **TestFieldUINT8**. 7. TESTER: Verify received SOME/IP Notification message. 8. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 9. TESTER: Verify received SOME/IP-SD message. 10. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0005. 11. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 12. TESTER: Verify received SOME/IP-SD message. 13. DUT: Sends SOME/IP Notification message for field **TestFieldUINT8**. 14. TESTER: Verify received SOME/IP Notification message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received within **TimeoutInitialEvents** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8006 (**TestFieldUINT8**). 4. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 5. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0005.    4. Entry: TTL is not equal to 0x0000. 6. SOME/IP Notification message is received within **TimeoutInitialEvents** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8006 (**TestFieldUINT8**). |
| Reference | DS\_SD\_0220 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_33] Initial Event is sent even when Explicit Initial Data Control Flag is not set

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT Initial Event is send even when Explicit Initial Data Control Flag is not set. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT must support Explicit Initial Data Control functionality. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Explicit Initial Data Control Flag set to zero, carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. DUT: Sends SOME/IP Notification message for field **TestFieldUINT8**. 7. TESTER: Verify received SOME/IP Notification message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received within **TimeoutInitialEvents** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8006 (**TestFieldUINT8**). |
| Reference | DS\_SD\_0211, DS\_SD\_0218 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_34] Initial Event is not sent when client set Initial Data Requested Flag to zero

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT Initial Event is not send when client set Initial Data Requested Flag to zero. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT must support Explicit Initial Data Control functionality. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables |  |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0002 with Initial Data Requested Flag set to zero. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Observes further DUT communication for a period of **TimeoutInitialEvents**. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification messages with Method ID equal to 0x8006. (**TestFieldUINT8**) are not received. |
| Reference | DS\_SD\_0213 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_35] Server send Field Value on change

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT send Field Value on change. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary field value. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. DUT: Sends SOME/IP Notification message for field **TestFieldUINT8**. 7. TESTER: Verify received SOME/IP Notification message. 8. TESTER: Sends SOME/IP Request message for setter of field **TestFieldUINT8** with payload set to **UINT8\_VALUE**. 9. DUT: Sends the corresponding SOME/IP Response message. 10. TESTER: Verify received SOME/IP Response message. 11. DUT: Sends SOME/IP Notification message for field **TestFieldUINT8**. 12. TESTER: Verify received SOME/IP Notification message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received within **TimeoutInitialEvents** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8006 (**TestFieldUINT8**). 4. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0027 (setter of **TestFieldUINT8**).    3. Result Code is equal to 0x00 (E\_OK). 5. SOME/IP Notification message is received within **TimeoutInitialEvents** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8006 (**TestFieldUINT8**). |
| Reference | DS\_SP\_0096, DS\_SP\_0101 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_36] Subscription shall be deleted if Stop Subscribe Eventgroup is received

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a Stop Subscribe Eventgroup Entry instructs an immediately unsubscription of an already subscribed Eventgroup. In consequence no more notification events related to the respective Eventgroup shall be sent to client. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **TTL\_SUBSCRIBE**: (Uint24) Subscription’s lifetime used for the test. Must be at least 2 seconds. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0002 and TTL value set to **TTL\_SUBSCRIBE**. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to **TTL\_SUBSCRIBE**.    * **debounceTime** is set 0x00000001 (1 second). 7. DUT: Sends SOME/IP Notification message for field **TestEventUINT8**. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends unicast SOME/IP-SD message carrying a Stop Subscribe Eventgroup Entry. 10. TESTER: Observes further DUT communication for a period of **TTL\_SUBSCRIBE**. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is equal to **TTL\_SUBSCRIBE**. 3. SOME/IP Notification message is received within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**).    3. Message Type is equal to 0x02 (NOTIFICATION). 4. SOME/IP Notification messages with Method ID equal to 0x8001 (**TestEventUINT8**) are not received. |
| Reference | DS\_SD\_0133 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_37] Subscription shall be deleted if TTL expires

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that expiration of lifetime (TTL) will lead to an immediately unsubscription of an already subscribed Eventgroup. In consequence no more notification events related to the respective Eventgroup shall be sent to client. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Lifetime of Eventgroup subscription used for the test. Shall be at least 3 seconds.  **TTL\_SUBSCRIBE** : (Uint24) Subscription’s lifetime used for the test. Has to be longer than update interval of the above selected event, but must not be 0xFFFFFF. (Must be less than T\_SUBSCRIPTION) |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0005 and a TTL value set to **TTL\_SUBSCRIBE**. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Send unicast SOME/IP Request-No-Return message for method **triggerEventUINT8** with the following parameters:  * **delay** is set to 0x00 (0 seconds) * **duration** is set to service **T\_SUBSCRIPTION**. * **debounceTime** is set to 0x01 (1 seconds)  1. DUT: Sends cyclically SOME/IP notification messages for field **TestEventUINT8** and a TTL value set to **TTL\_SUBSCRIBE**. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Observes further DUT communication for a period of **T\_SUBSCRIPTION**. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0005.    4. Entry: TTL is equal to **TTL\_SUBSCRIBE**. 3. First SOME/IP Notification message is received within **TimeoutInitialEvents** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**).    3. Message Type is equal to 0x02 (NOTIFICATION). 4. At least SOME/IP Notification messages with Method ID equal to 0x8001 (**TestEventUINT8**) are not received. |
| Reference | DS\_SD\_0131 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_38] Subscription shall be deleted if TCP connection to client is lost

|  |  |
| --- | --- |
| tem | Description |
| Purpose | Checks that a Subscription will be delete, if the TCP connection to the Client lost on the non Physical Layer. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **EVENT\_TIME\_SLOT**: (Uint16) Period of time until then the DUT shall send out an event notification. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8Reliable** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to 5 times **EVENT\_TIME\_SLOT**.    * **debounceTime** is set to **EVENT\_TIME\_SLOT**. 7. DUT: Sends SOME/IP Notification message for **TestEventUINT8Reliable**. 8. TESTER: Verify received SOME/IP message. 9. TESTER: Simulates loss of TCP connection by ignoring any incoming TCP segments for a period of 2 times **EVENT\_TIME\_SLOT**. 10. DUT: Deletes subscription for Eventgroup ID 0x0002. 11. TESTER: Resumes to normal answer behavior on the TCP connection. 12. TESTER: Observes further DUT communication for a period of 2 times **EVENT\_TIME\_SLOT**. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Eventgroup ID is equal to 0x0002.    3. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received within **EVENT\_TIME\_SLOT** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8003 (**TestEventUINT8Reliable**). 4. SOME/IP Notification messages with Method ID equal to 0x8003 (**TestEventUINT8Reliable**) are not received. |
| Reference | DS\_SD\_202 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_39] Subscription shall be deleted if TCP connection to client is refused

|  |  |
| --- | --- |
| tem | Description |
| Purpose | Checks that a Subscription will be delete, if the TCP connection from the Server will be refused by a client, with a specify answer. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **EVENT\_TIME\_SLOT**: (Uint16) Period of time until then the DUT shall send out an event notification. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8Reliable** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to 5 times **EVENT\_TIME\_SLOT**.    * **debounceTime** is set to **EVENT\_TIME\_SLOT**. 7. DUT: Sends SOME/IP Notification message for **TestEventUINT8Reliable**. 8. TESTER: Verify received SOME/IP message. 9. TESTER: Refuses any incoming TCP segments for a period of 2 times **EVENT\_TIME\_SLOT**. 10. DUT: Deletes subscription for Eventgroup ID 0x0002. 11. TESTER: Resumes to normal answer behavior on the TCP connection. 12. TESTER: Observes further DUT communication for a period of 2 times **EVENT\_TIME\_SLOT**. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Eventgroup ID is equal to 0x0002.    3. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received within **EVENT\_TIME\_SLOT** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8003 (**TestEventUINT8Reliable**). 4. SOME/IP Notification messages with Method ID equal to 0x8003 (**TestEventUINT8Reliable**) are not received. |
| Reference | DS\_SD\_0202 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_40] Subscription shall be deleted if Service is no longer offered

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that when a service is no longer offered from the DUT. In consequence no more notification events related to the respective Eventgroup shall be sent to client. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Lifetime of Eventgroup subscription used for the test. Shall be at least 3 seconds. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0002 and TTL set to **T\_SUBSCRIPTION**. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to **T\_SUBSCRIPTION**.    * **debounceTime** is set to 0x00000001. 7. DUT: Sends SOME/IP Notification message for **TestEventUINT8**. 8. TESTER: Verify received SOME/IP Notification message. 9. TESTER: Sends SOME/IP Request-No-Return message for ETS method **suspendInterface** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to **T\_SUBSCRIPTION**. 10. TESTER: Observes further DUT communication for a period of **T\_SUBSCRIPTION**. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**). 4. SOME/IP Notification messages with Method ID equal to 0x8001 (**TestEventUINT8**) are not received. |
| Reference | DS\_SP\_0184 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_41] Subscription shall be deleted if Reboot of client is detected

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that when the DUT detect that a client reboot over the reboot flag. In consequence no more notification events related to the respective Eventgroup shall be sent to client. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Lifetime in seconds of Eventgroup subscription used for the test. Must be at least 3 seconds. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0005 and TTL set to **T\_SUBSCRIPTION**. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **debounceTime** is set to 0x00000001 (1 second). 7. DUT: Sends SOME/IP Notification message for **TestEventUINT8**. 8. TESTER: Verify received SOME/IP Notification message. 9. TESTER: Sends multicast SOME/IP-SD message carryling a Find Service Entry for **ETS\_ServiceID\_0**. 10. TESTER: Sends multicast SOME/IP-SD message with Reboot Flag set to 1 and Session ID set to 0x0001, carrying a Find Service Entry for **ETS\_ServiceID\_0**. 11. TESTER: Observes further DUT communication for a period of **T\_SUBSCRIPTION**. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: TTL is equal to **ETS\_Lifetime**. 2. SOME/IP-SD message is received before **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0005.    4. Entry: TTL is equal to **T\_SUBSCRIPTION**. 3. SOME/IP Notification message is received within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**).   11. No SOME/IP Notification messages were observed which contain:   * 1. Service ID is equal to **ETS\_ServiceID\_0**.   2. Method ID is equal to 0x8001 (**TestEventUINT8**). |
| Reference | DS\_SD\_0024, DS\_SD\_0025, DS\_SD\_0026 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_42] All subscriptions shall be deleted if link on Ethernet interface is lost

|  |  |
| --- | --- |
| tem | Description |
| Purpose | In case of losing its link on the relevant Ethernet interface, the DUT is expected to stop sending event notifications to all clients who has subscribed so far, even when link status becomes up again. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  (For example, **suspendEthernetInterface** method may be called for this test) |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 address used to simulate a second client hosted by another ECU. Must be different than Tester\_IPv4Address.  **T\_EVENT\_NOTIFICATIONS**: (Uint8) Duration in seconds where the DUT shall sent out event notifications. Must be at least 3 seconds. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0002 with TTL set to 0xFFFFFF. Source IP address is **Tester\_IPv4Address**. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0002 with TTL set to 0xFFFFFF. Source IP address is **IP\_ADDRESS**. 7. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8Reliable** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **duration** is set to **T\_EVENT\_NOTIFICATIONS**.    * **debounceTime** is set to 0x00000001 (1 second). 10. DUT: Sends SOME/IP Notification message for **TestEventUINT8Reliable**. 11. TESTER: Verify received SOME/IP Notification message. 12. DUT: Sends SOME/IP Notification message for **TestEventUINT8 Reliable**. 13. TESTER: Verify received SOME/IP Notification message. 14. TESTER: Forces a link loss on DUT-side for duration of 1 second. 15. TESTER: Observes further DUT communication for a period of **T\_EVENT\_NOTIFICATIONS**. 16. TESTER: Try to establish TCP reconnect by sending SYN. 17. DUT: Accepts TCP connection by sending SYN + ACK. 18. TESTER: Accepts TCP connection by sending ACK. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is equal to 0xFFFFFF. 3. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is equal to 0xFFFFFF. 4. SOME/IP Notification message is received at endpoint **Tester\_IPv4Address** within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8003 (**TestEventUINT8Reliable**). 5. SOME/IP Notification message is received at endpoint **IP\_ADDRESS** within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8003 (**TestEventUINT8Reliable**). 6. No SOME/IP Notification messages were observed which contain:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8003 (**TestEventUINT8Reliable**). 7. TCP SYN+ACK is received within **TimeoutCloseTCPConnection**. |
| Reference | DS\_SD\_0223 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_43] Stop Subscribe Eventgroup is not answered with ACK

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT shall not respond to a unicast SubscribeEventgroup Request with TTL = 0 (i.e. StopSubscribeEventgroup). with any kind of positiv or negativ Acknowledge message. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Lifetime of Eventgroup subscription used for the test. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a unicast Stop Subscribe Eventgroup Entry for Eventgroup ID 0x0002 and TTL value set to 0x0000. 4. TESTER: Observes further DUT communication for a period of **T\_SUBSCRIPTION**. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. No SOME/IP-SD messages were received within **TimeoutSubscribeAck** wich contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup ACK/Nack).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002. |
| Reference | DS\_SD\_0136, DS\_SD\_0139 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_44] Stop sending Events if the last Client unsubscribes

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT Stop sending Events if the last Client unsubscribes. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Lifetime of Eventgroup subscription used for the test. Shall be at least 3 seconds. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0002 with TTL set to **T\_SUBSCRIPTION**. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to **T\_SUBSCRIPTION**.    * **debounceTime** is set to 0x00000001. 7. DUT: Sends SOME/IP Notification message for **TestEventUINT8**. 8. TESTER: Verify received SOME/IP Notification message. 9. TESTER: Sends SOME/IP Request-No-Return message for ETS method **suspendInterface** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to **T\_SUBSCRIPTION**. 10. TESTER: Observes further DUT communication for a period of **T\_SUBSCRIPTION**. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is equal to **TTL\_SUBSCRIBE** 3. SOME/IP Notification message is received within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**). 4. No SOME/IP Notification messages were observed which contain:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**). |
| Reference | DS\_SD\_0201 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_45] Subscribe after Stop Subscribe

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Entries of the same SOME/IP-SD message are processed in order they arrive. Therefore, the DUT expected to except a new subscription for a particular eventgroup, even when a stop subscribe eventgroup for same eventgroup is received from same message. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying the following entries in given order:    * Stop Subscribe Eventgroup Entry for Eventgroup ID 0x0002.    * Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is not equal to 0x0000. |
| Reference | DS\_SD\_0035 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_46] Stop and Subscribe combination in same message triggers Initial Event again

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send initial event notification for an already subscribed field again when a Stop Subscribe Eventgroup directly followed by an Subscribe Eventgroup Entry has been received together in one SOME/IP-SD message. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | none |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. DUT: Sends SOME/IP Notification message for field **TestFieldUINT8**. 7. TESTER: Verify received SOME/IP Notification message. 8. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 9. TESTER: Verify received SOME/IP-SD message. 10. TESTER: Sends unicast SOME/IP-SD message carrying the following entries in given order:     * Stop Subscribe Eventgroup Entry for Eventgroup ID 0x0002.     * Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 11. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 12. TESTER: Verify received SOME/IP-SD message. 13. DUT: Sends SOME/IP Notification message for field **TestFieldUINT8**. 14. TESTER: Verify received SOME/IP Notification message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is not equal to 0x0000. 3. SOME/IP Notification message is received within **TimeoutInitialEvents** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8006 (**TestFieldUINT8**). 4. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 5. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is not equal to 0x0000. 6. SOME/IP Notification message is received within **TimeoutInitialEvents** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8006 (**TestFieldUINT8**). |
| Reference | DS\_SD\_0219 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_47] Subscribe Eventgroup for an already existing subscription resets TTL

|  |  |
| --- | --- |
| tem | Description |
| Purpose | The DUT is expected to continuously send notification events for full duration of a subscription, whose lifetime has been extended by reception of a following Subscribe Eventgroup Entry in the meantime. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint24) Subscription’s lifetime used for the test. Must be at least 1 second. |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message carrying an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0005 with TTL set to **T\_SUBSCRIPTION**. 4. DUT: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to 2 x **T\_SUBSCRIPTION**.    * **debounceTime** is set to 0x00000001 (1 second). 7. TESTER: Observes further DUT communication until a period of ½ **T\_SUBSCRIPTION** is elapsed. 8. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0005 with TTL set to **T\_SUBSCRIPTION**. 9. TESTER: Observes further DUT communication until a period of **T\_SUBSCRIPTION** is elapsed. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Eventgroup ID is equal to 0x0005.    3. Service ID is equal to **ETS\_ServiceID\_0**.    4. Entry: TTL is equal to **T \_SUBSCRIPTION**. 3. SOME/IP Notification messages are cyclical received every second and contain:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**).    3. Message Type is equal to 0x02 (NOTIFICATION). 4. SOME/IP Notification message are cyclical received every second and contain:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8001 (**TestEventUINT8**).    3. Message Type is equal to 0x02 (NOTIFICATION). |
| Reference | DS\_SD\_0131 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_48] Events are sent to Static config of subscribed Clients without SD

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Events are sent to static config of subscribed Clients without Service Discovery. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8Implicit** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to 0x00000001.    * **debounceTime** is set to 0x00000001. 2. DUT: Sends SOME/IP Notification message for **TestEventUINT8Implicit**. 3. TESTER: Verify received SOME/IP Notification message. |
| Verification | 1. SOME/IP Notification message is received within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8501(**TestEventUINT8Implicit**).    3. IP Destination Address is equal to **ETS\_StaticIPv4Address**.    4. TP Destination Port Number is equal to **ETS\_StaticPortUDP**. |
| Reference | DS\_SD\_0226, DS\_SD\_0227 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_49] Static Endpoint config will be overwritten if SD provides different settings

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send event notifications to an updated IP endpoint instead of the statically configured one. The new IP endpoint is announced by regular Subscribe Eventgroup Entry. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_EVENT\_OBSERVATION**: (Uint8) Duration in seconds to observe event notifications sent by DUT. Must not be zero.  **IP\_ADDRESS**: IPv4 address used as new destination address for event notifications which is announced by Service Discovery. Must be different than ETS\_StaticIPv4MulticastAddress.  **PORT\_NUMBER**: (Uint16) Number of an UDP port used as new destination port for event notifications which is announced by Service Discovery. Must be different than ETS\_StaticPortUDP. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8Implicit** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to **T\_EVENT\_OBSERVATION**.    * **debounceTime** is set to 0x00000001. 2. DUT: Sends SOME/IP Notification message for **TestEventUINT8Implicit**. 3. TESTER: Verify received SOME/IP Notification message. 4. TESTER: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0002. Referenced IPv4 Endpoint Option for UDP binding announces the address **IP\_ADDRESS:PORT\_NUMBER and** Discardable Flag is set to 0. 5. DUT: Sends unicast SOME/IP-SD message carrying corresponding Subscribe Eventgroup Acknowledgement Entry. 6. TESTER: Verify received SOME/IP-SD message. 7. DUT: Sends SOME/IP Notification message for **TestEventUINT8Implicit**. 8. TESTER: Verify received SOME/IP Notification message. |
| Verification | 1. SOME/IP Notification message is received within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8501 (**TestEventUINT8Implicit**).    3. IP Destination Address is equal to **ETS\_StaticIPv4Address**.    4. TP Destination Port Number is equal to **ETS\_StaticPortUDP** 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. Entries Array provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is same as in the preceding Subscribe Eventgroup Entry. 3. SOME/IP Notification message is received within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8501 (**TestEventUINT8Implicit**).    3. IP Destination Address is equal to **IP\_ADDRESS**.    4. TP Destination Port Number is equal to **PORT\_NUMBER**. |
| Reference | DS\_SD\_0228 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_50] Static Subscribed Clients config will be overwritten if SD provides different settings

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Static Subscribed Clients config will be overwritten if SD provides different settings. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_EVENT\_OBSERVATION**: (Uint8) Duration in seconds to observe event notifications sent by DUT. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for ETS method **triggerEventUINT8Implicit** with the following parameters:    * **delay** is set to 0x00000000.    * **duration** is set to **T\_EVENT\_OBSERVATION**.    * **debounceTime** is set to 0x00000001. 2. DUT: Sends SOME/IP Notification message for **TestEventUINT8Implicit**. 3. TESTER: Verify received SOME/IP Notification message. 4. TESTER: Sends unicast SOME/IP-SD message carrying a Stop Subscribe Eventgroup Entry for Eventgroup ID 0x0002. 5. TESTER: Observes further DUT communication for a period of **T\_EVENT\_OBSERVATION**. |
| Verification | 1. SOME/IP Notification message is received within 1 second and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8501 (**TestEventUINT8Implicit**).    3. IP Destination Address is equal to **ETS\_StaticIPv4Address**.    4. TP Destination Port Number is equal to **ETS\_StaticPortUDP**. 2. No SOME/IP Notification messages were observed which contain:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x8501 (**TestEventUINT8Implicit**).    3. IP Destination Address is equal to **ETS\_StaticIPv4Address**.    4. TP Destination Port Number is equal to **ETS\_StaticPortUDP**. |
| Reference | DS\_SD\_0228 |
| Notes |  |

#### [SOMEIPSRV\_SD\_SUBSCRIBE\_51] Process multiple Subscribe Eventgroup Entries per SOME/IP-SD message

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to accept multiple Subscribe Eventgroup Entries from a single SOME/IP-SD message, and send an appropriate acknowledgement entry for each of it. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | none |
| Test Procedure | 1. DUT: Sends multicast SOME/IP-SD message with an Offer Service Entry. 2. TESTER: Verify received SOME/IP-SD message. 3. TESTER: Sends unicast SOME/IP-SD message carrying the following entries:    * Subscribe Eventgroup Entry for Eventgroup ID 0x0002.    * Subscribe Eventgroup Entry for Eventgroup ID 0x0005.    * Subscribe Eventgroup Entry for Eventgroup ID 0x0006. 4. DUT: Sends unicast SOME/IP-SD message with the corresponding Subscribe Eventgroup Acknowledgement Entries. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x01 (Offer Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: TTL is equal to **ETS\_Lifetime.** 2. SOME/IP-SD message is received within **TimeoutSubscribeAck** and contains:    1. One Entries Array entry provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    2. One Entries Array entry provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    3. One Entries Array entry provides an entry of type 0x07 (Subscribe Eventgroup Acknowledgement Entry).    4. All entries: Service ID is equal to **ETS\_ServiceID\_0**.    5. All entries: TTL is not equal to 0x000000.    6. One entry: Eventgroup ID is equal to 0x0002.    7. One entry: Eventgroup ID is equal to 0x0005.    8. One entry: Eventgroup ID is equal to 0x0006. |
| Reference | DS\_SD\_0039 |
| Notes |  |

### CAN Message Transport

#### [SOMEIPSRV\_CAN\_01] Header format for CAN base frame

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to apply the appropriate message header format when transmitting SOME/IP message with encapsulated CAN base frame. Value of each header field will be checked. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. DUT: Sends unicast SOME/IP Notification message for a CAN Encapsulated Messages periodically which is set to **ETSCycle\_0**. 2. TESTER: Verify received SOME/IP Notification. |
| Verification | 1. SOME/IP Notification is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. Bit position 3 to 20 is equal 0.    4. The bit position 21~31 are equal to 0x052 (**CanBaseFramePeriodic\_0**).    5. Length is equal to 0x000000D.    6. Client ID is equal to 0x0000.    7. Session ID is equal to 0x0000.    8. Protocol Version is equal to 0x01    9. Interface Version is equal to 0x00    10. Entry: Type is equal to 0x02 (Notification).    11. Return Code is equal to 0x00 (E\_OK). |
| Reference | DS\_CN\_0001, DS\_CN\_0003, DS\_CN\_0004, DS\_CN\_0005, DS\_CN\_0006, DS\_CN\_0007, DS\_CN\_0008 |
| Notes | **ETSCycle\_0 = 200 ms** |

#### [SOMEIPSRV\_CAN\_02] Header format for CAN extended frame

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to apply the appropriate message header format when transmitting SOME/IP message with encapsulated CAN extended frame. Value of each header field will be checked. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. DUT: Sends unicast SOME/IP Notification message for a CAN Encapsulated Messages periodically which is set to **ETSCycle\_0**. 2. TESTER: Verify received SOME/IP Notification. |
| Verification | 1. SOME/IP Notification is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 1 (CAN Extended frame).    3. The bit position 3~31 are equal to 0x0000 005C (**CanExtendedFramePeriodic\_0**).    4. Length is equal to 0x000000D.    5. Client ID is equal to 0x0000.    6. Session ID is equal to 0x0000.    7. Protocol Version is equal to 0x01    8. Interface Version is equal to 0x00    9. Entry: Type is equal to 0x02 (Notification).    10. Return Code is equal to 0x00 (E\_OK). |
| Reference | DS\_CN\_0002, DS\_CN\_0003, DS\_CN\_0004, DS\_CN\_0005, DS\_CN\_0006, DS\_CN\_0007, DS\_CN\_0009 |
| Notes | **ETSCycle\_0 = 200 ms** |

#### [SOMEIPSRV\_CAN\_03] Dedicated ports for UDP unicast

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to use a dedicated source and destination port for all CAN Encapsulated SOME/IP messages transmitted via UDP unicast. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. DUT: Sends unicast SOME/IP Notification message for a CAN Encapsulated Messages periodically which is set to **ETSCycle\_0**. 2. TESTER: Verify received SOME/IP Notification. |
| Verification | 1. SOME/IP Notification message from UDP Source Port 0xCAC000 is received at UDP Destination Port 0xCACC (Unicast) within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x052 (**CanBaseFramePeriodic\_0**). |
| Reference | DS\_CN\_0014, DS\_CN\_0010 |
| Notes |  |

#### [SOMEIPSRV\_CAN\_04] Dedicated ports for UDP multicast

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to use a dedicated source and destination port for all CAN Encapsulated SOME/IP messages transmitted via UDP multicast. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **None** |
| Test Procedure | 1. DUT: Sends multicast SOME/IP Notification message for a CAN Encapsulated Messages periodically which is set to **ETSCycle\_0**. 2. TESTER: Verify received SOME/IP Notification. |
| Verification | 1. SOME/IP Notification message from UDP Source Port 0xCAC00 is received at UDP Destination Port 0xCACD (Multicast) within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x052 (**CanBaseFramePeriodic\_0**). |
| Reference | DS\_CN\_0014, DS\_CN\_0010 |
| Notes |  |

#### [SOMEIPSRV\_CAN\_05] Multiple messages per UDP datagram

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send multiple CAN Encapsulated SOME/IP messages, which shall be transmitted at the same time, together in a single UDP datagram. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **None** |
| Test Procedure | 1. TESTER: Waits for **CAN\_StartDelay\_1**. 2. DUT: Sends three SOME/IP Notification messages in one UDP package. 3. TESTER: Verify received SOME/IP notification messages. |
| Verification | 1. SOME/IP notification messages are received within **TimeoutPeriodicCAN** and contains:    1. All messages: Bit position 0 and 1 is equal to 1.    2. All messages: Bit position 2 is equal to 0 (CAN Base frame).    3. Message: CAN Message ID is equal to 0x053 (**CanBaseFramePeriodic\_1**).    4. Message: CAN Message ID is equal to 0x054 (**CanBaseFramePeriodic\_2**).    5. Message: CAN Message ID is equal to 0x055 (**CanBaseFramePeriodic\_3**). |
| Reference | DS\_CN\_0011 |
| Notes |  |

#### [SOMEIPSRV\_CAN\_06] “On Event” message is sent on signal change

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to spontaneous send a sequence of CAN Encapsulated SOME/IP messages when the respective signal change event for this message occurs. The event is triggered by receiving a corresponding CAN message which provides at least a change to one of the signal values that are assigned as trigger signal to the message.  In detail, the CAN Encapsulated SOME/IP message with the updated trigger signal shall be sent in a predefined number of repetitions at first. Afterwards the respective reset message shall be sent in the same number of repetitions. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Observes further DUT communication until a period of **TimeoutPeriodicCAN** has elapsed. 2. TESTER: Sends CAN message **CanBaseFrameOnChange** with the following payload:    * + **SignalTriggerEC** is set to **UINT8\_VALUE**.      + **SignalUINT32** is set to **UINT32\_VALUE**. 3. DUT: Sends unicast SOME/IP Notification messages for a CAN Encapsulated Messages. 4. TESTER: Verify received SOME/IP Notification. 5. TESTER: Waits **TimeBeweenRepetition.** 6. TESTER: Sends CAN message **CanBaseFrameOnChange** with the following payload:    * + **SignalTriggerEC** is set to **UINT8\_VALUE**.      + **SignalUINT32** is set to **UINT32\_VALUE**. 7. DUT: Sends unicast SOME/IP Notification messages for a CAN Encapsulated Messages. 8. TESTER: Verify received SOME/IP Notification. 9. TESTER: Waits **TimeBeweenRepetition.** 10. TESTER: Sends CAN message **CanBaseFrameOnChange** with the following payload:     * + **SignalTriggerEC** is set to **UINT8\_VALUE**.       + **SignalUINT32** is set to **UINT32\_VALUE**. 11. DUT: Sends unicast SOME/IP Notification messages for a CAN Encapsulated Messages. 12. TESTER: Verify received SOME/IP Notification. |
| Verification | 1. No SOME/IP messages were observed which contain:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame). 2. SOME/IP Notification is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. Entry: **SignalTriggerEC** is equal to **UINT8\_VALUE**.    4. Entry: **SignalUINT32** is equal to **UINT32\_VALUE**.    5. Entry: CAN Message ID is equal to 0x050 (**CanBaseFrameOnChange**). 3. SOME/IP Notification is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. Entry: **SignalTriggerEC** is equal to **UINT8\_VALUE**.    4. Entry: **SignalUINT32** is equal to **UINT32\_VALUE**.    5. Entry: CAN Message ID is equal to 0x050 (**CanBaseFrameOnChange**). 4. SOME/IP Notification is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. Entry: **SignalTriggerEC** is equal to **UINT8\_VALUE**.    4. Entry: **SignalUINT32** is equal to **UINT32\_VALUE**.    5. Entry: CAN Message ID is equal to 0x050 (**CanBaseFrameOnChange**). |
| Reference | DS\_CN\_0015, DS\_CN\_0016, DS\_CN\_0017, DS\_CN\_0018, DS\_CN\_0019 |
| Notes |  |

#### [SOMEIPSRV\_CAN\_07] “On Event” message is sent on signal write

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to spontaneous send a CAN Encapsulated SOME/IP message when the respective signal write event for this message occurs. The event is triggered by receiving a corresponding CAN message which provides no changes to signal values that are assigned as trigger signal to the message. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Observes further DUT communication until a period of **TimeoutPeriodicCAN** has elapsed. 2. TESTER: Sends CAN message **CanBaseFrameOnWrite** with the following payload:    * + **SignalTriggerEW** is set to **UINT8\_VALUE**.      + **SignalUINT32** is set to **UINT32\_VALUE**. 3. DUT: Sends unicast SOME/IP Notification messages for a CAN Encapsulated Messages. 4. TESTER: Verify received SOME/IP Notification. |
| Verification | 1. No SOME/IP messages were observed which contain:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame). 2. SOME/IP Notification is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. Entry: **SignalTriggerEW** is equal to **UINT8\_VALUE**.    4. Entry: **SignalUINT32** is equal to **UINT32\_VALUE**.    5. Entry: CAN Message ID is equal to 0x051 (**CanBaseFrameOnWrite**). |
| Reference | DS\_CN\_0015, DS\_CN\_0016, DS\_CN\_0020 |
| Notes |  |

#### [SOMEIPSRV\_CAN\_11] “Periodic” message is sent cyclically

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to repeatedly send a CAN Encapsulated SOME/IP message according to its predefined cycle time and independent from receiving the respective CAN messages. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Observes further DUT communication until a period of **TimeoutPeriodicCAN** has elapsed. 2. DUT: Sends unicast SOME/IP Notification message for a CAN Encapsulated Messages periodically which is set to **ETSCycle\_0**. 3. TESTER: Verify received SOME/IP Notification. 4. TESTER: Waits nominal **of ETSCycle\_0.** 5. DUT: Sends unicast SOME/IP Notification message for a CAN Encapsulated Messages periodically which is set to **ETSCycle\_1**. 6. TESTER: Verify received SOME/IP Notification. |
| Verification | 1. No SOME/IP messages were observed which contain:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame). 2. SOME/IP Notification is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. Periodically Cycle is equal to **ETSCycle\_0**.    4. CAN Message ID is equal to 0x052 (**CanFramePeriodic\_0**). 3. SOME/IP Notification is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. Periodically Cycle is equal to **ETSCycle\_1**.    4. CAN Message ID is equal to 0x053 (**CanFramePeriodic\_1**). |
| Reference | DS\_CN\_0021, DS\_CN\_0022 |
| Notes |  |

#### [SOMEIPSRV\_CAN\_12] “Periodic and On Event” message is sent on signal change

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send cyclically as well as spontaneous a CAN Encapsulated SOME/IP message depending on the predefined cycle time and the respective signal change event for this message. The event is triggered by receiving a corresponding CAN message which provides at least a change to one of the signal values that are assigned as trigger signal to the message.  In detail, the CAN Encapsulated SOME/IP message with the updated trigger signal shall be sent in a predefined number of repetitions at first. Afterwards the respective reset message shall be sent in the same number of repetitions.  Overlapping of messages is not considered by this test case. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Observes further DUT communication until a period of **TimeoutPeriodicCAN** has elapsed. 2. DUT: Sends unicast SOME/IP Notification message for a CAN Encapsulated Messages within **ETSCycle\_0**. 3. TESTER: Verify received SOME/IP Notification. 4. TESTER: Sends CAN message **CanBaseFramePeriodicAndOnChange** with the following payload:    * + **SignalTriggerEC** is set to **UINT8\_VALUE**.      + **SignalUINT32** is set to **UINT32\_VALUE**. 5. DUT: Sends unicast SOME/IP Notification messages for a CAN Encapsulated Messages periodically within **ETSCycle\_0**. 6. TESTER: Verify received SOME/IP Notification. 7. TESTER: Waits **TimeBeweenRepetition.** 8. TESTER: Sends CAN message **CanBaseFramePeriodicAndOnChange** with the following payload:    * + **SignalTriggerEC** is set to **UINT8\_VALUE**.      + **SignalUINT32** is set to **UINT32\_VALUE**. 9. DUT: Sends unicast SOME/IP Notification messages for a CAN Encapsulated Messages periodically within **ETSCycle\_0**. 10. TESTER: Verify received SOME/IP Notification. 11. TESTER: Waits **TimeBeweenRepetition.** 12. TESTER: Sends CAN message **CanBaseFramePeriodicAndOnChange** with the following payload:     * + **SignalTriggerEC** is set to **UINT8\_VALUE**.       + **SignalUINT32** is set to **UINT32\_VALUE**. 13. DUT: Sends unicast SOME/IP Notification messages for a CAN Encapsulated Messages periodically within **ETSCycle\_0**. 14. TESTER: Verify received SOME/IP Notification. 15. Waits nominal **of ETSCycle\_0.** 16. DUT: Sends unicast SOME/IP Notification message for a CAN Encapsulated Messages within **ETSCycle\_1**. 17. TESTER: Verify received SOME/IP Notification. |
| Verification | 1. No SOME/IP messages were observed which contain:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame). 2. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x052 (**CanBaseFramePeriodic\_0**).    4. Periodically Cycle is equal to **ETSCycle\_0**. 3. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x056 (**CanBaseFramePeriodicAndOnChange**).    4. CAN Message ID is equal to 0x052 (**CanBaseFramePeriodic\_0**).    5. Entry: **SignalTriggerEC** is equal to **UINT8\_VALUE**.    6. Entry: **SignalUINT32** is equal to **UINT32\_VALUE**.    7. Periodically Cycle is equal to **ETSCycle\_0**. 4. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x056 (**CanBaseFramePeriodicAndOnChange**).    4. CAN Message ID is equal to 0x052 (**CanBaseFramePeriodic\_0**).    5. Entry: **SignalTriggerEC** is equal to **UINT8\_VALUE**.    6. Entry: **SignalUINT32** is equal to **UINT32\_VALUE**.    7. Periodically Cycle is equal to **ETSCycle\_0**. 5. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x056 (**CanBaseFramePeriodicAndOnChange**).    4. Entry: **SignalTriggerEC** is equal to **UINT8\_VALUE**.    5. Entry: **SignalUINT32** is equal to **UINT32\_VALUE**.    6. Periodically Cycle is equal to **ETSCycle\_0**. 6. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x053 (**CanBaseFramePeriodic\_1**).    4. Periodically Cycle is equal to **ETSCycle\_1**. |
| Reference | DS\_CN\_0023, DS\_CN\_0024, DS\_CN\_0025 |
| Notes |  |

#### [SOMEIPSRV\_CAN\_13] “Periodic and On Event” message is sent on signal write

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send cyclically as well as spontaneous a CAN Encapsulated SOME/IP message depending on the predefined cycle time and the respective signal write event for this message. The event is triggered by receiving a corresponding CAN message which provides no changes to signal values that are assigned as trigger signal to the message.  Overlapping of messages is not considered by this test case. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Observes further DUT communication until a period of **TimeoutPeriodicCAN** has elapsed. 2. DUT: Sends unicast SOME/IP Notification message for a CAN Encapsulated Messages within **ETSCycle\_0**. 3. TESTER: Verify received SOME/IP Notification. 4. TESTER: Sends CAN message **CanBaseFramePeriodicAndOnWrite** with the following payload:    * + **SignalTriggerEW** is set to **UINT8\_VALUE**.      + **SignalUINT32** is set to **UINT32\_VALUE**. 5. DUT: Sends unicast SOME/IP Notification messages for a CAN Encapsulated Messages periodically within **ETSCycle\_0**. 6. TESTER: Verify received SOME/IP Notification. 7. Waits nominal **of ETSCycle\_0.** 8. DUT: Sends unicast SOME/IP Notification message for a CAN Encapsulated Messages within **ETSCycle\_1**. 9. TESTER: Verify received SOME/IP Notification. |
| Verification | 1. No SOME/IP messages were observed which contain:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame). 2. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x052 (**CanBaseFramePeriodic\_0**).    4. Periodically Cycle is equal to **ETSCycle\_0**. 3. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x056 (**CanBaseFramePeriodicAndOnChange**).    4. Entry: **SignalTriggerEW** is equal to **UINT8\_VALUE**.    5. Entry: **SignalUINT32** is equal to **UINT32\_VALUE**.    6. Periodically Cycle is equal to **ETSCycle\_0**. 4. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x053 (**CanBaseFramePeriodic\_1**).    4. Periodically Cycle is equal to **ETSCycle\_1**. |
| Reference | DS\_CN\_0023, DS\_CN\_0024, DS\_CN\_0025 |
| Notes | TimeBeweenRepetition is set to 40 ms |

#### [SOMEIPSRV\_CAN\_14] Overlapping of cyclic timer and signal change event

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send only the signal change related sequence of CAN Encapsulated SOME/IP messages when a cyclic timer event and a signal change event for this message overlaps. The signal change event is triggered by receiving a corresponding CAN message which provides at least a change to one of the signal values that are assigned as trigger signal to the message.  In detail, the CAN Encapsulated SOME/IP message with the updated trigger signal shall be sent in a predefined number of repetitions at first. Afterwards the respective reset message shall be sent in the same number of repetitions. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Observes further DUT communication until a period of **TimeoutPeriodicCAN** has elapsed. 2. DUT: Sends unicast SOME/IP Notification message for a CAN Encapsulated Messages within **ETSCycle\_0**. 3. TESTER: Verify received SOME/IP Notification. 4. TESTER: Sends CAN message **CanBaseFramePeriodicAndOnChange** with the following payload:    * + **SignalTriggerEC** is set to **UINT8\_VALUE**.      + **SignalUINT32** is set to **UINT32\_VALUE**. 5. DUT: Sends unicast SOME/IP Notification messages for a CAN Encapsulated Messages periodically within **ETSCycle\_0**. 6. TESTER: Verify received SOME/IP Notification. 7. TESTER: Waits **90% of ETSCycle\_0.** 8. TESTER: Sends CAN message **CanBaseFramePeriodicAndOnChange** with the following payload:    * + **SignalTriggerEC** is set to **UINT8\_VALUE**.      + **SignalUINT32** is set to **UINT32\_VALUE**. 9. DUT: Sends unicast SOME/IP Notification messages for a CAN Encapsulated Messages periodically within **ETSCycle\_0**. 10. TESTER: Verify received SOME/IP Notification. 11. TESTER: Waits **TimeBeweenRepetition.** 12. TESTER: Sends CAN message **CanBaseFramePeriodicAndOnChange** with the following payload:     * + **SignalTriggerEC** is set to **UINT8\_VALUE**.       + **SignalUINT32** is set to **UINT32\_VALUE**. 13. DUT: Sends unicast SOME/IP Notification messages for a CAN Encapsulated Messages periodically within **ETSCycle\_1**. 14. TESTER: Verify received SOME/IP Notification. 15. Waits nominal **of ETSCycle\_0.** 16. DUT: Sends unicast SOME/IP Notification message for a CAN Encapsulated Messages within **ETSCycle\_1**. 17. TESTER: Verify received SOME/IP Notification. |
| Verification | 1. No SOME/IP messages were observed which contain:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame). 2. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x052 (**CanBaseFramePeriodic\_0**).    4. Periodically Cycle is equal to **ETSCycle\_0**. 3. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x056 (**CanBaseFramePeriodicAndOnChange**).    4. CAN Message ID is equal to 0x052 (**CanBaseFramePeriodic\_1**).    5. Periodically Cycle is equal to **ETSCycle\_0**. 4. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x056 (**CanBaseFramePeriodicAndOnChange**).    4. Entry: **SignalTriggerEC** is equal to **UINT8\_VALUE**.    5. Entry: **SignalUINT32** is equal to **UINT32\_VALUE**.    6. Periodically Cycle is equal to **ETSCycle\_0**. 5. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x056 (**CanBaseFramePeriodicAndOnChange**).    4. Entry: **SignalTriggerEC** is equal to **UINT8\_VALUE**.    5. Entry: **SignalUINT32** is equal to **UINT32\_VALUE**.    6. Periodically Cycle is equal to **ETSCycle\_1**. 6. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x053 (**CanBaseFramePeriodic\_1**).    4. Periodically Cycle is equal to **ETSCycle\_1**. |
| Reference | DS\_CN\_0026 |
| Notes | TimeBeweenRepetition is set to 40 ms |

#### [SOMEIPSRV\_CAN\_15] Overlapping of cyclic timer and signal write event

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to omit transmission of a cyclic CAN Encapsulated SOME/IP message when it overlaps with a signal write event for this message. Only the event triggered message is expected to be sent.  The signal write event is triggered by receiving a corresponding CAN message which provides no changes to signal values that are assigned as trigger signal to the message. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Observes further DUT communication until a period of **TimeoutPeriodicCAN** has elapsed. 2. DUT: Sends unicast SOME/IP Notification message for a CAN Encapsulated Messages within **ETSCycle\_0**. 3. TESTER: Waits **90% of ETSCycle\_0.** 4. TESTER: Verify received SOME/IP Notification. 5. TESTER: Sends CAN message **CanBaseFramePeriodicAndOnWrite** with the following payload:    * + **SignalTriggerEW** is set to **UINT8\_VALUE**.      + **SignalUINT32** is set to **UINT32\_VALUE**. 6. DUT: Sends unicast SOME/IP Notification messages for a CAN Encapsulated Messages periodically within **ETSCycle\_0**. 7. TESTER: Verify received SOME/IP Notification. 8. Waits nominal **of ETSCycle\_0.** 9. DUT: Sends unicast SOME/IP Notification message for a CAN Encapsulated Messages within **ETSCycle\_1**. 10. TESTER: Verify received SOME/IP Notification. |
| Verification | 1. No SOME/IP messages were observed which contain:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame). 2. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x052 (**CanBaseFramePeriodic\_0**).    4. Periodically Cycle is equal to **ETSCycle\_0**. 3. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x056 (**CanBaseFramePeriodicAndOnChange**).    4. Entry: **SignalTriggerEC** is equal to **UINT8\_VALUE**.    5. Entry: **SignalUINT32** is equal to **UINT32\_VALUE**.    6. Periodically Cycle is equal to **ETSCycle\_0**. 4. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x053 (**CanBaseFramePeriodic\_1**).    4. Periodically Cycle is equal to **ETSCycle\_1**. |
| Reference | DS\_CN\_0026 |
| Notes | TimeBeweenRepetition is set to 40 ms |

#### [SOMEIPSRV\_CAN\_16] Delay Time after cyclic message

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to delay transmission of an event triggered CAN Encapsulated SOME/IP message when occurrence took place while cyclic message of same CAN message ID has been sent shortly before. In detail, for a predefined time span after each cyclic message, transmission of respective event triggered messages shall be delayed to the end of this time span. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.) |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Observes further DUT communication until a period of **TimeoutPeriodicCAN** has elapsed. 2. DUT: Sends unicast SOME/IP Notification message for a CAN Encapsulated Messages within **ETSCycle\_0**. 3. TESTER: Verify received SOME/IP Notification. 4. TESTER: Sends CAN message **CanBaseFramePeriodicAndOnWrite** with the following payload:    * + **SignalTriggerEW** is set to **UINT8\_VALUE**.      + **SignalUINT32** is set to **UINT32\_VALUE**. 5. DUT: Sends unicast SOME/IP Notification messages for a CAN Encapsulated Messages periodically within **ETSCycle\_0**. 6. TESTER: Verify received SOME/IP Notification. 7. TESTER: Waits **T\_Delay.** 8. DUT: Sends unicast SOME/IP Notification message for a CAN Encapsulated Messages within **ETSCycle\_1**. 9. TESTER: Verify received SOME/IP Notification. |
| Verification | 1. No SOME/IP messages were observed which contain:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame). 2. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. Periodically Cycle is equal to **ETSCycle\_0.** 3. SOME/IP Notification message is received within **TimeoutTDelayCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x052 (**CanBaseFramePeriodicAndOnWrite**).    4. Entry: **SignalTriggerEW** is equal to **UINT8\_VALUE**.    5. Entry: **SignalUINT32** is equal to **UINT32\_VALUE**.    6. Periodically Cycle is equal to **ETSCycle\_0**. 4. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. Periodically Cycle is equal to **ETSCycle\_1.** |
| Reference | DS\_CN\_0034, DS\_CN\_0035 |
| Notes |  |

#### [SOMEIPSRV\_CAN\_17] Delay Time after event triggered message

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to delay transmission of an event triggered CAN Encapsulated SOME/IP message when occurrence took place while another event triggered message of same CAN message ID has been sent shortly before. In detail, for a predefined time span after each event triggered message, transmission of further event triggered messages shall be delayed to the end of this time span. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Observes further DUT communication until a period of **TimeoutDelayCAN** has elapsed. 2. DUT: Sends unicast SOME/IP Notification message for a CAN Encapsulated Messages within **ETSCycle\_0**. 3. TESTER: Verify received SOME/IP Notification. 4. TESTER: Sends CAN message **CanBaseFramePeriodicAndOnWrite** with the following payload:    * + **SignalTriggerEW** is set to **UINT8\_VALUE**.      + **SignalUINT32** is set to **UINT32\_VALUE**. 5. TESTER: Verify received SOME/IP Notification. 6. TESTER: Waits nominal **T\_Delay.** 7. TESTER: Sends CAN message **CanBaseFramePeriodicAndOnWrite** with the following payload:    * + **SignalTriggerEW** is set to **UINT8\_VALUE**.      + **SignalUINT32** is set to **UINT32\_VALUE**. 8. TESTER: Verify received SOME/IP Notification. 9. TESTER: Waits nominal **T\_Delay.** 10. TESTER: Sends CAN message **CanBaseFramePeriodicAndOnWrite** with the following payload:     * + **SignalTriggerEW** is set to **UINT8\_VALUE**.       + **SignalUINT32** is set to **UINT32\_VALUE**. 11. TESTER: Verify received SOME/IP Notification. |
| Verification | 1. No SOME/IP messages were observed which contain:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame). 2. SOME/IP Notification message is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. Periodically Cycle is equal to **ETSCycle\_0.** 3. SOME/IP Notification message is received within **TimeoutDelayCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x056 (**CanBaseFramePeriodicAndOnWrite**).    4. Entry: **SignalTriggerEW** is equal to **UINT8\_VALUE**.    5. Entry: **SignalUINT32** is equal to **UINT32\_VALUE**. 4. SOME/IP Notification message is received within **TimeoutDelayCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x052 (**CanBaseFramePeriodicAndOnWrite**).    4. Entry: **SignalTriggerEW** is equal to **UINT8\_VALUE**.    5. Entry: **SignalUINT32** is equal to **UINT32\_VALUE**. 5. SOME/IP Notification message is received within **TimeoutDelayCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. CAN Message ID is equal to 0x052 (**CanBaseFramePeriodicAndOnWrite**).    4. Entry: **SignalTriggerEW** is equal to **UINT8\_VALUE**.    5. Entry: **SignalUINT32** is equal to **UINT32\_VALUE**. |
| Reference | DS\_CN\_0034, DS\_CN\_0035 |
| Notes |  |

#### [SOMEIPSRV\_CAN\_19] Start Offset Delay of cyclic messages

|  |  |
| --- | --- |
| Item | Description |
| Purpose | After boot up, the DUT is expected to delay the initial transmission of cyclically sent CAN Encapsulated SOME/IP messages accordingly to its individual and predefined configuration of a Start Offset Delay. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT Start offset is set to **CAN\_StartDelay\_0.** |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Observes further DUT communication until a period of **CAN\_StartDelay\_0** has elapsed. 2. DUT: Sends unicast SOME/IP Notification message for a CAN Encapsulated Messages periodically which is set to **ETSCycle\_0**. 3. TESTER: Verify received SOME/IP Notification. 4. TESTER: Waits nominal **of ETSCycle\_0.** 5. DUT: Sends unicast SOME/IP Notification message for a CAN Encapsulated Messages periodically which is set to **ETSCycle\_1**. 6. TESTER: Verify received SOME/IP Notification. |
| Verification | 1. No SOME/IP messages were observed within **CAN\_StartDelay\_0** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame). 2. SOME/IP Notification is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. Periodically Cycle is equal to **ETSCycle\_0**.    4. CAN Message ID is equal to 0x052 (**CanFramePeriodic\_0**). 3. SOME/IP Notification is received within **TimeoutPeriodicCAN** and contains:    1. Bit position 0 and 1 is equal to 1.    2. Bit position 2 is equal to 0 (CAN Base frame).    3. Periodically Cycle is equal to **ETSCycle\_1**.    4. CAN Message ID is equal to 0x053 (**CanFramePeriodic\_1**). |
| Reference | DS\_CN\_0036 |
| Notes |  |

#### [SOMEIPSRV\_CAN\_20] E2E: Encode Profile 5 message

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to encode and transmit CAN Encapsulated SOME/IP messages accordingly to the AUTOSAR 4.2.2 E2E Profile 5. Format and value of the E2E header fields shall be checked. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT supports E2E Protection. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test.  **E2E\_CRC16**: The expected 16-bit CRC value of the encoded E2E data, depending on the selected values for UINT8\_VALUE and UINT32\_VALUE. |
| Test Procedure | 1. TESTER: Sends CAN message **CanBaseFrameOnWrite\_1** with the following payload:  * **SignalUINT8** is set to **UINT8\_VALUE**. * **SignalUINT32** is set to **UINT32\_VALUE**.  1. DUT: Sends unicast CAN Encapsulated SOME/IP message for **CanBaseFrameOnWrite\_1**. 2. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP message is received within **TimeoutCANMessage** and contains:    1. CAN Message ID is equal to 0x051 (**CanBaseFrameOnWrite\_1**).    2. Bit 0 to 15 of payload provides an UINT16 value equal to **E2E\_CRC16** (E2E CRC).    3. Bit 16 to 23 of payload provides an UINT8 value equal to 0x00 (E2E Alive Counter). |
| Reference | DS\_CN\_0048, DS\_CN\_0049 |
| Notes |  |

#### [SOMEIPSRV\_CAN\_21] E2E: Encode Profile 11 message

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to encode and transmit CAN Encapsulated SOME/IP messages accordingly to the AUTOSAR 4.3.1 E2E Profile 11. Format and value of the E2E header fields shall be checked. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT supports E2E Protection. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test.  **E2E\_CRC8**: The expected 8-bit CRC value of the encoded E2E data, depending on the selected values for UINT8\_VALUE and UINT32\_VALUE. |
| Test Procedure | 1. TESTER: Sends CAN message **CanBaseFrameOnWrite\_2** with the following payload:  * **SignalUINT8** is set to **UINT8\_VALUE**. * **SignalUINT32** is set to **UINT32\_VALUE**.  1. DUT: Sends unicast CAN Encapsulated SOME/IP message for **CanBaseFrameOnWrite\_2**. 2. TESTER: Verify received SOME/IP message. |
| Verification | 1. SOME/IP message is received within **TimeoutCANMessage** and contains:    1. CAN Message ID is equal to 0x052 (**CanBaseFrameOnWrite\_2**).    2. Bit 0 to 7 of payload provides an UINT8 value equal to **E2E\_CRC8** (E2E CRC).    3. Bit 12 to 15 of payload provides a 4-bit value equal to 0x00(E2E Alive Counter). |
| Reference | DS\_CN\_0048, DS\_CN\_0049 |
| Notes |  |

#### [SOMEIPSRV\_CAN\_22] E2E: Increment Alive Counter

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to initialize Alive Counter to zero for the first transmission of an appropriate CAN Encapsulated SOME/IP message and to increase it by one for every subsequent transmission. When the counter’s maximum value is reached, DUT shall reset Alive Counter to zero for the next transmission. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT supports E2E Protection. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **UINT32\_VALUE**: 32-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends CAN message **CanBaseFrameOnWrite\_2** with the following payload:  * **SignalUINT8** is set to **UINT8\_VALUE**. * **SignalUINT32** is set to **UINT32\_VALUE**.  1. DUT: Sends unicast CAN Encapsulated SOME/IP message for **CanBaseFrameOnWrite\_2**. 2. TESTER: Verify received SOME/IP message. 3. Repeat previous steps 1 to 4 for 15 times. 4. Repeat previous steps 1 to 4 once more. |
| Verification | 1. SOME/IP message is received within **TimeoutCANMessage** and contains:    1. CAN Message ID is equal to 0x052 (**CanBaseFrameOnWrite\_2**).    2. Payload provides E2E Alive Counter equal to 0x00. 2. All SOME/IP messages are received within the respective **TimeoutCANMessage** and contains:    1. CAN Message ID is equal to 0x052 (**CanBaseFrameOnWrite\_2**).    2. Payload provides E2E Alive Counter equal to value of the previously received message incremented by one. 3. SOME/IP message is received within **TimeoutCANMessage** and contains:    1. CAN Message ID is equal to 0x052 (**CanBaseFrameOnWrite\_2**).    2. Payload provides E2E Alive Counter equal to 0x00. |
| Reference | DS\_CN\_0050 |
| Notes |  |

## SOME/IP Client

### RPC Protocol Specification

#### [SOMEIPCLT\_RPC\_01] Client create proper header for Request (Message ID / Request ID)

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT has to create a proper header for the Request (Message ID / Request ID). |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8**. 8. TESTER: Verify received SOME/IP Request message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**).    3. Client ID is equal to **ETS\_ClientID**.    4. Interface Version is equal to **ETS\_MajorVersion\_0**.    5. Message Type is equal to 0x00 (REQUEST).    6. Return Code is equal to 0x00 (E\_OK). |
| Reference | DS\_ID\_0011, DS\_ID\_0012, DS\_SP\_0005, DS\_SP\_0024, DS\_SP\_0029, DS\_SP\_0030, DS\_SP\_0082, DS\_SP\_0102 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_02] Send Request to proper server address

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expect that a Request of the DUT is send to a proper UDP-IPv4 server address. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8**. 8. TESTER: Verify received SOME/IP Request message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**).    3. IP Destination Address is equal to **ETS\_RemoteIPv4Address**.    4. TP Destination Port is equal to **ETS\_RemotePort\_UDP**. |
| Reference | DS\_SD\_0109, DS\_SD\_0111 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_03] Session ID of Request is not zero

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The session ID of a SOME/IP Request of the DUT is not Zero |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8**. 8. TESTER: Verify received SOME/IP Request message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**).    3. Session ID is not equal to 0x0000. |
| Reference | DS\_SP\_0016, DS\_SP\_0018 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_04] Session ID of Request is incremented after each call

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expect a SOME/IP Request from the DUT and the session ID of a Request is incremented after each call. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8**. 8. TESTER: Verify received SOME/IP Request message. 9. TESTER: Sends corresponding SOME/IP Response message. 10. Repeat preceding steps 6 to 9 until a total number of 0xFFFE (65534) SOME/IP Request messages are processed. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**).    3. Session ID is equal to 0x0001. 3. Each SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**).    3. Session ID is incremented by one as compared to value of the preceding message. |
| Reference | DS\_SP\_0016, DS\_SP\_0020 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_05] Session ID of Request is one after wrapping

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expect that the DUT sends lots of Request Messages in order to observe the session id incrementation and wrap-around responses. After wrap-around the session id has to start from 1. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  In order to save test time, **setSessionID** may be called with sessionID close to wrap-around. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8**. 8. TESTER: Verify received SOME/IP Request message. 9. TESTER: Sends corresponding SOME/IP Response message. 10. Repeat preceding steps 6 to 9 until a total number of 0xFFFE (65534) SOME/IP Request messages are processed. 11. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:     * **delay** is set to 0x00.     * **value** is set to **UINT8\_VALUE**. 12. DUT: Sends SOME/IP Request message for method **echoUINT8**. 13. TESTER: Verify received SOME/IP Request message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**).    3. Session ID is equal to 0x0001. 3. Each SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**).    3. Session ID is incremented by one as compared to value of the preceding message. 4. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**).    3. Session ID is equal to 0x0001. |
| Reference | DS\_SP\_0016, DS\_SP\_0019 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_06] Request ID is not reused until Response has arrived or timed out

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Request ID is not reused until Response has arrived or timed out. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8**. 8. TESTER: Verify received SOME/IP Request message. 9. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 10. DUT: Sends SOME/IP Request message for method **echoUINT8**. 11. TESTER: Verify received SOME/IP Request message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**).    3. Session ID is equal to 0x0001. 3. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**).    3. Session ID is equal to 0x0002. |
| Reference | DS\_SP\_0012, DS\_SP\_0016 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_07] Client create proper order of arguments in payload for Request

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Client create proper order of arguments in payload for Request. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Bitfield of 8 bit length used as arbitrary payload for the test.  **UINT16\_VALUE**: Bitfield of 16 bit length used as arbitrary payload for the test.  **UINT32\_VALUE**: Bitfield of 32 bit length used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoBitfields** with the following parameters:    * **delay** is set to 0x00.    * **value1** is set to **UINT8\_VALUE**.    * **value2** is set to **UINT16\_VALUE**.    * **value3** is set to **UINT32\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoBitfields**. 8. TESTER: Verify received SOME/IP Request message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0041 (**echoBitfields**).    3. Payload provides an UINT8 value equal to **UINT8\_VALUE** but in reverse bit order.    4. Payload provides an UINT16 value equal to **UINT16\_VALUE** but in reverse bit order.    5. Payload provides an UINT32 value equal to **UINT32\_VALUE** but in reverse bit order. |
| Reference | DS\_SP\_0083 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_08] Client expect Response for Request

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT sends a SOME/IP Request message to the Tester and expect a Some/IP response message. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8**. 8. TESTER: Verify received SOME/IP Request message. 9. TESTER: Waits for **TimeoutResponse**. 10. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError** of ETS instance **ETS\_InstanceID\_0**. 11. DUT: Sends corresponding SOME/IP Response message. 12. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**). 3. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides UINT8 value equal to 0x06 (E\_TIMEOUT). |
| Reference | DS\_SP\_0081 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_09] Response is mapped to proper Client on same ECU via Client ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to map Response message to the correct client instance on same ECU. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSelectInstance** with parameter **instanceKey** set to 0x01 (Duplicated instance). 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 7. DUT: Activates client service instance and enters Service Discovery start-up sequence. 8. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 9. TESTER: Verify received SOME/IP-SD message. 10. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 11. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:     * **delay** is set to 0x00.     * **value** is set to **UINT8\_VALUE**. 12. DUT: Sends SOME/IP Request message for method **echoUINT8**. 13. TESTER: Verify received SOME/IP Request message. 14. TESTER: Sends corresponding SOME/IP Response message with Return Code set to 0x01 (E\_NOT\_OK). 15. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError** of ETS instance **ETS\_InstanceID\_0**. 16. DUT: Sends corresponding SOME/IP Response message. 17. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 3. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**). 4. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides UINT8 value equal to 0x01 (E\_NOT\_OK). |
| Reference | DS\_SP\_0014 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_10] Ignore Response with wrong Protocol Version

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to ignore a received SOME/IP Response message for a pending request when field Protocol Version is set to an unsupported value. The request shall be aborted by timeout instead. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test.  **WRONG\_PROTOCOL\_VERSION**: (Uint8) SOME/IP Protocol Version number which is not supported by DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8**. 8. TESTER: Verify received SOME/IP Request message. 9. TESTER: Sends corresponding SOME/IP Response message with modified Protocol Version set to **WRONG\_PROTOCOL\_VERSION**. 10. TESTER: Waits for **TimeoutResponse**. 11. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError**. 12. DUT: Sends corresponding SOME/IP Response message. 13. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**). 3. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides UINT8 value equal to 0x06 (E\_TIMEOUT). |
| Reference | DS\_SP\_0107, DS\_SP\_0113 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_11] Ignore Response with unknown Service ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to ignore a received SOME/IP Response message for a pending request when field Service ID is set to an unknown value. The request shall be aborted by timeout instead. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test.  **UNKNOWN\_SERVICE\_ID**: (Uint8) Service ID which is unknown to DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8**. 8. TESTER: Verify received SOME/IP Request message. 9. TESTER: Sends corresponding SOME/IP Response message with modified Service ID set to **UNKNOWN\_SERVICE\_ID**. 10. TESTER: Waits for **TimeoutResponse**. 11. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError** of ETS instance **ETS\_InstanceID\_0**. 12. DUT: Sends corresponding SOME/IP Response message. 13. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**). 3. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides UINT8 value equal to 0x06 (E\_TIMEOUT). |
| Reference | DS\_SP\_0107, DS\_SP\_0113 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_12] Ignore Response with wrong Interface Version

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to ignore a received SOME/IP Response message for a pending request when field Interface Version is set to a wrong value. The request shall be aborted by timeout instead. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test.  **WRONG\_INTERFACE\_VERSION**: (Uint8) Major version number of the ETS which is not supported by DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8**. 8. TESTER: Verify received SOME/IP Request message. 9. TESTER: Sends corresponding SOME/IP Response message with modified Interface Version set to **WRONG\_INTERFACE\_VERSION**. 10. TESTER: Waits for **TimeoutResponse**. 11. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError** of ETS instance **ETS\_InstanceID\_0**. 12. DUT: Sends corresponding SOME/IP Response message. 13. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**). 3. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides UINT8 value equal to 0x06 (E\_TIMEOUT). |
| Reference | DS\_SP\_0107, DS\_SP\_0113 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_13] Ignore Response with unknown Method ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to ignore a received SOME/IP Response message for a pending request when field Method ID is set to an unknown value. The request shall be aborted by timeout instead. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test.  **UNKNOWN\_METHOD\_ID**: (Uint8) Method ID which is unknown to the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8**. 8. TESTER: Verify received SOME/IP Request message. 9. TESTER: Sends corresponding SOME/IP Response message with modified Method ID set to **UNKNOWN\_METHOD\_ID**. 10. TESTER: Waits for **TimeoutResponse**. 11. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError** of ETS instance **ETS\_InstanceID\_0**. 12. DUT: Sends corresponding SOME/IP Response message. 13. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**). 3. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides UINT8 value equal to 0x06 (E\_TIMEOUT). |
| Reference | DS\_SP\_0107, DS\_SP\_0113 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_14] Ignore Response with wrong Message Type

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to ignore a received SOME/IP Response message for a pending request when field Message Type is set to a wrong value. The request shall be aborted by timeout instead. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test.  **WRONG\_MESSAGE\_TYPE**: (Uint8) Message Type different than 0x80 (RESPONSE) and 0x81 (ERROR). |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8**. 8. TESTER: Verify received SOME/IP Request message. 9. TESTER: Sends corresponding SOME/IP Response message with modified Message Type set to **WRONG\_MESSAGE\_TYPE**. 10. TESTER: Waits for **TimeoutResponse**. 11. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError** of ETS instance **ETS\_InstanceID\_0**. 12. DUT: Sends corresponding SOME/IP Response message. 13. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**). 3. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError**).    3. Return Code is equal to 0x00 (E\_OK)    4. Payload provides UINT8 value equal to 0x06 (E\_TIMEOUT). |
| Reference | DS\_SP\_0107, DS\_SP\_0113 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_15] Ignore Response with wrong Session ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to ignore a received SOME/IP Response message for a pending request when field Session ID is set to a wrong value. The request shall be aborted by timeout instead. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test.  **WRONG\_SESSION\_ID**: (Uint8) Session ID different than the one which is received by SOME/IP Request message. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8**. 8. TESTER: Verify received SOME/IP Request message. 9. TESTER: Sends corresponding SOME/IP Response message with modified Session ID set to **WRONG\_SESSION\_ID**. 10. TESTER: Waits for **TimeoutResponse**. 11. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError** of ETS instance **ETS\_InstanceID\_0**. 12. DUT: Sends corresponding SOME/IP Response message. 13. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**). 3. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides UINT8 value equal to 0x06 (E\_TIMEOUT). |
| Reference | DS\_SP\_0021 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_16] Response with Return Code different than E\_OK

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester is expected to receive an appropriate error code when for a pending request when field Return Code is set to a value different than E\_OK. The request shall be aborted by the correct error message for wrong Return Code. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test.  **RETURN\_CODE\_1**: (Uint8) Return Code different than 0x00 (E\_OK). |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8**. 8. TESTER: Verify received SOME/IP Request message. 9. TESTER: Sends corresponding SOME/IP Response message with modified Return Code set to **RETURN\_CODE\_1**. 10. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError** of ETS instance **ETS\_InstanceID\_0**. 11. DUT: Sends corresponding SOME/IP Response message. 12. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**). 3. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides UINT8 value equal to **RETURN\_CODE\_1**. |
| Reference | DS\_SP\_0107, DS\_SP\_0113 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_17] Response with non-parsable payload

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to treat response for a pending request as error, when received payload is not parsable. Error code E\_MALFORMED\_MESSAGE is expected to be returned to application in this case. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **STRING\_LENGTH**: Length of the UTF8 string value  **UTF8\_STRING**: UTF8 string with fixed length  **BROKEN\_STRING**: UTF8 string of fixed size without any termination symbol and only half of its expected length from data type definition. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUTF8DYNAMIC** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **dataLength** is set to **STRING\_LENGTH.**    * **data** is set to **UTF8\_STRING**. 7. DUT: Sends SOME/IP Request message for method **echoUTF8DYNAMIC**. 8. TESTER: Verify received SOME/IP Request message. 9. TESTER: Sends corresponding SOME/IP Response message with modified payload field set to **BROKEN\_STRING**. 10. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError** of ETS instance **ETS\_InstanceID\_0**. 11. DUT: Sends corresponding SOME/IP Response message. 12. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0015 (**echoUTF8DYNAMIC**). 3. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides UINT8 value equal to 0x09 (E\_MALFORMED\_MESSAGE). |
| Reference | DS\_SP\_0107, DS\_SP\_0113 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_18] Client ignore Response if corresponding Request is not sent completely (more than one UDP segment)

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to ignore reception of a response when the corresponding request, which is transported via SOME/IP-TP, has not yet been sent completely. After transmission of the first TP-segments, the DUT will already be confronted with an appropriate response (Return Code is E\_OK). |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT shall support message segmentation via SOME/IP-TP. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **ARRAY\_LENGTH**: Size of message payload in bytes. Must be greater than 5600 byte  **UINT8\_ARRAY**: Array of UINT8 values used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8Array** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **dataLength** is set to **ARRAY\_LENGTH**.    * **data** is set to **UINT8\_ARRAY**. 7. DUT: Sends 1st segment of SOME/IP Request message for method **echoUINT8Array**. 8. TESTER: Verify received SOME/IP-TP segment. 9. TESTER: Sends corresponding SOME/IP Response message for method **echoUINT8Array** with Return Code set to 0x00 (E\_OK). Payload provides an empty UINT8 array. 10. TESTER: Waits for **TimeoutResponse**. 11. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError** of ETS instance **ETS\_InstanceID\_0**. 12. DUT: Sends corresponding SOME/IP Response message. 13. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-TP segment is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0009 (**echoUINT8Array**).    3. Message Type is equal to 0x20 (TP\_REQUEST). 3. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides UINT8 value equal to 0x06 (E\_TIMEOUT). |
| Reference | DS\_SP\_0087 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_19] Client create proper header for Fire & Forgot Request

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expects a proper header for a F&F Request from the DUT. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **ARRAY \_LENGTH**: Length of UINT8 array value  **UINT8\_ ARRAY**: UINT8 array with fixed length |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallSetUINT8Array** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **dataLength** is set to **ARRAY\_LENGTH**.    * **data** is set to **UINT8\_ARRAY.** 7. DUT: Sends SOME/IP Request-No-Return message for method **setUINT8Array**. 8. TESTER: Verify received SOME/IP Request-No-Return message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x050C (**setUINT8Array**).    3. Client ID is equal to **ETS\_ClientID**.    4. Interface Version is equal to **ETS\_MajorVersion\_0**.    5. Message Type is equal to 0x01 (REQUEST\_NO\_RETURN).    6. Return Code is equal to 0x00 (E\_OK). |
| Reference | DS\_SP\_0024, DS\_SP\_0029, DS\_SP\_0030, DS\_SP\_0089, DS\_SP\_0102 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_20] Fire & Forget Request expect no Response

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester sends no Response for a Fire & Forget Request from the DUT. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **ARRAY \_LENGTH**: Length of UINT8 array value  **UINT8\_ ARRAY**: UINT8 array with fixed length |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallSetUINT8Array** with the following parameters:    * **delay** is set to 0x00.    * **dataLength** is set to **ARRAY\_LENGTH**    * **data** is set to **UINT8\_ARRAY**. 7. DUT: Sends SOME/IP Request message for method **setUINT8Array**. 8. TESTER: Verify received SOME/IP Request message. 9. TESTER: Waits for **TimeoutResponse**. 10. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError** of ETS instance **ETS\_InstanceID\_0**. 11. DUT: Sends corresponding SOME/IP Response message. 12. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x050C (**setUINT8Array**). 3. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides UINT8 value equal to 0x00 (E\_OK). |
| Reference | DS\_SP\_0088 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_21] Client send Getter of a Field with MsgType Request / empty payload

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expects that the DUT send a Getter of a Field with MsgType Request / empty payload. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallFieldUINT8Getter** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds). 7. DUT: Sends SOME/IP Request message for getter of field **TestFieldUINT8**. 8. TESTER: Verify received SOME/IP Request message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0026 (getter of **TestFieldUINT8**).    3. Length is equal to 0x00000008 (8 bytes).    4. Payload is empty and does not provide any data. |
| Reference | DS\_SP\_0098 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_22] Client send Setter of a Field with MsgType Request / payload

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expects that the DUT sends a Setter of a Field with MsgType Request / payload. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallFieldUINT8Setter** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for setter of field **TestFieldUINT8**. 8. TESTER: Verify received SOME/IP Request message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0027 (setter of **TestFieldUINT8**).    3. Length is equal to 0x00000009 (9 bytes).    4. Payload provides an UINT8 value equal to **UINT8\_VALUE**. |
| Reference | DS\_SP\_0099 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_23] Forward Negative Return Code of Response to application

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expects that the DUT forwards a Negative Return Code of Response to the application. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test.  **RETURN\_CODE**: (Uint8) Return Code different than 0x00 (E\_OK). |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8**. 8. TESTER: Verify received SOME/IP Request message. 9. TESTER: Sends corresponding SOME/IP Response message with modified Return Code set to **RETURN\_CODE**. 10. TESTER: Waits for **TimeoutResponse**. 11. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError** of ETS instance **ETS\_InstanceID\_0**. 12. DUT: Sends corresponding SOME/IP Response message. 13. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**). 3. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides UINT8 value equal to **RETURN\_CODE**. |
| Reference | DS\_SP\_0033 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_24] Client forwards received error from Error msg to application

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expects that the DUT forwards a received error from Error msg to the application. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test.  **UTF8\_STRING**: UTF8 encoded string used as arbitrary payload for the test.  **RETURN\_CODE**: (Uint8) Return Code different than 0x00 (E\_OK). |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8**. 8. TESTER: Verify received SOME/IP Request message. 9. TESTER: Sends SOME/IP Error message in order to answer the preceding request with the following parameters:    * Return Code is set to **RETURN\_CODE**.    * Payload is set to **UTF8\_STRING**. 10. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError** of ETS instance **ETS\_InstanceID\_0**. 11. DUT: Sends corresponding SOME/IP Response message. 12. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**). 3. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides UINT8 value equal to **RETURN\_CODE**. |
| Reference | DS\_SP\_0110 |
| Notes |  |

#### [SOMEIPCLT\_RPC\_25] Receive Error msg with specific payload layout

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expects that the DUT receive an Error msg with a specific payload layout. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **RETURN\_CODE**: (Uint8) Return Code different than 0x00 (E\_OK).  **UINT8\_VALUE\_1**: Unsigned integer value used as arbitrary payload for the test.  **UINT8\_VALUE\_2**: Unsigned integer value used as arbitrary payload for the test. (Must be different value from **UINT8\_VALUE\_1**). |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE\_1**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8**. 8. TESTER: Verify received SOME/IP Request message. 9. TESTER: Sends SOME/IP Error message in order to answer the preceding request with the following parameters:    * Return Code is set to **RETURN\_CODE**.    * Payload is set to **UINT8\_VALUE\_2** 10. TESTER: Sends SOME/IP Request message for method **clientSericeGetLastError** of ETS instance **ETS\_InstanceID\_0**. 11. DUT: Sends corresponding SOME/IP Response message. 12. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**).    3. requstValue equal to **UINT8\_VALUE\_1**. 3. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload equal to **RETURN\_CODE**. |
| Reference | DS\_SP\_0111 |
| Notes |  |

### Transport Protocol

#### [SOMEIPCLT\_TP\_01] UDP Binding really use UDP

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that an UDP binding is used for the communication, that some/IP really use the UDP protocol for an SOME/IP-SD Message. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8**. 8. TESTER: Verify received SOME/IP Request message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. SOME/IP Request message is received via UDP within **TimeoutRequest.** |
| Reference | DS\_SP\_0115, DS\_SP\_0122 |
| Notes |  |

#### [SOMEIPCLT\_TP\_03] Receive Multiple SOME/IP messages in one UDP package

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The tester Sending two SOMEIP messages in one UDP package. DUT hast to reply on both SOMEIP messages and send the correct response. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **TIME\_SUBSCRIPTION\_1**: (Uint8) Duration of Eventgroup subscription in seconds.  **UINT8\_VALUE\_1**: Unsigned integer value used as arbitrary payload for the test.  **UINT8\_VALUE\_2**: Unsigned integer value used as arbitrary payload for the test. Shall be different than **UINT8\_VALUE\_1.** |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay set to 0x00 (0 seconds).** 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00.    * **duration** is set to **TIME\_SUBSCRIPTION\_1**.    * **eventgroupID** is set to 0x0005. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 8. DUT: Sends unicast SOME/IP-SD Subscribe Eventgroup entry. 9. TESTER: Verify received SOME/IP-SD message. 10. TESTER: Sends unicast SOME/IP-SD message with the corresponding Subscribe Eventgroup Acknowledgement Entry. 11. TESTER: Sends UDP datagram carrying the following SOME/IP messages in the prescribed order:     * Notification for **TestEventUINT8** with value set to **UINT8\_VALUE\_1**.     * Notification for **TestEventUINT8** with value set to **UINT8\_VALUE\_2**. 12. TESTER: Sends SOME/IP Request for method **clientServiceGetLastValueOfEventUDPUnicast** of ETS instance **ETS\_InstanceID\_0.** 13. DUT: Sends corrosponding SOME/IP response message. 14. TESTER: Verify received SOME/IP response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0** 2. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Eventgroup ID is equal to 0x0005.    3. Entry: TTL is not Equal 0x0000.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 3. SOME/IP response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Entry: Instance ID is equal to **ETS\_InstanceID\_0**.    3. Method ID is equal to 0x3C (**clientServiceGetLastValueOfEventUDPUnicast**).    4. Return Code is equal to 0x00 (E\_OK).    5. Payload provides an UINT8 value equal to **UINT8\_VALUE\_2**. |
| Reference | DS\_SP\_0118, DS\_SP\_0124, DS\_SP\_0125 |
| Notes |  |

#### [SOMEIPCLT\_TP\_04] TCP Binding really use TCP

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that when an TCP binding is chosen for the communication, that some/IP really use the TCP protocol |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8Reliable** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8RELIABLE** of ETS instance **ETS\_InstanceID\_0**. 8. TESTER: Verify received SOME/IP Request message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. SOME/IP Request message is received via TCP within **TimeoutRequest.** |
| Reference | DS\_SP\_0115 |
| Notes |  |

#### [SOMEIPCLT\_TP\_05] TCP Binding transports message with a length larger than 1400

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Sending a SOMEIP messages over TCP Binding transports message with a length larger than 1400 |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING**: UTF8 string value used as arbitrary payload for the test.  **UTF8\_STRING\_LENGTH**: UTF8 string size must be greater than 1400. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds).   1. DUT: Activates client service instance and enters Service Discovery start-up sequence. 2. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 3. TESTER: Verify received SOME/IP-SD message. 4. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 5. TESTER: Sends SOME/IP Request message for method **clientServiceCallFieldUTF8DynamicReliableSetter** with the following parameters:    * **delay** is set to 0x00.    * **dataLength** is set to **UTF8\_STRING\_LENGTH**.    * **data** is set to **UTF8\_STRING.** 6. DUT: Sends SOME/IP Request message for method setter of **TestFieldUTF8DynamicReliable** of ETS instance **ETS\_InstanceID\_0**. 7. TESTER: Verify received SOME/IP Request message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0561 (Setter of **TestFieldUTF8DynamicReliable**).    3. Payload provides UINT8 array equal to **UTF8\_STRING.** |
| Reference | DS\_SP\_0035, DS\_SP\_0128 |
| Notes |  |

#### [SOMEIPCLT\_TP\_06] Client opens TCP connection before Request

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Client opens TCP connection before Request. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT have no existing TCP connection at the beginning |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 6. DUT: Try to establish TCP connection by sending SYN. 7. TESTER: Accepts TCP connection by sending SYN + ACK. 8. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8Reliable** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 9. DUT: Sends SOME/IP Request message for method **echoUINT8RELIABLE** of ETS instance **ETS\_InstanceID\_0**. 10. TESTER: Verify received SOME/IP Request message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. TCP SYN is received within **TimeoutCloseTcpConnection**. 3. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0000A (**echoUINT8RELIABLE**). |
| Reference | DS\_SP\_0133, DS\_SD\_0127 |
| Notes |  |

#### [SOMEIPCLT\_TP\_07] Client opens TCP connection before Event

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Client opens TCP connection before Event. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT have no existing TCP connection at the beginning |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **TIME\_SUBSCRIPTION\_1**: (Uint8) Duration of Eventgroup subscription in seconds. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 6. DUT: Try to establish TCP connection by sending SYN. 7. TESTER: Accepts TCP connection by sending SYN + ACK. 8. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00.    * **duration** is set to **TIME\_SUBSCRIPTION\_1**.    * **eventgroupID** is set to 0x0002. 9. DUT: Sends unicast SOME/IP-SD Subscribe Eventgroup entry. 10. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. TCP SYN is received within **TimeoutCloseTcpConnection**. 3. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Eventgroup ID is equal to 0x0002.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**.    4. Entry: TTL is not Equal 0x0000. |
| Reference | DS\_SP\_0133, DS\_SD\_0127 |
| Notes |  |

#### [SOMEIPCLT\_TP\_08] Request use already opened TCP connection from Event

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Request use already opened TCP connection from Event. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **TIME\_SUBSCRIPTION\_1**: (Uint8) Duration of Eventgroup subscription in seconds. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00.    * **duration** is set to **TIME\_SUBSCRIPTION\_1**.    * **eventgroupID** is set to 0x0002. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 8. DUT: Sends unicast SOME/IP-SD Subscribe Eventgroup entry. 9. TESTER: Verify received SOME/IP-SD message. 10. TESTER: Sends unicast SOME/IP-SD message with the corresponding Subscribe Eventgroup Acknowledgement Entry. 11. TESTER: Verify existing TCP connections. 12. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 13. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8Reliable** with the following parameters:     * **delay** is set to 0x00.     * **value** is set to **UINT8\_VALUE**. 14. DUT: Sends SOME/IP Request message for method **echoUINT8RELIABLE** of ETS instance **ETS\_InstanceID\_0.** 15. TESTER: Verify received SOME/IP Request message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Eventgroup ID is equal to 0x0002.    3. Entry: TTL is not Equal 0x0000.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0**.    5. TCP connection to the DUT is established. 3. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x000A (**echoUINT8RELIABLE**).    3. Received request message’s source IP and Port is equal to **ETS\_IPv4Address** and **ETS\_PortTCP\_0**. |
| Reference | DS\_SP\_0130, DS\_SP\_0132 |
| Notes |  |

#### [SOMEIPCLT\_TP\_09] Request opens new TCP connection to 2nd Service instance

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Subscriber: Uses two simulated server instance (TESTER) at the same IP Address (same ECU) and two client instance (DUT)  The client DUT opens one TCP connection to the first simulated Server instance and after this it opens a second connection to the second Server instance insteadt of continue uses the first TCP connection.  In this case because of the definition of the ETS specification we uses 2 Client instances, too, because there is defined that one Server instance is connected only to one Client instance. |
| Preconditions | DUT is running and offering the Enhanced Testability Service in two instances which only differs by its instance key. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test  **PORT\_NUMBER**: (Uint16) Number of the TCP port used for the first service instance of ETS\_InstanceID\_0.  **PORT\_NUMBER\_1**: (Uint16) Number of the TCP port used for the second service instance of **ETS\_InstanceID\_1**. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends SOME/IP-SD unicast message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance of Instance ID **ETS\_InstanceID\_0** which following IPv4 endpoint options:    * IPv4 Address is set to **IP\_ADDRESS**.    * L4-Proto is set to 0x06 (TCP).    * Port Number is set to **PORT\_NUMBER**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8Reliable** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8RELIABLE** of ETS instance **ETS\_InstanceID\_0**. 8. TESTER: Verify received SOME/IP Request message. 9. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSelectInstance** with parameter **instanceKey** set to 0x02 (Specific service version instance). 10. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 11. DUT: Activates 2nd client service instance and enters Service Discovery start-up sequence. 12. DUT: Sends unicast SOME/IP-SD message carrying a Find Service Entry. 13. TESTER: Verify received SOME/IP-SD message. 14. TESTER: Sends unicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0** which following IPv4 endpoint options:     * IPv4 Address is set to **IP\_ADDRESS**.     * L4-Proto is set to 0x06 (TCP).     * Port Number is set to **PORT\_NUMBER\_1**. 15. DUT: Try to establish TCP connection by sending SYN to **PORT\_NUMBER\_1**. 16. TESTER: Accepts TCP connection by sending SYN + ACK. 17. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8Reliable** with the following parameters:     * **delay** is set to 0x00.     * **value** is set to **UINT8\_VALUE** 18. DUT: Sends SOME/IP Request message for method **echoUINT8RELIABLE** of ETS instance **ETS\_InstanceID\_0**. 19. TESTER: Verify received SOME/IP Request message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x000A (**echoUINT8RELIABLE**). 3. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 4. TCP SYN is received within **TimeoutCloseTcpConnecton.** 5. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x000A (**echoUINT8RELIABLE**). |
| Reference | DS\_SP\_0130 |
| Notes |  |

#### [SOMEIPCLT\_TP\_10] Receive Unaligned SOME/IP message over TCP

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The tester is sending two SOMEIP messages in one TCP package and one SOMEIP message is unaligned. DUT has to reply to all two SOMEIP messages and send the correct response. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **TIME\_SUBSCRIPTION\_1**: (Uint8) Duration of Eventgroup subscription in seconds.  **UINT8\_VALUE\_1**: Unsigned integer value used as arbitrary payload for the test.  **UINT8\_VALUE\_2**: Unsigned integer value used as arbitrary payload for the test. Shall be different than **UINT8\_VALUE\_1.** |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00.    * **duration** is set to **TIME\_SUBSCRIPTION\_1**.    * **eventgroupID** is set to 0x0002. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 8. DUT: Sends unicast SOME/IP-SD Subscribe Eventgroup entry. 9. TESTER: Verify received SOME/IP-SD message. 10. TESTER: Sends unicast SOME/IP-SD message with the corresponding Subscribe Eventgroup Acknowledgement Entry. 11. TESTER: Sends TCP segment carrying the following SOME/IP messages in the prescribed order:     * Unaligned notification for **TestEventUINT8Reliable** with value set to **UINT8\_VALUE\_1**.     * Unaligned Notification for **TestEventUINT8Reliable** with value set to **UINT8\_VALUE\_2**. 12. TESTER: Sends SOME/IP Request for method **clientServiceGetLastValueOfEventTCP** of ETS instance **ETS\_InstanceID\_0.** 13. DUT: Sends corrosponding SOME/IP response message. 14. TESTER: Verify received SOME/IP response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Eventgroup ID is equal to 0x0002.    3. Entry: TTL is not Equal 0x0000.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 3. SOME/IP response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x3B (**clientServiceGetLastValueOfEventTCP**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **UINT8\_VALUE\_2**. |
| Reference | DS\_SP\_0119 |
| Notes |  |

#### [SOMEIPCLT\_TP\_11] Receive multiple SOME/IP messages in one TCP segment

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The tester sending two SOMEIP messages in one TCP segment. DUT has to reply on all two SOMEIP messages and send the correct response.  Check the handling when more SOMEIP messages are in one transport protocol frame. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **TIME\_SUBSCRIPTION\_1**: (Uint8) Duration of Eventgroup subscription in seconds.  **UINT8\_VALUE\_1**: Unsigned integer value used as arbitrary payload for the test.  **UINT8\_VALUE\_2**: Unsigned integer value used as arbitrary payload for the test. Shall be different than **UINT8\_VALUE\_1.** |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00.    * **duration** is set to **TIME\_SUBSCRIPTION\_1**.    * **eventgroupID** is set to 0x0002. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 8. DUT: Sends unicast SOME/IP-SD Subscribe Eventgroup entry. 9. TESTER: Verify received SOME/IP-SD message. 10. TESTER: Sends unicast SOME/IP-SD message with the corresponding Subscribe Eventgroup Acknowledgement Entry. 11. TESTER: Sends TCP segment carrying the following SOME/IP messages in the prescribed order:     * Notification for **TestEventUINT8Reliable** with value set to **UINT8\_VALUE\_1**.     * Notification for **TestEventUINT8Reliable** with value set to **UINT8\_VALUE\_2**. 12. TESTER: Sends SOME/IP Request for method **clientServiceGetLastValueOfEventTCP** of ETS instance **ETS\_InstanceID\_0.** 13. DUT: Sends corresponding SOME/IP response message. 14. TESTER: Verify received SOME/IP response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Eventgroup ID is equal to 0x0002.    3. Entry: TTL is not Equal 0x0000.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 3. SOME/IP response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x3B (**clientServiceGetLastValueOfEventTCP**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **UINT8\_VALUE\_2**. |
| Reference | DS\_SP\_0118 |
| Notes |  |

#### [SOMEIPCLT\_TP\_12] Close not needed TCP connection

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to close a TCP connection when this connection is not needed anymore. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8Reliable** with the following parameters:    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. DUT: Sends SOME/IP Request message for method **echoUINT8RELIABLE**. 8. TESTER: Verify received SOME/IP Request message. 9. TESTER: Sends corresponding SOME/IP response message. 10. TESTER: Sends SOME/IP Request message for method **clientServiceDeactivate** with parameter delay set to 0x00 (0 seconds). 11. DUT: Close TCP connection by sending FIN. 12. TESTER: confirms TCP connection close by sending ACK. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x000A(**echoUINT8RELIABLE**) 3. TCP FIN is received within **TimeoutCloseTcpConnection**. |
| Reference | DS\_SP\_0134 |
| Notes |  |

#### [SOMEIPCLT\_TP\_13] Client close TCP connection if Service no longer available

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT close the TCP connection if a service of the server isn’t available anymore. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **TIME\_SUBSCRIPTION\_1**: (Uint8) Duration of Eventgroup subscription in seconds. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay set to 0x00 (0 seconds).** 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00.    * **duration** is set to **TIME\_SUBSCRIPTION\_1**.    * **eventgroupID** is set to 0x0002. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 8. DUT: Sends unicast SOME/IP-SD Subscribe Eventgroup entry. 9. TESTER: Verify received SOME/IP-SD message. 10. TESTER: Sends unicast SOME/IP-SD message with the corresponding Subscribe Eventgroup Acknowledgement Entry. 11. TESTER: Sends unicast SOME/IP-SD message with a Stop Offer Service entry. 12. DUT: Close TCP connection by sending FIN. 13. TESTER: confirms TCP connection close by sending ACK. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Eventgroup ID is equal to 0x0002.    3. Entry: TTL is not Equal 0x0000.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 3. TCP FIN is received within **TimeoutCloseTcpConnection**. |
| Reference | DS\_SP\_0135 |
| Notes |  |

#### [SOMEIPCLT\_TP\_14] Reset TCP connection when reboot is detected

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to reestablish an existing TCP connection when detecting a reboot of the peer. The reboot is simulated by sending a SOME/IP-SD message with modified values for Reboot Flag and Session ID to the DUT. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **TIME\_SUBSCRIPTION\_1**: (Uint8) Duration of Eventgroup subscription in seconds. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00.    * **duration** is set to **TIME\_SUBSCRIPTION\_1**.    * **eventgroupID** is set to 0x0006. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 8. DUT: Sends unicast SOME/IP-SD Subscribe Eventgroup entry. 9. TESTER: Verify received SOME/IP-SD message. 10. TESTER: Sends unicast SOME/IP-SD message with Reboot Flag set to 1 and Session ID set to 0x0001, carrying an Offer Service Entry for the ETS. 11. DUT: Resets TCP Connection by sending SYN. 12. TESTER: Accepts TCP connection by sending SYN + ACK. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0** 2. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides one entry.    2. Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry).    3. Entry:Instance ID is equal to **ETS\_InstanceID\_0.** 3. TCP SYN is received within **TimeoutCloseTcpConnection**. |
| Reference | DS\_SD\_0027 |
| Notes |  |

#### [SOMEIPCLT\_TP\_15] Reset TCP connection on error

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Reset TCP connection on error |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **TIME\_SUBSCRIPTION\_1**: (Uint8) Duration of Eventgroup subscription in seconds. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00.    * **duration** is set to **TIME\_SUBSCRIPTION\_1**.    * **eventgroupID** is set to 0x0006. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 8. DUT: Try to establish TCP connection by sending SYN. 9. TESTER: Denies TCP Connection by RST. 10. DUT: Resets TCP Connection by sending SYN. 11. TESTER: Accepts TCP connection by sending SYN + ACK. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. TCP SYN is received within **TimeoutSubscribe**. |
| Reference | DS\_SP\_0137 |
| Notes |  |

#### [SOMEIPCLT\_TP\_17] Request shall timed out when TCP connection is lost

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT Request shall timed out, when a TCP connection is lost. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Unsigned integer value used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8Reliable** with following parameters**:**    * **delay** is set to 0x00.    * **value** is set to **UINT8\_VALUE**. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 8. DUT: Try to establish TCP connection by sending SYN. 9. TESTER: Accepts TCP connection by sending SYN + ACK. 10. TESTER: Simulates lost TCP connection by ignoring any incoming TCP segments for a period of **TimeoutRequest**. 11. DUT: Request for method **echoUINT8RELIABLE** timed out**.** 12. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError** of ETS instance **ETS\_InstanceID\_0.** 13. DUT: Sends corresponding SOME/IP Response message. 14. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. TCP SYN is received within **TimeoutOpenTcpConnection**. 3. SOME/IP response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x521 (**clientServiceGetLastError**).    3. Payload provides an UINT8 value equal to 0x06 (E\_TIMEOUT). |
| Reference | DS\_SP\_0136 |
| Notes |  |

#### [SOMEIPCLT\_TP\_18] Magic Cookie Message Format Client

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks the Format of the SOME/IP Magic Cookie Message |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT supports sending of Magic Cookies messages.  DUT related settings for period threshold of sending Magic Cookies must be 100 milliseconds [msec]. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING:** Null terminated character string used as arbitrary payload.  **STRING\_LENGTH:** (Uint32) Length of the character string. Must be at least 30000. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request No-Return message for method **clientServiceCallFieldUTF8DynamicReliableSetter** with the following parameters:    * **delay** is set to 0x00.    * UINT32 is set to **STRING\_LENGTH.**    * UTF8 string is set to **UTF8\_STRING**. 7. DUT: Starts TCP stream for transmitting of SOME/IP Request message for setter of ETS field **TestFieldUTF8DynamicReliable.** 8. DUT: Sends SOME/IP Magic Cookie while TCP transmission of step 7 is still running. 9. TESTER: Verify received Magic Cookie message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. SOME/IP message is received within **TimeoutResponse** and contains:    1. Message ID is equal to 0xFFFF 0000.    2. Length is equal to 0x0000 0008    3. Request ID is equal to =0xDEAD BEEF    4. Protocol Version is equal to 0x01.    5. Interface version is equal to 0x01.    6. Message Type is equal to 0x01 (REQUEST\_NO\_RETURN).    7. Return Code is equal to 0x00 (E\_OK). |
| Reference | DS\_SP\_0144, DS\_SP\_0145 |
| Notes | TCP Stream triggered by the Test procedure finds no mention because test focus is on Magic cookie messages only. |

#### [SOMEIPCLT\_TP\_19] Magic Cookie Cyclic sending Client

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send Magic Cookie messages in a cyclic manner during a TCP transmission. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT supports sending of Magic Cookies messages.  DUT related settings for period threshold of sending Magic Cookies must be 100 milliseconds [msec]. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING:** Null terminated character string used as arbitrary payload.  **STRING\_LENGTH:** (Uint32) Length of the character string. Must be at least 30000.  **N\_REPETITIONS**: (Uint8) Number of interactions the cyclic delay of Magic Cookie shall be measured. Must be at least 3.  **MAGIC\_COOKIE\_CYCLIC\_DELAY**: (Uint8) Interval of sending Magic Cookie messages used by the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request No-Return message for method **clientServiceCallFieldUTF8DynamicReliableSetter** with the following parameters:    * **delay** is set to 0x00.    * UINT32 is set to **STRING\_LENGTH.**    * UTF8 string is set to **UTF8\_STRING**. 7. DUT: Starts TCP stream for transmitting of SOME/IP Request message for setter of ETS field **TestFieldUTF8DynamicReliable.** 8. DUT: Sends SOME/IP Magic Cookie while TCP transmission of step 7 is still running. 9. TESTER: Verify received Magic Cookie message. 10. Repeat preceding steps 6 and 8 until a total number of **N\_REPETITIONS** are processed. 11. TESTER: Verify the measured cyclic delay of the SOME/IP Magic Cookie messages received in step 9. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0** 2. SOME/IP message is received within **TimeoutResponse** and contains:    1. Message ID is equal to 0xFFFF 0000.    2. Request ID is equal to 0xDEAD BEEF 3. Measured delay value between two consecutive SOME/IP Magic Cookie messages is equal to **MAGIC\_COOKIE\_CYCLIC\_DELAY** +/-**TimingTolerance.** |
| Reference | DS\_SP\_0148 |
| Notes | TCP Stream triggered by the Test procedure finds no mention because test focus is on Magic cookie messages only. |

#### [SOMEIPCLT\_TP\_20] SOME/IP-TP message type set to TP\_Request

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT sends a SOME/IP message larger as 1400 bytes and the Tester checks that the received message is a SOME/IP-TP message where the message type is set to TP\_Request. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT shall support message segmentation via SOME/IP-TP.  DUT related settings for period threshold of sending Magic Cookies must be 100 milliseconds [msec]. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **ARRAY\_LENGTH**: Size of message payload in bytes. Must be greater than 1400  **UINT8\_ARRAY**: Array of UINT8 values used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8Array** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **dataLength** is set to **ARRAY\_LENGTH**.    * **data** is set to **UINT8\_ARRAY**. 7. DUT: Starts TCP stream for transmitting of SOME/IP Request message for setter of ETS field **echoUINT8Array.** 8. DUT: Sends SOME/IP Magic Cookie while TCP transmission of step 7 is still running. 9. TESTER: Verify received SOME/IP-TP segment. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0** 2. SOME/IP-TP segment is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0009 (**echoUINT8Array**).    3. Message Type is equal to 0x20 (TP\_REQUEST). |
| Reference | DS\_SP\_0028, DS\_SP\_0157 |
| Notes |  |

#### [SOMEIPCLT\_TP\_21] SOME/IP-TP message type set to TP\_RequestNoReturn

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT sends a SOME/IP message larger as 1400 bytes and the Tester checks that the received message is a SOME/IP-TP message where the message type is set to TP\_RequestNoReturn. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT shall support message segmentation via SOME/IP-TP. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **ARRAY\_LENGTH**: Size of message payload in bytes. Must be greater than 1400  **UINT8\_ARRAY**: Array of UINT8 values used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance ID **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallSetUINT8Array** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **dataLength** is set to **ARRAY\_LENGTH**.    * **data** is set to **UINT8\_ARRAY**. 7. DUT: Sends 1st segment of SOME/IP Request-No-Return message for method **setUINT8Array**. 8. TESTER: Verify received SOME/IP-TP segment. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0** 2. SOME/IP-TP segment is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0509 (**echoUINT8Array**).    3. Message Type is equal to 0x21 (TP\_REQUEST\_NO\_RETURN). |
| Reference | DS\_SP\_0028, DS\_SP\_0157 |
| Notes |  |

### Service Discovery Message Format

#### [SOMEIPCLT\_SD\_FORMAT\_01] Find Service Entry

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send a well-formed Find Service Entry which conforms to the on-wire format specified for SOME/IP-SD related Service Entries. Value of each entry field will be checked.  Referenced options are not expected for this test. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | none |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides one entry.    2. Length of Entries Array is equal to 0x00000010 (16 bytes).    3. Entry: Type is equal to 0x00 (Find Service Entry).    4. Entry: Index of 1st Option Run is equal to 0x00 (0).    5. Entry: Index of 2nd Option Run is equal to 0x00 (0).    6. Entry: Number of Option 1 is equal to 0x0 (0).    7. Entry: Number of Option 2 is equal to 0x0 (0).    8. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    9. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    10. Entry: Major Version is equal to **ETS\_MajorVersion\_0**.    11. Entry: TTL is equal to **ETS\_Lifetime**.    12. Entry: Minor Version is equal to **ETS\_MinorVersion\_0**. |
| Reference | DS\_ID\_0001, DS\_ID\_0002, DS\_ID\_0006, DS\_ID\_0007, DS\_SD\_0040, DS\_SD\_0041DS\_SD\_0042, DS\_SD\_0043, DS\_SD\_0044, DS\_SD\_0045, DS\_SD\_0046, DS\_SD\_0047, DS\_SD\_0048, DS\_SD\_0049, DS\_SD\_0050, DS\_SD\_0051 DS\_SD\_0052, DS\_SD\_104, DS\_SD\_0105, DS\_SD\_0106, DS\_SD\_0107, DS\_SD\_0108, DS\_SD\_0120 |
| Notes |  |

#### [SOMEIPCLT\_SD\_FORMAT\_02] Subscribe Eventgroup Entry

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send a well-formed Subscribe Eventgroup Entry which conforms to the on-wire format specified for SOME/IP-SD related Eventgroup Entries. Value of each entry field will be checked.  A respective option for the ETS specific UDP binding is expected to be referenced in the entry. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: Duration of Eventgroup subscription in seconds. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0005. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 8. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 9. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides one entry.    2. Length of Entries Array is equal to 0x00000010 (16 bytes).    3. Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry).    4. Entry: Index of 1st Option Run is equal to 0x0.    5. Entry: Index of 2nd Option Run is equal to 0x0.    6. Entry: Number of Option 1 is equal to 0x1.    7. Entry: Number of Option 2 is equal to 0x0.    8. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    9. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    10. Entry: Major Version is equal to **ETS\_MajorVersion\_0**.    11. Entry: TTL is equal to **ETS\_SubscriptionLifetime**.    12. Entry: All reserved and undefined bits are set to 0.    13. Entry: Initial Data Requested Flag is set to 1.    14. Entry: Counter is equal to 0x0.    15. Entry: Eventgroup ID is set to 0x0005. |
| Reference | DS\_ID\_0001, DS\_ID\_0002, DS\_ID\_0006, DS\_ID\_0007, DS\_ID\_0013, DS\_ID\_0014,DS\_SD\_0040, DS\_SD\_0053, DS\_SD\_0054, DS\_SD\_0055, DS\_SD\_0056, DS\_SD\_0057, DS\_SD\_0058, DS\_SD\_0059, DS\_SD\_0060, DS\_SD\_0061, DS\_SD\_0062, DS\_SD\_0063, DS\_SD\_0064, DS\_SD\_0065, DS\_SD\_0066, DS\_SD\_0067,DS\_SD\_0068, DS\_SD\_0104, DS\_SD\_0105, DS\_SD\_0106, DS\_SD\_0107, DS\_SD\_0108, DS\_SD\_0131 |
| Notes |  |

#### [SOMEIPCLT\_SD\_FORMAT\_03] Stop Subscribe Eventgroup Entry

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send a well-formed Stop Subscribe Eventgroup Entry which conforms to the on-wire format specified for SOME/IP-SD related Eventgroup Entries. Value of each entry field will be checked.  For Stop Subscribe Eventgroup Entries is expected that all fields, except the both for entry type and TTL, shall provide the same values as in the Subscribe Eventgroup Entry that is being stopped.  A respective option for the ETS specific UDP binding is expected to be referenced in the entry. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0005. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 8. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 9. TESTER: Verify received SOME/IP-SD message. 10. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 11. TESTER: Waits for **T\_SUBSCRIPTION**. 12. DUT: Sends unicast SOME/IP-SD message carrying an Eventgroup Entry of type 0x06. 13. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides one entry.    2. Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry).    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 3. SOME/IP-SD message is received before **TimeoutStopSubscribe** and contains:    1. Entries Array provides one entry.    2. Length of Entries Array is equal to 0x00000010 (16 bytes).    3. Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry).    4. Entry: Index of 1st Option Run is equal to 0x0.    5. Entry: Index of 2nd Option Run is equal to 0x0.    6. Entry: Number of Option 1 is equal to 0x1.    7. Entry: Number of Option 2 is equal to 0x0.    8. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    9. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    10. Entry: Major Version is equal to **ETS\_MajorVersion\_0**.    11. Entry: TTL is equal to 0x000000.    12. Entry: All reserved and undefined bits are set to 0.    13. Entry: Initial Data Requested Flag is set to 1.    14. Entry: Counter is equal to 0x0.    15. Entry: Eventgroup ID is set to 0x0005. |
| Reference | DS\_SD\_0134 |
| Notes |  |

#### [SOMEIPCLT\_SD\_FORMAT\_04] IPv4 Endpoint Option for TCP

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks format an IPv4 Endpoint Option. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0002. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0** with following IPv4 endpoint options.    * IPv4 Address is set to **TESTER\_IPv4Address**.    * L4-Proto is set to 0x06 (TCP).    * Port Number is set to **TESTER\_PortTCP**. 8. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 9. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Options Array provides an option of type 0x04 (IPv4 Endpoint Option).    2. Option: Length is equal to 0x0009.    3. Option: Discardable Flag equal to 0.    4. Option: All reserved bits are set to 0.    5. Option: IPv4-Address is equal to **TESTER\_IPv4Address**.    6. Option: L4-Proto is equal to 0x06 (TCP).    7. Option: Port Number is equal to **TESTER\_PortTCP**. |
| Reference | DS\_SD\_0070, DS\_SD\_0075 |
| Notes |  |

#### [SOMEIPCLT\_SD\_FORMAT\_05] IPv4 Endpoint Option for UDP

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks format an IPv4 Endpoint Option. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0** with following IPv4 endpoint options.    * IPv4 Address is set to **TESTER\_IPv4ADDRESS**.    * L4-Proto is set to 0x11 (UDP).    * Port Number is set to **TESTER\_PortUDP**. 8. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 9. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Options Array provides an option of type 0x04 (IPv4 Endpoint Option).    2. Option: Length is equal to 0x0009.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Discardable Flag equal to 0.    5. Option: All reserved bits are set to 0.    6. Option: IPv4-Address is equal to **TESTER\_IPv4ADDRESS**.    7. Option: L4-Proto is equal to 0x11 (UDP).    8. Option: Port Number is equal to **TESTER\_PortUDP**. |
| Reference | DS\_SD\_0070, DS\_SD\_0075 |
| Notes |  |

#### [SOMEIPCLT\_SD\_FORMAT\_06] IPv4 SD Endpoint Option

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks format an IPv4 SD Endpoint Option. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 address used in the SD Endpoint Option for the test.  **PORT\_NUMBER**: (Uint16) Number of the UDP port used in the SD Endpoint Option for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. TESTER: Sends SOME/IP Request-No-Return message for ETS method **provideSDEndpointOption** with the following parameters:    * **duration** is set to **TimeoutOfferService**.    * **ip\_address** is set to **IP\_ADDRESS**.    * **portNumber** is set to **PORT\_NUMBER**. 4. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provides an option of type 0x24 (IPv4 SD Endpoint Option).    2. Option: Length is equal to 0x0009.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Discardable Flag equal to 0.    5. Option: All reserved bits are set to 0.    6. Option: IPv4-Address is equal to **IP\_ADDRESS**.    7. Option: L4-Proto is equal to 0x11 (UDP).    8. Option: Port Number is equal to **PORT\_NUMBER**. |
| Reference | DS\_SD\_0089 |
| Notes |  |

#### [SOMEIPCLT\_SD\_FORMAT\_07] Configuration Option: Length and Type

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the received Configuration Option have the proper of the Length and Type |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING\_1**: Arbitrary name part of a configuration string related item.  **UTF8\_STRING\_2**: Arbitrary value part of a configuration string related item.  **STRING\_LENGTH\_1**: (Uint32) Length of name part in number of characters.  **STRING\_LENGTH\_2**: (Uint32) Length of value part in number of characters. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. TESTER: Sends SOME/IP Request message for ETS method **provideConfigurationOption** with the following parameters:    * **duration** is set to **TimeoutOfferService**.    * **keyLength** is set to **STRING\_LENGTH\_1**.    * **key** is set to **UTF8\_STRING\_1**.    * **valueLength** is set to **STRING\_LENGTH\_2**.    * **value** is set to **UTF8\_STRING\_2**.    * **key2Length** is set to 0x00.    * **key2** is set empty string. 4. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provides an option of type 0x01 (Configuration Option).    2. Option: Length is equal to sum of **STRING\_LENGTH\_1, STRING\_LENGTH\_2** and 4.    3. Option: All reserved bits are set to 0.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** |
| Reference | DS\_SD\_0090 |
| Notes |  |

#### [SOMEIPCLT\_SD\_FORMAT\_08] Configuration Option: String contains character sequence separated with length field

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the received Configuration Option String contains a character sequence separated with length field. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING\_1**: Arbitrary name part of a configuration string related item.  **UTF8\_STRING\_2**: Arbitrary value part of a configuration string related item.  **STRING\_LENGTH\_1**: (Uint32) Length of name part in number of characters.  **STRING\_LENGTH\_2**: (Uint32) Length of value part in number of characters. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. TESTER: Sends SOME/IP Request message for ETS method **provideConfigurationOption** with the following parameters:    * **duration** is set to **TimeoutOfferService**.    * **keyLength** is set to **STRING\_LENGTH\_1**.    * **key** is set to **UTF8\_STRING\_1**.    * **valueLength** is set to **STRING\_LENGTH\_2**.    * **value** is set to **UTF8\_STRING\_2**.    * **key2Length** is set to 0x00.    * **key2** is set empty string. 4. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provides an option of type 0x01 (Configuration Option).    2. Option: Value of 1st byte in Configuration String is equal to the sum of **STRING\_LENGTH\_1, STRING\_LENGTH\_2** and 1.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** |
| Reference | DS\_SD\_0092, DS\_SD\_0093 |
| Notes |  |

#### [SOMEIPCLT\_SD\_FORMAT\_09] Configuration Option: String contains key with value

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the received Configuration Option String contains a key with value. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING\_1**: Arbitrary name part of a configuration string related item.  **UTF8\_STRING\_2**: Arbitrary value part of a configuration string related item.  **STRING\_LENGTH\_1**: (Uint32) Length of name part in number of characters.  **STRING\_LENGTH\_2**: (Uint32) Length of value part in number of characters. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. TESTER: Sends SOME/IP Request message for ETS method **provideConfigurationOption** with the following parameters:    * **duration** is set to **TimeoutOfferService**.    * **keyLength** is set to **STRING\_LENGTH\_1**.    * **key** is set to **UTF8\_STRING\_1**.    * **valueLength** is set to **STRING\_LENGTH\_2**.    * **value** is set to **UTF8\_STRING\_2**.    * **key2Length** is set to 0x00.    * **key2** is set to empty string. 4. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provides an option of type 0x01 (Configuration Option).    2. Option: Configuration String consists of the following items in the preceding order:       * Single byte with a value equal to the sum of **STRING\_LENGTH\_1**, **STRING\_LENGTH\_2** and 1.       * Substring equal to **UTF8\_STRING\_1**.       * Single byte with a value equal to 0x3D (“=”).       * Substring equal to **UTF8\_STRING\_2**.       * Single byte with a value equal to 0x00 (“\0”).    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** |
| Reference | DS\_SD\_0091, DS\_SD\_0095, DS\_SD\_0096, DS\_SD\_0097, DS\_SD\_0098 |
| Notes |  |

#### [SOMEIPCLT\_SD\_FORMAT\_10] Configuration Option: String contains key with empty value

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the received Configuration Option String contains key with empty value. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING**: Arbitrary name part of a configuration string related item.  **STRING\_LENGTH**: (Uint32) Length of name part in number of characters. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. TESTER: Sends SOME/IP Request message for ETS method **provideConfigurationOption** with the following parameters:    * **duration** is set to **TimeoutOfferService**.    * **keyLength** is set to **STRING\_LENGTH**.    * **key** is set to **UTF8\_STRING**.    * **valueLength** is set to 0x00.    * **value** is set to 0x00 (empty string).    * **key2Length** is set to 0x00.    * **key2** is set to empty string. 4. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provides an option of type 0x01 (Configuration Option).    2. Option: Configuration String consists of the following items in the preceding order:       * Single byte with a value equal to **STRING\_LENGTH** + 1.       * Substring equal to **UTF8\_STRING**.       * Single byte with a value equal to 0x3D (“=”).       * Single byte with a value equal to 0x00 (“\0”).    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** |
| Reference | DS\_SD\_0091, DS\_SD\_0095, DS\_SD\_0096, DS\_SD\_0097, DS\_SD\_0098, DS\_SD\_0100 |
| Notes |  |

#### [SOMEIPCLT\_SD\_FORMAT\_11] Configuration Option: String contains key without value

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the received Configuration Option String contains a key without any value. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING\_1**: 1st arbitrary name part of a configuration string related item.  **UTF8\_STRING\_2**: 2nd arbitrary name part of a configuration string related item.  **STRING\_LENGTH\_1**: (Uint32) Length of 1st name part in number of characters.  **STRING\_LENGTH\_2**: (Uint32) Length of 2nd name part in number of characters. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. TESTER: Sends SOME/IP Request message for ETS method **provideConfigurationOption** with the following parameters:    * **duration** is set to **TimeoutOfferService**.    * **keyLength** is set to **STRING\_LENGTH\_1**.    * **key** is set to **UTF8\_STRING\_1**.    * **valueLength** is set to 0x00.    * **value** is set to 0x00 (empty string).    * **key2Length** is set to **STRING\_LENGTH\_2**.    * **key2** is set to **UTF8\_STRING\_2**. 4. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provides an option of type 0x01 (Configuration Option).    2. Option: Configuration String consists of the following items in the preceding order:       * Single byte with a value equal to **STRING\_LENGTH\_1** + 1.       * Substring equal to **UTF8\_STRING\_1**.       * Single byte with a value equal to 0x3D (“=”).       * Single byte with a value equal to **STRING\_LENGTH\_2** + 1.       * Substring equal to **UTF8\_STRING\_2**.       * Single byte with a value equal to 0x00 (“\0”).    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** |
| Reference | DS\_SD\_0091, DS\_SD\_0095, DS\_SD\_0097, DS\_SD\_0098, DS\_SD\_0099 |
| Notes |  |

#### [SOMEIPCLT\_SD\_FORMAT\_12] Configuration Option: String contains multiple key/value pairs of same key

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the received Configuration Option String contains multiple key/value pairs of same key. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING\_1**: Arbitrary name part of a configuration string related item.  **UTF8\_STRING\_2**: Arbitrary value part of a configuration string related item.  **STRING\_LENGTH\_1**: (Uint32) Length of name part in number of characters.  **STRING\_LENGTH\_2**: (Uint32) Length of value part in number of characters. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. TESTER: Sends SOME/IP Request message for ETS method **provideConfigurationOption** with the following parameters:    * **duration** is set to **TimeoutOfferService**.    * **keyLength** is set to **STRING\_LENGTH\_1**.    * **key** is set to **UTF8\_STRING\_1**.    * **valueLength** is set to **STRING\_LENGTH\_2**.    * **value** is set to **UTF8\_STRING\_2**.    * **key2Length** is set to **STRING\_LENGTH\_1**.    * **key2** is set to **UTF8\_STRING\_1**. 4. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Options Array provides an option of type 0x01 (Configuration Option).    2. Option: Configuration String consists of the following items in the preceding order:       * Single byte with a value equal to the sum of **STRING\_LENGTH\_1**, **STRING\_LENGTH\_2** and 1.       * Substring equal to **UTF8\_STRING\_1**.       * Single byte with a value equal to 0x3D (“=”).       * Substring equal to **UTF8\_STRING\_2**.       * Single byte with a value equal to **STRING\_LENGTH\_1** + 1.       * Substring equal to **UTF8\_STRING\_1**.       * Single byte with a value equal to 0x00 (“\0”).    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** |
| Reference | DS\_SD\_0091, DS\_SD\_0101 |
| Notes |  |

#### [SOMEIPCLT\_SD\_FORMAT\_13] Length covers message length from first byte after the field

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the received message Length covers message length. The first byte after the length field. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | none |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Length is equal to the sum of Length of Entries Array, Length of Options Array and 20. |
| Reference | DS\_SD\_0008 |
| Notes |  |

#### [SOMEIPCLT\_SD\_FORMAT\_14] Find Service shall not reference IPv4 Endpoint or Multicast Option

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the send Find Service shall not reference IPv4 Endpoint or Multicast Option. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | none |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Option Runs do not reference any options of type 0x04 (IPv4 Endpoint Option) and 0x14 (IPv4 Multicast Option) |
| Reference | DS\_SD\_0121 |
| Notes |  |

#### [SOMEIPCLT\_SD\_FORMAT\_15] Subscribe Eventgroup shall reference IPv4 Endpoint Option

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the send subscribe Eventgroup shall reference IPv4 Endpoint Option. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: Duration of Eventgroup subscription in seconds. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 8. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 9. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Subscribe Eventgroup Entry references an option of type 0x04 (IPv4 Endpoint Option) and Discardable Flag equal to 0    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** |
| Reference | DS\_SD\_0078, DS\_SD\_0132 |
| Notes |  |

#### [SOMEIPCLT\_SD\_FORMAT\_16] Stop Subscribe Eventgroup shall reference same Options as Subscribe

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the send Stop Subscribe Eventgroup shall reference same Options as Subscribe. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero. It is smaller than the TTL value of the Offer Service Entry. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 8. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 9. TESTER: Verify received SOME/IP-SD message. 10. TESTER: Sends SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 11. TESTER: Waits for **T\_SUBSCRIPTION**. 12. DUT: Sends unicast SOME/IP-SD message carrying a Stop Subscribe Eventgroup Entry. 13. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides one entry.    2. Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry).    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 3. SOME/IP-SD message is received before **TimeoutStopSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Stop Subscribe Eventgroup Entry).    2. Entry: TTL is set to 0x0000.    3. Entry: Option Runs reference same options as the preceding Subscribe Eventgroup Entry that shall be stopped.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** |
| Reference | DS\_SD\_0135 |
| Notes |  |

#### [SOMEIPCLT\_SD\_FORMAT\_17] Support of multiple Entries per message

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that the send SOME/IP-SD message Support of multiple Entries per message. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0005. 7. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 8. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0**. 9. DUT: Sends multicast SOME/IP-SD message carrying multiple entries. 10. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides two entries.    2. 1st Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry).    3. 2nd Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. 1nd Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry).    5. 2nd Entry: Instance ID is equal to **ETS\_InstanceID\_0.** |
| Reference | DS\_SD\_0039 |
| Notes |  |

### Service Discovery Communication Behavior

#### [SOMEIPCLT\_SD\_BEHAVIOR\_01] Initial Wait Phase: Delay sending of 1st Find Service

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Initial Wait Phase: Delay sending of 1st Find Service. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT related settings for INITIAL\_DELAY must define a minimum delay of at least 100 milliseconds [msec]. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_INITIAL\_DELAY\_MIN**: (Uint8) The minimal expected delay before sending out the first Find Service Entry for a searched SOME/IP service.  **T\_INITIAL\_DELAY\_MAX**: (Uint8) The maximal expected delay before sending out the first Find Service Entry for a searched SOME/IP service. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received after **T\_INITIAL\_DELAY\_MIN** but before **T\_INITIAL\_DELAY\_MAX** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0143, DS\_SD\_0146 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_02] Initial Wait Phase: Delay of 1st Find Service is random value

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Initial Wait Phase: Delay of 1st Find Service is random value. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT related settings for INITIAL\_DELAY must define different values for lower and upper limit. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **N\_ITERATIONS**: (Uint8) Number of iterations the initial delay shall be measured. Must not be less than a number of 3.  **T\_INITIAL\_DELAY\_MIN**: (Uint8) The minimal expected delay before sending out the first Find Service Entry for a searched SOME/IP service.  **T\_INITIAL\_DELAY\_MAX**: (Uint8) The maximal expected delay before sending out the first Find Service Entry for a searched SOME/IP service. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceDeactivate** with parameter **delay** set to 0x00 (0 seconds). 6. DUT: Deactivates client service instance. 7. Repeat preceding steps 1 to 6 until a total number of **N\_ITERATIONS** are processed. 8. TESTER: Compares the measured initial delay for SOME/IP-SD messages received in step 1 to 7. |
| Verification | 1. SOME/IP-SD message is received after **T\_INITIAL\_DELAY\_MIN** but before **T\_INITIAL\_DELAY\_MAX** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. Measured delay values differ in at least 1 millisecond [msec] +/- **TimingToleranceMillisec**. |
| Reference | DS\_SD\_0143, DS\_SD\_0147, DS\_SD\_0148 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_03] Repetition Phase: Delay sending of 2nd Find Service

|  |  |
| --- | --- |
| tem | Description |
| Purpose | Repetition Phase: Delay sending of 2nd Find Service. The Tester expect a well-defined delay after sending the first Find service of the DUT. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP parameter REPETITIONS\_MAX must not be zero.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_REPETITION\_BASE\_DELAY**: (Uint16) Base delay value in milliseconds [msec] configured for the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 6. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received immediately after **T\_REPETITION\_BASE\_DELAY** +/- **TimingToleranceMillisec** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0143, DS\_SD\_0144, DS\_SD\_0153, DS\_SD\_0154 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_04] Repetition Phase: Number of sent Find Service Entries

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks the number of Find Service entries which were sent for a specific service instance during Repetition Phase of the Service Discovery start-up sequence. The maximum number of repetitions shall correspond to a predefined value. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP Parameter REPETITIONS\_MAX must be a value greater than 1.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **N\_REPETITIONS**: (Uint8) Number of message repetitions configured for the SOME/IP-SD related Repetition Phase of the DUT.  **T\_REPETITION\_PHASE**: (Uint8) The expected overall duration in milliseconds [msec] of the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. Repeat preceding steps 3 and 4 for a total number of **N\_REPETITIONS**. 6. TESTER: Verify SOME/IP-SD messages received in step 5. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. All SOME/IP-SD messages were received before **T\_REPETITION\_PHASE** is elapsed. |
| Reference | DS\_SD\_0143, DS\_SD\_0156 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_05] Repetition Phase: Delay is doubled after each following Find Service

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks delay between Find Service entries of a specific service instance, which were sent during Repetition Phase of the Service Discovery start-up sequence. After each following entry the delay shall be doubled. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP parameter REPETITIONS\_MAX must be a value greater than 1.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **N\_REPETITIONS**: (Uint8) Number of message repetitions configured for the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. Repeat preceding steps 3 and 4 for a total number of **N\_REPETITIONS**. 6. TESTER: Compares the measured repetition delay for SOME/IP-SD messages received in step 5. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. Measured delay value between two consecutive SOME/IP-SD messages is doubled by each following message, considering a tolerance of+/- **TimingToleranceMillisec**. |
| Reference | DS\_SD\_0143, DS\_SD\_0155 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_06] Repetition Phase: Same settings for all clients within an ECU

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to use the same Repetition Phase related settings for all hosted client instances within one ECU. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP parameter REPETITIONS\_MAX must be a value greater than 1.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **N\_REPETITIONS**: (Uint8) Number of message repetitions configured for the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. Repeat preceding steps 3 and 4 for a total number of **N\_REPETITIONS**. 6. TESTER: Compares the measured repetition delay for SOME/IP-SD messages received in step 5. 7. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceDeactivate** with parameter **delay** set to 0x00 (0 seconds). 8. DUT: Deactivates client service instance. 9. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSelecInstance** with parameter **instanceKey** set to 0x01 (Duplicated instance). 10. Repeat preceding steps 1 to 6 to send a second consecutive message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. Measured delay value between two consecutive SOME/IP-SD messages is doubled by each following message, considering a tolerance of+/- **TimingToleranceMillisec**. |
| Reference | DS\_SD\_0143, DS\_SD\_0161 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_07] Repetition Phase: After Link-loss/Link-up jump to Initial Wait Phase

|  |  |
| --- | --- |
| Item | Description |
| Purpose | After Link-loss/Link-up jump to Initial Wait Phase |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP Parameter REPETITIONS\_MAX must be a value greater than 1.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero.  (For example, **suspendEthernetInterface** method may be called for this test) |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_REPETITION\_BASE\_DELAY**: (Uint16) Base delay value in milliseconds [msec] configured for the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 6. TESTER: Verify received SOME/IP-SD message. 7. TESTER: Forces link-down on the DUT’s network interface until a period of 2 x **T\_REPETITION\_BASE\_DELAY** is elapsed. 8. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 9. TESTER: Verify received SOME/IP-SD message. 10. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 11. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received immediately after **T\_REPETITION\_BASE\_DELAY** +/- **TimingToleranceMillisec** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 3. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 4. SOME/IP-SD message is received immediately after **T\_REPETITION\_BASE\_DELAY** +/- **TimingToleranceMillisec** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0143, DS\_SD\_0144, DS\_SD\_0183 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_08] Main Phase: Find Service shall no longer be sent

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Checks that a client instance is not sending any Find Service entries during Main Phase of Service Discovery. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **N\_REPETITIONS**: (Uint8) Number of message repetitions configured for the SOME/IP-SD related Repetition Phase of the DUT.  **T\_MAIN\_PHASE\_OBSERVATION**: (Uint8) Duration in seconds [sec] to observe the SOME/IP-SD related Main Phase for Find Service Entry occurrences. Must be greater than the DUT’s configured value for SOME/IP parameter CYCLIC\_OFFER\_DELAY. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. Repeat preceding steps 3 and 4 for a total number of **N\_REPETITIONS**. 6. TESTER: Observes further DUT communication for a period of **T\_MAIN\_PHASE\_OBSERVATION**. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0143, DS\_SD\_0144, DS\_SD\_0163, DS\_SD\_0170 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_09] Main Phase: After Link-loss/Link-up jump to Initial Wait Phase

|  |  |
| --- | --- |
| Item | Description |
| Purpose | After Link-loss/Link-up jump to Initial Wait Phase. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP Parameter REPETITIONS\_MAX must not be zero.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero.  (For example, **suspendEthernetInterface** method may be called for this test) |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **N\_REPETITIONS**: (Uint8) Number of message repetitions configured for the SOME/IP-SD related Repetition Phase of the DUT.  **T\_REPETITION\_BASE\_DELAY**: (Uint16) Base delay value in milliseconds [msec] configured for the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. Repeat preceding steps 3 and 4 for a total number of **N\_REPETITIONS**. 6. TESTER: Waits until **TimeoutOfferService** is elapsed. 7. TESTER: Forces link-down on the DUT’s network interface until a period of **TimeoutOfferService** is elapsed. 8. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 9. TESTER: Verify received SOME/IP-SD message. 10. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 11. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 3. SOME/IP-SD message is received immediately after **T\_REPETITION\_BASE\_DELAY** +/- **TimingToleranceMillisec** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0143, DS\_SD\_0144, DS\_SD\_0183 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_10] Jump from Repetition Phase to Main Phase if Offer Service has been received

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Jump from Repetition Phase to Main Phase if Offer Service has been received. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP parameter REPETITIONS\_MAX should be greater than 1 . |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_REPETITION\_PHASE**: (Uint16) The expected overall duration in milliseconds [msec] of the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0**. 6. TESTER: Observes further DUT communication for a period of **T\_REPETITION\_PHASE**. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0119, DS\_SD\_0143, DS\_SD\_0144, DS\_SD\_0157 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_11] Skip Repetition Phase if max repetitions is set to zero

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Skip Repetition Phase if max repetitions is set to zero. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP parameter REPETITIONS\_MAX must be zero.  (For example, **suspendEthernetInterface** method may be called for this test) |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_MAIN\_PHASE\_OBSERVATION**: (Uint16) Duration to observe the Service Discovery related Main Phase for Find Service Entry occurrences. Must be greater than the DUT’s configured value for SOME/IP parameter REPETITIONS\_BASE\_DELAY. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Observes further DUT communication for a period of **T\_MAIN\_PHASE\_OBSERVATION**. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0143, DS\_SD\_0144, DS\_SD\_0162 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_12] Find Service shall not be sent when service is no longer offered

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to understand that the service it is looking for is no longer available. It should stop sending Find Service Entries for this service after reception of a respective Stop Offer Service Entry. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP parameter REPETITIONS\_MAX must be set to a value greater than zero. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_REPETITION\_PHASE**: (Uint16) Expected duration of the Service Discovery related Repetition Phase depending on the DUT’s configuration. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying a Stop Offer Service Entry for ETS instance of ETS\_InstanceID\_0. 6. TESTER: Observes further DUT communication for a period of **T\_REPETITION\_PHASE**. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0186 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_13] Request Response Delay for Subscribe as answer of multicast Offer Service

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Request Response Delay for Subscribe as answer of multicast Offer Service. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT related settings for REQUEST\_RESPONSE\_DELAY must define a minimum delay of at least 1 second [sec]. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds [sec]. Must not be zero.  **T\_RESPONSE\_DELAY\_MIN**: (Uint16) The minimal expected delay for answering to Service Discovery entries when DUT related setting of REQUEST\_RESPONSE\_DELAY is got involved.  **T\_RESPONSE\_DELAY\_MAX**: (Uint16) The maximal expected delay for answering to Service Discovery entries when DUT related setting of REQUEST\_RESPONSE\_DELAY is got involved. It is set to the Time of T\_SUBSCRIPTION**.** |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0**. 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 8. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received after **T\_RESPONSE\_DELAY\_MIN** but before **T\_RESPONSE\_DELAY\_MAX** and contains:    1. Entries Array provides entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER).    4. Entry: Eventgroup ID is equal to 0x0006. |
| Reference | DS\_SD\_0172 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_14] Request Response Delay shall not be applied if unicast message is answered

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Request Response Delay shall not be applied if unicast message is answered. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT related settings for REQUEST\_RESPONSE\_DELAY must define a minimum delay of at least 1 second [sec]. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds [sec]. Must not be zero.  **T\_RESPONSE\_DELAY\_MIN**: (Uint16) The minimal expected delay for answering to Service Discovery entries when DUT related setting of REQUEST\_RESPONSE\_DELAY is got involved. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends unicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0**. 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 8. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received before **T\_RESPONSE\_DELAY\_MIN** and contains:    1. Entries Array provides entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER).    4. Entry: Eventgroup ID is equal to 0x0006. |
| Reference | DS\_SD\_0173 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_15] Request Response Delay is randomly chosen between min and max

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Request Response Delay is randomly chosen between min and max. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT related settings for REQUEST\_RESPONSE\_DELAY must define different values for lower and upper limit. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds [sec]. Must not be zero.  **T\_RESPONSE\_DELAY\_MIN**: (Uint16) The minimal expected delay for answering to Service Discovery entries when DUT related setting of REQUEST\_RESPONSE\_DELAY is got involved.  **T\_RESPONSE\_DELAY\_MAX**: (Uint16) The maximal expected delay for answering to Service Discovery entries when DUT related setting of REQUEST\_RESPONSE\_DELAY is got involved. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0**. 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends unicast SOME/IP-SD message carrying the corresponding Subscribe Eventgroup Acknowledgement Entry. 10. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0**. 11. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:     * **delay** is set to 0x00000000 (0 seconds).     * **duration** is set to **T\_SUBSCRIPTION**.     * **eventgroupID** is set to 0x0005. 12. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0**. 13. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 14. TESTER: Verify received SOME/IP-SD message. 15. TESTER: Compares the measured response delay for SOME/IP-SD messages received in step 8 and 14. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received after **T\_RESPONSE\_DELAY\_MIN** but before **T\_RESPONSE\_DELAY\_MAX** and contains:    1. Entries Array provides entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER).    4. Entry: Eventgroup ID is equal to 0x0006. 3. SOME/IP-SD message is received after **T\_RESPONSE\_DELAY\_MIN** but before **T\_RESPONSE\_DELAY\_MAX** and contains:    1. Entries Array provides entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER).    4. Entry: Eventgroup ID is equal to 0x0005. 4. Measured delay values differ in at least 1 millisecond +/- **TimingToleranceMillissec**. |
| Reference | DS\_SD\_0174, DS\_SD\_0175 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_16] Same Request Response Delay within an ECU

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to use same Request Response Delay for all hosted client instances. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT related settings for REQUEST\_RESPONSE\_DELAY must define a fix delay of at least 1 second [sec] (minimum delay = maximum delay ). |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds [sec]. Must not be zero.  **T\_RESPONSE\_DELAY**: (Uint8) The expected delay for answering to Service Discovery entries when DUT related setting of REQUEST\_RESPONSE\_DELAY is got involved. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0**. 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSelectInstance** with parameter **instanceKey** set to 0x01 (Duplicated instance). 10. Repeat preceding steps 1 to 8(except 3 and 4) to send a second consecutive message with the same Request Response Delay. 11. TESTER: Verify measured Request Response Delay of step 8 and 10. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received before **TimeoutOfferService** and contains:    1. Entries Array provides entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER).    4. Entry: Eventgroup ID is equal to 0x0006. 3. Measured delay values are equal to **T\_RESPONSE\_DELAY** +/ **TimingTolerance**. |
| Reference | DS\_SD\_0178 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_17] Sending Unicast and Multicast use different Session ID counters

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to use a Session ID counter starting with 1 when sending the first SOME/IP-SD message of an unicast relation, while using a different Session ID counter for sending multicast messages. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT related Session ID counter must be < 1. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds [sec]. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends unicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance of ETS\_InstanceID\_0. 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0006. 8. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER).    4. Session ID is equal to 0x0001. 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER).    4. Entry: Eventgroup ID is equal to 0x0006.    5. Session ID is equal to 0x0001. |
| Reference | DS\_SD\_0250, DS\_SD\_0251 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_18] Each Unicast relation use different Session ID counters for sending

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Two server with different Endpoint options, communicate with the DUT. The session ID counter of each unicast message is different, cause both server are sending the Subscription for the DUT for the first time. Both session ID counters runs independently and the sending Session ID doesn’t increment. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  Two ETS Services Instances running at the ECU. Use a fresh unicast relation, so the session id counter start with 1. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS\_1**: IPv4 unicast address used as source address of the simulated SOME/IP server for the test.  **IP\_ADDRESS\_2**: Arbitrary IPv4 unicast address used for the test. Must be different than value of test variable IP\_ADDRESS\_1.  **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds [sec]. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends multicast SOME/IP-SD message from IP Endpoint address **IP\_ADDRESS\_1**, carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0**. 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0006. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSelectInstance** with parameter **instanceKey** set to 0x02 (Specific service version instance). 10. Repeat steps 1 to 5. 11. TESTER: Sends multicast SOME/IP-SD message from IP Endpoint address **IP\_ADDRESS\_2**, carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0**. 12. DUT: Sends unicast SOME/IP-SD carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0006. 13. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER).    4. Entry: Eventgroup ID is equal to 0x0006.    5. Session ID is equal to 0x0001. 3. Repeat step 4. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 4. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER).    4. Entry: Eventgroup ID is equal to 0x0006.    5. Session ID is equal to 0x0001. |
| Reference | DS\_SD\_0250, DS\_SD\_0251 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_19] Different Session ID counters for received Unicast and Multicast messages of same ECU

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Subscriber: Uses a simulated server instance (TESTER) and a client instance (DUT)  The Tester sends a unicast Offer Service with Session ID 1. This initialize the unicast Session Counter ID of the DUT with 1.  Then the DUT subscribe an Eventgroup and the TESTER sends an unicast subscribe ACK as answer with Session ID 2. This Session ID is expected. After this the Tester sends a multicast Offer service with Session ID 2.    If Unicast and multicast would count over a common Session ID counter, the second Session ID value would be a conflict because in this case it would be expected a Session ID value of 3,which would trigger a Reboot-Detection.  At least the DUT sends a subscribe again as direct answer for the multicast Offer service before. On the basis of the Initial Data Request Flag, it can be checked whether the Reboot-Detection has really triggered.  For an initial subscribe the Flag always is set to 1. For a following subscribe to prolonging the TTL the flag is set to 0. if a Reboot-Detection is triggered a  Flag of 1 is expected. If the flag is 0, the DUT it has reacted correctly.  So unicast and multicast are really counted by different Session Id counters. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds [sec]. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends unicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0** with Session ID set to 0x0001 and Reboot Flag set to 1. 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0006. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends unicast SOME/IP-SD message carrying the corresponding Subscribe Eventgroup Acknowledgement Entry with Session ID set to 0x0002 and Reboot Flag set to 1. 10. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0** with Session ID set to 0x0002 and Reboot Flag set to 1. 11. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0006. 12. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER).    4. Entry: Eventgroup ID is equal to 0x0006.    5. Entry: Initial Data Requested Flag is equal to 1. 3. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER).    4. Entry: Eventgroup ID is equal to 0x0006.    5. Entry: Initial Data Requested Flag is equal to 0. |
| Reference | DS\_SD\_0250, DS\_SD\_0252 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_20] Different Session ID counters per ECU for received Multicast messages

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Subscriber: Uses two simulated server instance (TESTER) and a client instance (DUT)  The Tester sends a multicast Offer Service with Session ID 1 for the first simulated server instance. This initialize the multicast Session Counter ID of the DUT with 1. Than the DUT subscribe an Eventgroup and the TESTER  sends a unicast subscribe ACK.  After this the Tester sends a multicast Offer service again with Session ID 1, but for the second simulated server instance.  If multicast for all ECUs would be counted over a common Session ID counter, the second Session ID value would be a conflict because in this case it would be expected a Session ID value of 2, which would trigger a Reboot-Detection.  After this the Tester sends a second multicast offer service for the first simulated server instance with the session ID value of 2.  At least the DUT sends a subscribe again as direct answer for the multicast Offer service before. On the basis of the Initial Data Request Flag it can be checked whether the Reboot-Detection has really triggered.  For an initial subscribe the Flag always is set to 1. For a following subscribe to prolonging the TTL the flag is set to 0.  if a Reboot-Detection is triggered a  Flag of 1 is expected. If the flag is 0, the DUT it has reacted correctly.  So multicast messages per ECU are really counted by different Session Id counters. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS\_1**: IPv4 unicast address used as source address of the simulated SOME/IP server for the test.  **IP\_ADDRESS\_2**: Arbitrary IPv4 unicast address used for the test. Must be different than value of test variable IP\_ADDRESS\_1.  **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds [sec]. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends multicast SOME/IP-SD message from IP Endpoint address **IP\_ADDRESS\_1**, carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0** with Session ID set to 0x0001 and Reboot Flag set to 1. 7. DUT: Sends unicast SOME/IP-SD carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0006. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends unicast SOME/IP-SD message carrying the corresponding Subscribe Eventgroup Acknowledgement Entry. 10. TESTER: Sends multicast SOME/IP-SD message from IP Endpoint address **IP\_ADDRESS\_2**, carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0** with Session ID set to 0x0001 and Reboot Flag set to 1. 11. TESTER: Sends multicast SOME/IP-SD message from IP Endpoint address **IP\_ADDRESS\_1**, carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0** with Session ID set to 0x0002 and Reboot Flag set to 1. 12. Sends unicast SOME/IP carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0006. 13. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER).    4. Entry: Eventgroup ID is equal to 0x0006.    5. Entry: Initial Data Requested Flag is equal to 1. 3. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER).    4. Entry: Eventgroup ID is equal to 0x0006.    5. Entry: Initial Data Requested Flag is equal to 0. |
| Reference | DS\_SD\_0250, DS\_SD\_0252 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_21] Different Session ID counters per ECU for received Unicast messages

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Subscriber: Uses two simulated server instance (TESTER) and a client instance (DUT)  The Tester sends a unicast Offer Service with Session ID 1 for the first simulated server instance. This initialize the unicast Session Counter ID of the DUT with 1. Than the DUT subscribe an Eventgroup and the TESTER sends a unicast subscribe ACK with Session ID 2.  After this the Tester sends a unicast Offer service again with Session ID 1, but for the second simulated server instance.  If unicast for all ECUs would be counted over a common Session ID counter, the second Session ID value would be a conflict because in this case it would be expected a Session ID value of 3, which would trigger a Reboot-Detection.  After this the Tester sends a second unicast Offer service for the first simulated server instance with the session ID value of 3.  At least the DUT sends a subscribe again as direct answer for the unicast Offer service before. On the basis of the Initial Data Request Flag it can be checked whether the Reboot-Detection has really triggered.  For an initial subscribe the Flag always is set to 1. For a following subscribe to prolonging the TTL the flag is set to 0.  if a Reboot-Detection is triggered a Flag of 1 is expected. If the flag is 0, the DUT it has reacted correctly. So unicast messages per ECU are really counted by different Session Id counters. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS\_1**: IPv4 unicast address used as source address of the simulated SOME/IP server for the test.  **IP\_ADDRESS\_2**: Arbitrary IPv4 unicast address used for the test. Must be different than value of test variable IP\_ADDRESS\_1.  **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds [sec]. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends unicast SOME/IP-SD message from IP Endpoint address **IP\_ADDRESS\_1**, carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0** with Session ID set to 0x0001 and Reboot Flag set to 1. 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0006. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends unicast SOME/IP-SD message carrying the corresponding Subscribe Eventgroup Acknowledgement Entry with Session ID set to 0x0002 and Reboot Flag set to 1. 10. TESTER: Sends unicast SOME/IP-SD message from IP Endpoint address **IP\_ADDRESS\_2**, carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0** with Session ID set to 0x0001 and Reboot Flag set to 1. 11. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0006. 12. TESTER: Verify received SOME/IP-SD message. 13. TESTER: Sends unicast SOME/IP-SD message from IP Endpoint address **IP\_ADDRESS\_1**, carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0** with Session ID set to 0x0003 and Reboot Flag set to 1. 14. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0006. 15. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER).    4. Entry: Eventgroup ID is equal to 0x0006.    5. Entry: Initial Data Requested Flag is equal to 1. 3. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER).    4. Entry: Eventgroup ID is equal to 0x0006.    5. Entry: Initial Data Requested Flag is equal to 0. 4. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER).    4. Entry: Eventgroup ID is equal to 0x0006.    5. Entry: Initial Data Requested Flag is equal to 0. |
| Reference | DS\_SD\_0250, DS\_SD\_0252 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_22] Ignore undefined bits in SD message header

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT ignore messages with undefined bits in the flag and Reserved fields in the SD message header. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_REPETITION\_PHASE**: (Uint16) The expected overall duration in milliseconds [msec] of the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message with all undefined bits in Flag and Reserved fields changed to 1, carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0**. 6. TESTER: Observes further DUT communication for a period of **T\_REPETITION\_PHASE**. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0032, DS\_SD\_0033 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_23] Ignore redundant options referenced by Offer Service

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT ignore messages with redundant options referenced by Offer Service. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_REPETITION\_PHASE**: (Uint16) The expected overall duration in milliseconds [msec] of the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0** which references an option in addition, which is a duplicate of a regular referenced IPv4 Endpoint Option. 6. TESTER: Observes further DUT communication for a period of **T\_REPETITION\_PHASE**. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0115 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_24] Ignore not required options referenced by Offer Service

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT ignore messages with not required options referenced by Offer Service. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: Arbitrary IPv4 multicast address used for the test.  **PORT\_NUMBER**: (Uint8) Arbitrary number of an UDP port used for the test.  **T\_REPETITION\_PHASE**: (Uint16) The expected overall duration in milliseconds [msec] of the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance of ETS\_InstanceID\_0which references the following option in addition:    * Type is set to 0x14 (IPv4 Multicast Option).    * Discardable Flag is set to 0.    * IPv4 Address is set to **IP\_ADDRESS**.    * L4-Proto is set to 0x11 (UDP).    * Port Number is set to **PORT\_NUMBER**. 6. TESTER: Observes further DUT communication for a period of **T\_REPETITION\_PHASE**. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0114 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_25] Ignore unknown option referenced by Offer Service

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT ignore messages withs unknown option in Subscribe Eventgroup. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UNKNOWN\_OPTION\_TYPE**: (Uint32) Type of SOME/IP-SD option which is unknown to the DUT.  **T\_REPETITION\_PHASE**: (Uint16) The expected overall duration in milliseconds [msec] of the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0** which references the following option in addition:    * Type is set to **UNKNOWN\_OPTION\_TYPE**.    * Length is set to 0x0005.    * Discardable Flag is set to 0.    * Reserved bits are set to 0.    * Remaining bytes of option are set to 0xDEADBEEF. 6. TESTER: Observes further DUT communication for a period of **T\_REPETITION\_PHASE**. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. No SOME/IP-SD messages were observed which contain:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0074, DS\_SD\_0113 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_26] Process Configuration Option referenced by Offer Service

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to deserialize content of Configuration Options and forward it to the application. In this test, the ETS stores the content for further verification into a field. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UTF8\_STRING**: Zero-terminated character string which represents a valid SOME/IP-SD Configuration Option string with arbitrary key/value-pairs.  **STRING\_LENGTH**: (Uint32) Length of configuration string. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance of ETS\_InstanceID\_0. In addition, the entry shall reference the following option:    * Type is set to 0x01 (Configuration Option).    * Length is set to **STRING\_LENGTH** + 8.    * Discardable Flag is set to 0.    * Reserved bits are set to 0.    * Configuration String is set to **UTF8\_STRING**. 6. TESTER: Sends SOME/IP Request message for getter of field **TestFieldUTF8DynamicReliable**. 7. DUT: Sends corresponding SOME/IP Response message. 8. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP Response message is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0560 (getter of field **TestFieldUTF8DynamicReliable**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UTF8 string equal to **UTF8\_STRING**. |
| Reference | DS\_SD\_0102 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_27] Answer Offer Service with setup from IPv4 SD Endpoint Option

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Answer Offer Service with setup from IPv4 SD Endpoint Option |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS**: IPv4 address used in the SD Endpoint Option for the test.  **PORT\_NUMBER**: (Uint8) Number of the UDP port used in the SD Endpoint Option for the test.  **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds [sec]. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter delay set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message carrying an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0**. Options Array includes the following option in addition:    * Type is set to 0x24 (IPv4 SD Endpoint Option).    * Discardable Flag is set to 0.    * IPv4 Address is set to **IP\_ADDRESS**.    * L4-Proto is set to 0x11 (UDP).    * Port Number is set to **PORT\_NUMBER**. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00000000 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 8. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received at IPv4 endpoint address **IP\_ADDRESS** and port **PORT\_NUMBER** before **TimeoutSubscribe** and contains:    1. Entries Array provides entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0006. |
| Reference | DS\_SD\_0087, DS\_SD\_0088 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_28] Ignore Offer Service with unknown Service ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT ignore Offer Service with unknown Service ID. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP parameter REPETITIONS\_MAX must not be zero.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UNKNOWN\_SERVICE\_ID**: (Uint16) Service ID which is unknown to DUT.  **T\_REPETITION\_PHASE**: (Uint16) The expected overall duration in milliseconds [msec] of the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message with an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0** where Service ID is changed to **UNKNOWN\_SERVICE\_ID**. 6. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 7. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received before **T\_REPETITION\_PHASE** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0194, DS\_SD\_0195, DS\_SD\_0196 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_29] Ignore Offer Service with unknown Instance ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT ignore Offer Service with unknown Instance ID |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP parameter REPETITIONS\_MAX must not be zero.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UNKNOWN\_INSTANCE\_ID**: (Uint16) An Instance ID different than the one used for the ETS.  **T\_REPETITION\_PHASE**: (Uint16) The expected overall duration in milliseconds [msec] of the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message with an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0** where Instance ID is changed to **UNKNOWN\_INSTANCE\_ID**. 6. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 7. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received before **T\_REPETITION\_PHASE** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0194, DS\_SD\_0195, DS\_SD\_0196 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_30] Ignore Offer Service with wrong Major Version

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT ignore Service with wrong Major Version |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP parameter REPETITIONS\_MAX must not be zero.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **WRONG\_MAJOR\_VERSION**: (Uint8) A major version number different than the one used for the ETS.  **T\_REPETITION\_PHASE**: (Uint16) The expected overall duration in milliseconds [msec] of the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message with an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0** where Major Version is changed to **WRONG\_MAJOR\_VERSION**. 6. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 7. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received before **T\_REPETITION\_PHASE** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0194, DS\_SD\_0195, DS\_SD\_0196 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_31] Ignore Offer Service without any Options

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT ignore Service without any Options |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP parameter REPETITIONS\_MAX must not be zero.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_REPETITION\_PHASE**: (Uint16) The expected overall duration in milliseconds [msec] of the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter delay set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message with an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0** without any referenced options. 6. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 7. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received before **T\_REPETITION\_PHASE** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0117, DS\_SD\_0194, DS\_SD\_0195, DS\_SD\_0196, DS\_SD\_0197 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_32] Ignore Offer Service with invalid Options

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT expects an invalid IPV4 Option, which is changed from the Tester and ignore the offer Service. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP parameter REPETITIONS\_MAX must not be zero.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UNKNOWN\_L4\_PROTO**: (Uint8) IANA/IETF number of a transport protocol which is not supported by SOME/IP.  **T\_REPETITION\_PHASE**: (Uint16) The expected overall duration in milliseconds [msec] of the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter delay set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message with an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0** where L4-Proto of referenced IPv4 Endpoint Option is changed to **UNKNOWN\_L4\_PROTO**. 6. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 7. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received before **T\_REPETITION\_PHASE** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0194, DS\_SD\_0195, DS\_SD\_0196, DS\_SD\_0197 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_33] Ignore Offer Service with conflict in Endpoint Options

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT expects a conflict in the Endpoint Options. For this the Port Number and IPv4 Address are different from this one used to offer the ETS Instance. So it describes a conflict. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **IP\_ADDRESS\_0**: IPv4 address where the Enhanced Testability Service is reachable.  **PORT\_NUMBER**: (Uint8) UDP port number different than the one used to offer the ETS instance for the test.  **T\_REPETITION\_PHASE**: (Uint16) The expected overall duration in milliseconds [msec] of the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter delay set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message with an Offer Service Entry for ETS instance of Instance **ETS\_InstanceID\_0** which references additionally to the default ETS related Endpoint Options the followig option:    * Type is set to 0x04 (IPv4 Endpoint Option)    * IPv4 Address is set to **IP\_ADDRESS**    * L4 Protol is set to 0x11 (UDP)    * Port Number is set to **PORT\_NUMBER**    * Discardable Flag is set to 0. 6. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 7. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received before **TimeoutFindService** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0116, DS\_SD\_0197 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_34] Ignore Offer Service if Session ID is set to zero

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT ignore Offer Service if Session ID is set to zero |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP parameter REPETITIONS\_MAX must not be zero.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_REPETITION\_PHASE**: (Uint16) The expected overall duration in milliseconds [msec] of the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter delay set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message with an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0** where Session ID is changed to 0x0000. 6. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 7. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received before **T\_REPETITION\_PHASE** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0013 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_35] Ignore Offer Service if Unicast flag is set to zero

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT ignore Offer Service if Unicast flag is set to zero. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  SOME/IP parameter REPETITIONS\_MAX must not be zero.  SOME/IP parameter REPETITIONS\_BASE\_DELAY must not be zero. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_REPETITION\_PHASE**: (Uint16) The expected overall duration in milliseconds [msec] of the SOME/IP-SD related Repetition Phase of the DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter delay set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends unicast SOME/IP-SD message with an Offer Service Entry for ETS instance of **ETS\_InstanceID\_0** where Unicast Flag is changed to 0. 6. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 7. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received before **T\_REPETITION\_PHASE** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0029 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_36] Client send Find Service entries for specific Service version

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send Find Service Entries for a specific major version of the service. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | none |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSelecInstance** with parameter **instanceKey** set to 0x02 (Specific service version instance). 2. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 3. DUT: Activates client service instance and enters Service Discovery start-up sequence. 4. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 5. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER).    4. Entry: Major Version is equal to 0x02. |
| Reference | DS\_SD\_0253 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_37] Client send Find Service entries with wildcard Service version

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send Find Service Entries for a service without any preferences in major version. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | none |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER).    4. Entry: Major Version is equal to 0xFF. |
| Reference | DS\_SD\_0253 |
| Notes |  |

#### [SOMEIPCLT\_SD\_BEHAVIOR\_38] Process multiple Offer Service Entries per SOME/IP-SD message

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to stop sending find service entries in respond to corresponding offer service entries which are received together in one message. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. TESTER: Sends SOME/IP Request message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSelectInstance** with parameter **instanceKey** set to 0x02 (Specific service version instance). 6. TESTER: Sends SOME/IP Request message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 7. DUT: Activates client service instance and enters Service Discovery start-up sequence. 8. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 9. TESTER: Verify received SOME/IP-SD message. 10. TESTER: Sends multicast SOME/IP-SD message carrying two Offer Services Entry for ETS instance of **ETS\_InstanceID\_0**. 11. TESTER: Observes further DUT communication for **TimeoutOfferService.** |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 2. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). 3. No SOME/IP-SD messages are received which contain:    1. Entries Array provides entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** (ETS instance simulated by TESTER). |
| Reference | DS\_SD\_0253 |
| Notes |  |

### Service Discovery Publish/Subscribe Communication

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_01] Subscribe Eventgroup is triggered by Offer Service

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expect a Subscribe Eventgroup from the DUT when it is triggered by an Offer Service |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must be at least a value of 2 seconds. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 8. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of Type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Eventgoup ID is equal to 0x0006. |
| Reference | DS\_SD\_0171, DS\_SD\_0198 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_02] Subscribe Eventgroup is sent for each Offer Service

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expect that a Subscribe Eventgroup is send from the DUT for each Offer Service which is send from the Server. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 10. Repeat preceding steps 5 to 9 for three more times. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgoup ID is equal to 0x0006.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 3. Each SOME/IP-SD message is received before its respective **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgoup ID is equal to 0x0006.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** |
| Reference | DS\_SD\_0205 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_03] Subscribe Eventgroup is only send if client is interested in Events

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The Tester expect that a Subscribe Eventgroup is only send if client is interested in Events. In this Case the Client isn’t interest and so while the observes time, there don’t may receive a Subscribe Eventgroup Entry. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 7. TESTER: Verify received SOME/IP-SD message. 8. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 9. TESTER: Waits until a period of **T\_SUBSCRIPTION** is elapsed. 10. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 11. TESTER: Observes further DUT communication for a period of **TimeoutSubscribe**. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgoup ID is equal to 0x0006.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0..** 3. No SOME/IP-SD messages have been observed which carried an entry with the following parameters:    1. Type is equal to 0x06 (Subscribe Eventgroup Entry).    2. Service ID is equal to **ETS\_ServiceID\_0**.    3. Eventgroup ID is equal to 0x0006.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0..** |
| Reference | DS\_SD\_0130, DS\_SD\_0204 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_04] Subscribe Eventgroup is sent before TTL expires

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT shall send a Subscribe Eventgroup before the TTL expires |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for ETS **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to 2 x **ETS\_SubscriptionLifetime**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 10. TESTER: Waits until **ETS\_SubscriptionLifetime** is elapsed. 11. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 12. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgoup ID is equal to 0x0006.    4. Entry: TTL is equal to **ETS\_SubscriptionLifetime**.    5. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 3. SOME/IP-SD message is received before **ETS\_SubscriptionLifetime** is elapsed and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgoup ID is equal to 0x0006.    4. Entry: TTL is equal to **ETS\_SubscriptionLifetime**.    5. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** |
| Reference | DS\_SD\_0208 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_05] Subscribe Eventgroup is sent when reboot of server has been detected

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to detect the reboot of the Sever and renew the existing Subscription by sending subscribe eventgroup entry immediately. For the test its required that the first Offer and subscribe eventgroup is done normally.  And after it the DUT send a prepared subscribe eventgroup where the duration is set to the maximum value (TTL is set to maximum). Now the Tester sends a prepared Offer Service where the Reboot Flag is set to 1.  So the Server has reboot. Because the TTL is set to the maximum value normally the Tester doesn’t expect a new subscribe eventgroup from the DUT. But because of the Server reboot, which trigger the subscribe eventgroup from the DUT. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UNKNOWN\_SERVICE\_ID**: (Uint16) Service ID which is unknown to DUT. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds).   1. DUT: Activates client service instance and enters Service Discovery start-up sequence. 2. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 3. TESTER: Verify received SOME/IP-SD message. 4. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 6. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 7. TESTER: Verify received SOME/IP-SD message. 8. TESTER: Sends unicast SOME/IP-SD message with the corresponding Subscribe Eventgroup Acknowledgement Entry. 9. TESTER: Sends multicast SOME/IP-SD message with Reboot Flag set to 1 and Session ID set to 0x0001 carrying an Offer Service Entry. 10. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 11. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0006.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 3. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0006.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    5. Entry: Initial request flag is equal to 1. |
| Reference | DS\_SD\_0208 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_06] Subscribe Eventgroup is sent as unicast

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Subscribe Eventgroup is send from the DUT as unicast. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 8. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. IP Destination Address is a unicast address.    2. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    3. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    4. Entry: Eventgroup ID is equal to 0x0006.    5. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** |
| Reference | DS\_SD\_0209 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_07] TCP connection shall be opened before sending Subscribe Eventgroup

|  |  |
| --- | --- |
| Item | Description |
| Purpose | TCP connection shall be opened before sending Subscribe Eventgroup. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT have no existing TCP connection at the beginning |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **TIME\_SUBSCRIPTION\_1**: (Uint8) Duration of Eventgroup subscription in seconds. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00.    * **duration** is set to **TIME\_SUBSCRIPTION\_1**.    * **eventgroupID** is set to 0x0002. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 7. DUT: Try to establish TCP connection by sending SYN. 8. TESTER: Accepts TCP connection by sending SYN + ACK. 9. DUT: Sends unicast SOME/IP-SD Subscribe Eventgroup entry. 10. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. TCP SYN is received within **TimeoutSubscribe**. 3. SOME/IP-SD message is received within **TimeoutSubscribe** and contain:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: TTL is not Equal 0x0000.    5. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** |
| Reference | DS\_SD\_0207, DS\_SD\_0239 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_08] Check TCP connection if receiving Subscribe Eventgroup NACK

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Check TCP connection if receiving Subscribe Eventgroup NACK |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **TIME\_SUBSCRIPTION\_1**: (Uint8) Duration of Eventgroup subscription in seconds. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00.    * **duration** is set to **TIME\_SUBSCRIPTION\_1**.    * **eventgroupID** is set to 0x0002. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 7. DUT: Sends unicast SOME/IP-SD message with Subscribe Eventgroup Entry of type 0x0002. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends unicast SOME/IP-SD message with the corresponding Subscribe Eventgroup NAcknowledgement Entry. 10. DUT: Check existing TCP connections by sending SYN + ACK. 11. TESTER: Verify established TCP connection. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides one entry.    2. Entry: Eventgroup ID is equal to 0x0002.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry). 3. TCP SYN + ACK is received within **TimeoutCloseTcpConnection.** |
| Reference | DS\_SD\_0142 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_09] Reception of event notification via TCP

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to understand a SOME/IP Notification message which is received via TCP binding. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero.  **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0002. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 10. TESTER: Sends SOME/IP Notification message for ETS event **TestEventUINT8Reliable** with payload set to **UINT8\_VALUE**. 11. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastValueOfEventTCP**. 12. DUT: Sends corresponding SOME/IP Response message. 13. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0002.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 3. SOME/IP Response message is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x003B (**clientServiceGetLastValueOfEventTCP**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **UINT8\_VALUE**. |
| Reference | DS\_SP\_0091, DS\_SD\_0238 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_10] Reception of event notification via UDP unicast

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to understand a SOME/IP Notification message which is received via unicast UDP binding. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero.  **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0005. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 10. TESTER: Sends SOME/IP Notification message for ETS event **TestEventUINT8** with payload set to **UINT8\_VALUE**. 11. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastValueOfEventUDPUnicast**. 12. DUT: Sends corresponding SOME/IP Response message. 13. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0005.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 3. SOME/IP Response message is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x003C (**clientServiceGetLastValueOfEventUDPUnicast**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **UINT8\_VALUE**. |
| Reference | DS\_SP\_0091, DS\_SD\_0238 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_11] Reception of event notification via UDP multicast

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to understand a SOME/IP Notification message which is received via multicast UDP binding. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero.  **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 10. TESTER: Sends SOME/IP Notification message for ETS event **TestEventUINT8Multicast** with payload set to **UINT8\_VALUE**. 11. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastValueOfEventUDPMulticast**. 12. DUT: Sends corresponding SOME/IP Response message. 13. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Eventgroup ID is equal to 0x0006.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 3. SOME/IP Response message is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    3. Method ID is equal to 0x003D (**clientServiceGetLastValueOfEventUDPMulticast**).    4. Return Code is equal to 0x00 (E\_OK).    5. Payload provides an UINT8 value equal to **UINT8\_VALUE**. |
| Reference | DS\_SP\_0091, DS\_SD\_0243 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_12] Client ignore invalid event notification (no error msg)

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to not send any error messages in response of a received event notification message. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 10. TESTER: Sends SOME/IP Notification message with modified Protocol Version set to **WRONG\_PROTOCOL\_VERSION**. 11. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError**. 12. DUT: Sends corresponding SOME/IP Response message. 13. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Eventgroup ID is equal to 0x0006.    3. Entry: TTL is not Equal 0x0000.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 3. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides UINT8 value equal to 0x00 (E\_OK). |
| Reference | DS\_SP\_0104 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_13] Client replicates received Event notification for all other Clients on same ECU

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Client replicates received Event notification for all other Clients on same ECU |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds.  **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry for Eventgroup ID 0x0006. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 10. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSelectInstance** with parameter **instanceKey** set to 0x01 (Duplicated instance). 11. Repeat steps 1 to 9(except 3 and 4). 12. TESTER: Sends SOME/IP Notification message for ETS event **TestEventUINT8Multicast** with payload set to **UINT8\_VALUE**. 13. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastValueOfEventUDPMulticast**. 14. DUT: Sends corresponding SOME/IP Response message. 15. TESTER: Verify received SOME/IP Response message. 16. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSelectInstance** with parameter **instanceKey** set to 0x00 (Default instance). 17. Repeat steps 13 to 15. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Eventgroup ID is equal to 0x0006.    3. Entry: TTL is not equal to 0x0000.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 3. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x003D (**clientServiceGetLastValueOfEventUDPMulticast**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides UINT8 value equal to **UINT8\_VALUE**. 4. SOME/IP Response message is received within **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x003D (**clientServiceGetLastValueOfEventUDPMulticast**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides UINT8 value equal to **UINT8\_VALUE**. |
| Reference | DS\_SP\_0093 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_14] Stop Subscribe Eventgroup is sent to unsubscribe event

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to notify the server that it will no longer subscribe to an Eventgroup after the Duration Timed out. It shall send out a Stop Subscribe Eventgroup Entry for the subscribed Eventgroup. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 10. TESTER: Waits until **T\_SUBSCRIPTION** is elapsed. 11. DUT: Sends unicast SOME/IP-SD message carrying a Stop Subscribe Eventgroup Entry. 12. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0**. 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Eventgroup ID is equal to 0x0006. 3. SOME/IP-SD message is received before **TimeoutStopSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Eventgroup ID is equal to 0x0006.    5. Entry: TTL is equal to 0x0000. |
| Reference | DS\_SD\_0133, DS\_SD\_0206 |
| Notes | The ETS is defined in this way, that it holds the Subscription so long as it timed out. If the Duration of the Subscription timed out, the Stop subscription will be sent automatically. |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_15] Stop Subscribe Eventgroup is sent as unicast

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to notify the server that it will no longer subscribe to an Eventgroup after the Duration Timed out.  So stop Subscribe Eventgroup is send from the DUT as unicast. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 10. TESTER: Waits until **T\_SUBSCRIPTION** is elapsed. 11. DUT: Sends unicast SOME/IP-SD message carrying a Stop Subscribe Eventgroup Entry. 12. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Eventgroup ID is equal to 0x0006. 3. SOME/IP-SD message is received before **TimeoutStopSubscribe** and contains:    1. IP Destination Address is not a multicast address.    2. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    3. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    5. Entry: Eventgroup ID is equal to 0x0006.    6. Entry: TTL is equal to 0x0000. |
| Reference | DS\_SD\_0209 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_16] Unsubscribe Eventgroup if client service is being stopped

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to notify the server that it will no longer subscribe to an Eventgroup when client service instance has been stopped. It shall send out a Stop Subscribe Eventgroup Entry for the subscribed Eventgroup. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** of ETS instance **ETS\_InstanceI** with the following parameters:    * **delay** is set to 0x00.    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 8. TESTER: Verify received SOME/IP-SD messages. 9. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 10. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceDeactivate** with parameter **delay** set to 0x00 (0 seconds). 11. DUT: Sends unicast SOME/IP-SD message carrying a Stop Subscribe Eventgroup Entry. 12. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Eventgroup ID is equal to 0x0006. 3. SOME/IP-SD message is received before **TimeoutStopSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Eventgroup ID is equal to 0x0006.    5. Entry: TTL is equal to 0x0000. |
| Reference | DS\_SD\_0187 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_17] Unsubscribe Eventgroups if Stop Offer Service has been received

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected that if a Stop Offer Service has been received from the tester, that the DUT Unsubscribe the subscribed Eventgroups |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** of ETS instance **ETS\_InstanceI** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set to **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 10. TESTER: Sends multicast SOME/IP-SD message carrying a Stop Offer Service Entry for **ETS\_InstanceID\_0.** 11. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 12. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 13. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Eventgroup ID is equal to 0x0006. 3. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Eventgroup ID is equal to 0x0006.    5. Entry: Initial Data Requested Flag is set to 1. |
| Reference | DS\_SD\_0185 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_18] Unsubscribe all Eventgroups if ECU is being shut down

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Unsubscribe all Eventgroups if ECU is being shut down |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **TIME\_SUBSCRIPTION\_1**: (Uint8) Duration of Eventgroup subscription in seconds. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay set to 0x00 (0 seconds).** 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00.    * **duration** is set to **TIME\_SUBSCRIPTION\_1**.    * **eventgroupID** is set to 0x0005. 7. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00.    * **duration** is set to **TIME\_SUBSCRIPTION\_1**.    * **eventgroupID** is set to 0x0006. 8. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 9. DUT: Sends unicast two SOME/IP-SD Subscribe Eventgroups entry of 0x0005 and 0x0006. 10. TESTER: Verify received SOME/IP-SD messages. 11. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceDeactivate** with parameter **delay** set to 0x00 (0 seconds). 12. DUT: Sends SOME/IP-SD message carrying two Stop Subscribe Eventgroup Entrys for 0x0005 and 0x0006. 13. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Eventgroup ID is equal to 0x0005.    5. Entry: Eventgroup ID is equal to 0x0006. 3. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Eventgroup ID is equal to 0x0005.    5. Entry: Eventgroup ID is equal to 0x0006.    6. Entry: TTL is equal to 0x0000. |
| Reference | DS\_SD\_0188 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_19] Send Subscribe with Initial Data Requested Flag set to 1

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to set the Initial Data Requested Flag of the next Subscribe Eventgroup Entry to 1 when no Subscribe Eventgroup Acknowledgement Entry has been received for a preceeding Subscribe Eventgroup Entry. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT must support Explicit Initial Data Control functionality. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00.    * **duration** is set to 0xFFFF.    * **eventgroupID** is set to 0x0006. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 8. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 9. TESTER: Verify received SOME/IP-SD message. 10. TESTER: Waits for **TimeoutOfferService**. 11. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 12. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 13. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 3. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Eventgroup ID is equal to 0x0006.    5. Entry: Initial Data Requested Flag is set to 1. |
| Reference | DS\_SD\_0214 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_20] Send Stop and Subscribe in same message

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send a Stop Subscribe Eventgroup Entry and a Subscribe Eventgroup Entry together in one message when no Subscribe Eventgroup Acknowledgement Entry has been received for a preceeding Subscribe Eventgroup Entry. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  Explicit Initial Data Control functionality is not supported by DUT. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | None |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00.    * **duration** is set to 0xFFFF.    * **eventgroupID** is set to 0x0006. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 8. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 9. TESTER: Verify received SOME/IP-SD message. 10. TESTER: Waits for **TimeoutOfferService**. 11. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 12. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup and a Stop Subscribe Eventgroup Entry. 13. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 3. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides two entries of type 0x06 (Subscribe Eventgroup Entry).    2. Entry 1: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry 1: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry 1: Eventgroup ID is equal to 0x0006.    5. Entry 1: TTL is equal to 0x0000.    6. Entry 2: Service ID is equal to **ETS\_ServiceID\_0**.    7. Entry:2: Instance ID is equal to **ETS\_InstanceID\_0.**    8. Entry 2: Eventgroup ID is equal to 0x0006.    9. Entry 2: TTL is equal to **ETS\_SubscriptionLifetime**. |
| Reference | DS\_SD\_0214 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_21] Timer for sending Subscribe Eventgroup is reset

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Whenever the DUT is sending a Subscribe Eventgroup, after receiving an Offer Service entry, the Timer for sending Subscribe Eventgroup of the DUT is to reset. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | none |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00.    * **duration** is set to 2 x **ETS\_SubscriptionLifetime**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 7. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 8. TESTER: Verify received SOME/IP-SD message. 9. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 10. TESTER: Waits until a period of ½ **ETS\_SubscriptionLifetime** is elapsed. 11. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 12. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 13. TESTER: Verify received SOME/IP-SD message. 14. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 15. TESTER: Observes further DUT communication until a period of **ETS\_SubscriptionLifetime** is elapsed. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Eventgroup ID is equal to 0x0006.    5. Entry: TTL is equal to **ETS\_SubscriptionLifetime**. 3. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Eventgroup ID is equal to 0x0006.    5. Entry: TTL is equal to **ETS\_SubscriptionLifetime**. 4. No SOME/IP-SD message were observed which contain:    1. Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Eventgroup ID is equal to 0x0006.    5. Entry: TTL is equal to **ETS\_SubscriptionLifetime**. |
| Reference | DS\_SD\_0210 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_23] Subscribe Eventgroup is sent again after Link-Loss/Link-up

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Subscriptions shall be deleted if link on Ethernet interface is lost.  Subscribe Eventgroup is sent again after Link-Loss/Link-up. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT must support Explicit Initial Data Control functionality.  (For example, **suspendEthernetInterface** method may be called for this test) |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **TIME\_SUBSCRIPTION\_1**: (Uint8) Duration of Eventgroup subscription in seconds. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00.    * **duration** is set to 2\***TIME\_SUBSCRIPTION\_1**.    * **eventgroupID** is set to is set to 0x0006. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 8. DUT: Sends unicast SOME/IP-SD Subscribe Eventgroup entry of 0x0006. 9. TESTER: Verify received SOME/IP-SD message. 10. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 11. TESTER: Forces a link loss on DUT-side for duration of 1/2\***TIME\_SUBSCRIPTION\_1**. 12. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 13. TESTER: Verify received SOME/IP-SD message. 14. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 15. DUT: Sends unicast SOME/IP-SD Subscribe Eventgroup entry. 16. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides one entry.    2. Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry).    3. Entry: Service ID is equal to **ETS\_ServiceID\_0.**    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 3. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 4. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides one entry.    2. Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry).    3. Entry: Service ID is equal to **ETS\_ServiceID\_0.**    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    5. Entry: Initial Data Requested Flag is set to 1.    6. Entry: Reboot Flag is equal to 1.    7. Entry: Session ID is equal to 1. |
| Reference | DS\_SD\_0222, DS\_SD\_0225 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_24] Subscription shall be renewed if Reboot of server is detected

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Subscription shall be renewed if Reboot of server is detected |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **TIME\_SUBSCRIPTION\_1**: (Uint8) Duration of Eventgroup subscription in seconds. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00.    * **duration** is set to **TIME\_SUBSCRIPTION\_1**.    * **eventgroupID** is set to 0x0006. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 8. DUT: Sends unicast SOME/IP-SD Subscribe Eventgroup entry of 0x0006. 9. TESTER: Verify received SOME/IP-SD message. 10. Refuse TCP Connection. 11. TESTER: Sends unicast SOME/IP-SD message with Reboot Flag set to 1 and Session ID set to 0x0001, carrying an Offer Service Entry for the ETS. 12. DUT: Resets TCP Connection by sending SYN. 13. TESTER: Accepts TCP connection by sending SYN + ACK. 14. DUT: DUT: Sends unicast SOME/IP-SD Subscribe Eventgroup entry. 15. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received within **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides one entry.    2. Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry).    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 3. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides one entry.    2. Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry).    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** |
| Reference | DS\_SD\_0024, DS\_SD\_0025, DS\_SD\_0026 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_25] Counter is used to differentiate multiple Subscribe Eventgroups for same Eventgroup from same ECU

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to use different values for Counter in the Subscribe Eventgroup Entries when two client instances subscribe for the same Eventgroup. |
| Preconditions | DUT is running and offering the Enhanced Testability Service in two instances which only differs by its Endpoint Options. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates first client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    1. **delay** is set to 0x00 (0 seconds).    2. **duration** is set **T\_SUBSCRIPTION**.    3. **eventgroupID** is set to 0x0006. 7. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSelectInstance** with parameter **instanceKey** set to 0x01 (Duplicated instance). 8. DUT: Sends unicast SOME/IP-SD Subscribe Eventgroup entry of 0x0006. 9. TESTER: Verify received SOME/IP-SD message. 10. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 11. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 12. DUT: Activates second client service instance. 13. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:     1. **delay** is set to 0x00 (0 seconds).     2. **duration** is set **T\_SUBSCRIPTION**.     3. **eventgroupID** is set to 0x0006. 14. DUT: Sends unicast SOME/IP-SD Subscribe Eventgroup entry of 0x0006. 15. TESTER: Verify received SOME/IP-SD message. 16. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. 17. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 18. DUT: Sends unicast SOME/IP-SD message carrying two Subscribe Eventgroup Entries. 19. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides one entry.    2. Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry).    3. Entry: Service ID is equal to **ETS\_ServiceID\_0.**    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 3. SOME/IP-SD message is received within **TimeoutSubscribe** and contains:    1. Entries Array provides one entry.    2. Entry: Type is equal to 0x06 (Subscribe Eventgroup Entry).    3. Entry: Service ID is equal to **ETS\_ServiceID\_0.**    4. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 4. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides two entries of type 0x06 (Subscribe Eventgroup Entry).    2. Both Entries: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Both Entries: Eventgroup ID is equal to 0x0006.    5. Both Entries: TTL is not equal to 0x0000.    6. 1st Entry Counter is different from 2nd Entry Counter. |
| Reference | DS\_SD\_0069 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_26] Request is sent to static address without Offer Service

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Request is sent to static address without Offer Service. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSelectInstance** with parameter **instanceKey** set to 0x03 (Static configuration instance). 2. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 3. DUT: Activates client service instance. 4. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **value** is set to **UINT8\_VALUE**. 5. DUT: Sends SOME/IP Request message for method **echoUINT8**. 6. TESTER: Verify received SOME/IP Request message. |
| Verification | 1. SOME/IP Request message is received within **TimeoutRequest** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**). |
| Reference | DS\_SD\_0226, DS\_SD\_0227 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_27] Events are received from static address without Subscribe Eventgroup

|  |  |
| --- | --- |
| Item | Description |
| Purpose | Events are received from static address without Subscribe Eventgroup. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSelectInstance** with parameter **instanceKey** set to 0x03 (Static configuration instance). 2. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 3. DUT: Activates client service instance. 4. TESTER: Sends SOME/IP Notification message for ETS event **TestEventUINT8Multicast** with payload set to **UINT8\_VALUE**. 5. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastValueOfEventUDPMulticast**. 6. DUT: Sends corresponding SOME/IP Response message. 7. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x003D (**clientServiceGetLastValueOfEventUDPMulticast**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **UINT8\_VALUE**. |
| Reference | DS\_SD\_0226, DS\_SD\_0227 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_28] Static Endpoint config of service will be overwritten if SD provides different settings

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to send SOME/IP Request message to an updated IP endpoint of the service instead of the statically configured one, after an Offer Service Entry with appropriate endpoint settings has been received. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: 8-bit unsigned integer used as arbitrary payload for the test.  **PORT\_NUMBER**: (Uint8) Number of an UDP port used as new IP endpoint which is announced by Service Discovery. Must be different than ETS\_StaticPortUDP. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSelectInstance** with parameter **instanceKey** set to 0x03 (Static configuration instance). 2. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 3. DUT: Activates client service instance. 4. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **value** is set to **UINT8\_VALUE**. 5. DUT: Sends SOME/IP Request message for method **echoUINT8**. 6. TESTER: Verify received SOME/IP Request message. 7. TESTER: Sends corresponding SOME/IP Response message. 8. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_2.** Referenced IPv4 Endpoint Option for UDP binding announces the IP endpoint **ETS\_StaticIPv4Address:PORT\_NUMBER**. 9. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceCallEchoUINT8** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **value** is set to **UINT8\_VALUE**. 10. DUT: Sends SOME/IP Request message for method **echoUINT8**. 11. TESTER: Verify received SOME/IP Request message. |
| Verification | 1. SOME/IP Request message is received at IP endpoint **ETS\_StaticIPv4Address:ETS\_StaticPortUDP** before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**).    3. Payload provides an UINT8 value equal to **UINT8\_VALUE**. 2. SOME/IP Request message is received at IP endpoint **ETS\_StaticIPv4Address:PORT\_NUMBER** before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x0008 (**echoUINT8**).    3. Payload provides an UINT8 value equal to **UINT8\_VALUE**. |
| Reference | DS\_SD\_0228 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_29] Static Endpoint config of events will be overwritten if SD provides different settings

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to receive and process multicast event notifications which have been sent to an updated IP multicast address instead of the statically configured one. The new address is announced by regular Subscribe Eventgroup Acknowledgement Entry. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE\_1**: Unsigned integer value used as arbitrary payload for the test.  **UINT8\_VALUE\_2**: Unsigned integer value used as arbitrary payload for the test. Shall be different than **UINT8\_VALUE\_1.**  **MULTICAST\_PORT\_NUMBER:** (Uint16) Number of an UDP port used as new destination port for multicast event notifications which is announced by Service Discovery. Must be different than ETS\_StaticPortUDP.  **MULTICAST\_IP\_ADDRESS:** IPv4 multicast address used as new IP endpoint which is announced by Service Discovery. Must be different than ETS\_StaticIPv4MulticastAddress. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSelectInstance** with parameter **instanceKey** set to 0x03 (Static configuration instance). 2. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 3. DUT: Activates client service instance. 4. TESTER: Sends SOME/IP Notification message for ETS event **TestEventUINT8Multicast** with payload set to **UINT8\_VALUE\_1**. Destination address is **ETS\_StaticIPv4MulticastAddress:ETS\_StaticMulticastPort**. 5. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastValueOfEventUDPMulticast**. 6. DUT: Sends corresponding SOME/IP Response message. 7. TESTER: Verify received SOME/IP Response message. 8. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_2** which references an IP endpoint option for **MULTICAST\_IP\_ADDRESS** and **MULTICAST\_PORT\_NUMBER.** 9. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 10. TESTER: Verify received SOME/IP-SD message. 11. TESTER: Sends unicast SOME/IP-SD message with corresponding Subscribe Eventgroup Acknowledgement Entry. Referenced IPv4 Multicast Option announces the address **MULTICAST\_IP\_ADDRESS: MULTICAST\_PORT\_NUMBER**. 12. TESTER: Sends SOME/IP Notification message for ETS event **TestEventUINT8Multicast** with payload set to **UINT8\_VALUE\_2**. Destination address is **MULTICAST\_IP\_ADDRESS:MULTICAST\_PORT\_NUMBER**. 13. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastValueOfEventUDPMulticast**. 14. DUT: Sends corresponding SOME/IP Response message. 15. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response message is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x003D (**clientServiceGetLastValueOfEventUDPMulticast**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **UINT8\_VALUE\_1**. 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_2**.    4. Entry: Eventgoup ID is equal to 0x0006.    5. Entry: TTL is equal to **ETS\_SubscriptionLifetime**. 3. SOME/IP Response message is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID\_0**.    2. Method ID is equal to 0x003D (**clientServiceGetLastValueOfEventUDPMulticast**).    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides an UINT8 value equal to **UINT8\_VALUE\_2**. |
| Reference | DS\_SD\_0228 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_30] Client subscribe Eventgroup of explicit Service version

|  |  |
| --- | --- |
| tem | Description |
| Purpose | The DUT is expected to address explicit a particular service version with a Subscribe Eventgroup Entry. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSelectInstance** with parameter **instanceKey** set to 0x02 (Specific service version instance). 2. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 3. DUT: Activates first client service instance and enters Service Discovery start-up sequence. 4. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 5. TESTER: Verify received SOME/IP-SD message. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 7. TESTER: Sends multicast SOME/IP-SD message carrying an Offer Service Entry for **ETS\_InstanceID\_0.** 8. DUT: Sends unicast SOME/IP-SD message carrying a Subscribe Eventgroup Entry. 9. TESTER: Verify received SOME/IP-SD message. |
| Verification | 1. SOME/IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Eventgroup ID is equal to 0x0006.    5. Entry: TTL is not equal to 0x0000.    6. Entry: Major Version is equal to 0x02. |
| Reference | DS\_SD\_0247 |
| Notes |  |

#### [SOMEIPCLT\_SD\_SUBSCRIBE\_31] Client SD Entries has same Service and Instance IDs but different Major Versions

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to use different values for Major Version, but same for Service and Instance ID, when client instances addressing same service with different requirements on its major version. |
| Preconditions | DUT is running and offering the Enhanced Testability Service. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_SUBSCRIPTION**: (Uint8) Duration of Eventgroup subscription in seconds. Must not be zero. |
| Test Procedure | 1. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 2. DUT: Activates first client service instance and enters Service Discovery start-up sequence. 3. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 4. TESTER: Verify received SOME/IP-SD message. 5. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:    * **delay** is set to 0x00 (0 seconds).    * **duration** is set **T\_SUBSCRIPTION**.    * **eventgroupID** is set to 0x0006. 6. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSelectInstance** with parameter **instanceKey** set to 0x02 (Specific service version instance). 7. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceActivate** with parameter **delay** set to 0x00 (0 seconds). 8. DUT: Activates second client service instance. 9. DUT: Sends multicast SOME/IP-SD message carrying a Find Service Entry. 10. TESTER: Verify received SOME/IP-SD message. 11. TESTER: Sends multicast SOME/IP-SD message carrying two Offer Service Entry for **ETS\_InstanceID\_0** with different Major Versions. 12. TESTER: Sends SOME/IP Request-No-Return message for method **clientServiceSubscribeEventgroupSingle** with the following parameters:     * **delay** is set to 0x00 (0 seconds).     * **duration** is set **T\_SUBSCRIPTION**.     * **eventgroupID** is set to 0x0006. 13. DUT: Sends two unicast SOME/IP-SD message carrying Subscribe Eventgroup Entry. 14. TESTER: Verify received two SOME/IP-SD messages. |
| Verification | 1. SOME /IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 2. SOME /IP-SD message is received before **TimeoutStartupSD** and contains:    1. Entries Array provides an entry of type 0x00 (Find Service Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.** 3. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Eventgroup ID is equal to 0x0006.    5. Entry: TTL is not equal to 0x0000.    6. Entry: Major Version is equal to 0x01. 4. SOME/IP-SD message is received before **TimeoutSubscribe** and contains:    1. Entries Array provides an entry of type 0x06 (Subscribe Eventgroup Entry).    2. Entry: Service ID is equal to **ETS\_ServiceID\_0**.    3. Entry: Instance ID is equal to **ETS\_InstanceID\_0.**    4. Entry: Eventgroup ID is equal to 0x0006.    5. Entry: TTL is not equal to 0x0000.    6. Entry: Major Version is equal to 0x02. |
| Reference | DS\_SD\_0248 |
| Notes |  |

### CAN Message Transport

#### [SOMEIPCLT\_CAN\_01] Dedicated ports for UDP unicast

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to receive CAN Encapsulated SOME/IP messages, which are transmitted via UDP unicast, at a dedicated port. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFrameOnEvent** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**.   Source port is 0xCA00 and destination port is 0xCACC.   1. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANMessageInfo** with parameter **messageID** set to0x050 (**BaseFrameOnEvent**). 2. DUT: Sends corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0540 (**clientServiceGetCANMessageInfo)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **signalValue1** is equal to **UINT8\_VALUE**.       * **signalValue2** is equal to **UINT32\_VALUE**. |
| Reference | DS\_CN\_0010, DS\_CN\_0014 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_02] Dedicated ports for UDP multicast

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to receive CAN Encapsulated SOME/IP messages, which are transmitted via UDP multicast, at a dedicated port. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal. |
| Test Procedure | 1. TESTER: Sends multicast CAN Encapsulated SOME/IP message **BaseFrameOnEventMulitcast** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**.   Source port is 0xCA00 and destination port is 0xCACD.   1. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANMessageInfo** with parameter **messageID** set to0x051 (**BaseFrameOnEventMulitcast**). 2. DUT: Sends corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0540 (**clientServiceGetCANMessageInfo)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **signalValue1** is equal to **UINT8\_VALUE**.       * **signalValue2** is equal to **UINT32\_VALUE**. |
| Reference | DS\_CN\_0010, DS\_CN\_0014 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_03] Multiple messages per UDP datagram

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to receive multiple CAN Encapsulated SOME/IP messages, which are transmitted together in a single UDP datagram. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal. |
| Test Procedure | 1. TESTER: Sends unicast UDP datagram that contains the following three CAN Encapsulated SOME/IP messages:  * **BaseFrameOnEvent** * **BaseFramePeriodicOnEvent** * **ExtendedFrameOnEvent**   Payload of all messages is set as follow:   * + **signalUINT8** is set to **UINT8\_VALUE**.   + **signalUINT32** is set to **UINT32\_VALUE**.  1. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANMessageInfo** with parameter **messageID** set to 0x050 (**BaseFrameOnEvent**). 2. DUT: Sends corresponding SOME/IP Response message 3. TESTER: Verify received SOME/IP Response message. 4. Repeat steps 2 to 4 for message ID 0x54 (**BaseFramePeriodicOnEvent**) and 0x0000 0060 (**ExtendedFrameOnEvent**). |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0540 (**clientServiceGetCANMessageInfo)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **signalValue1** is equal to **UINT8\_VALUE**.       * **signalValue2** is equal to **UINT32\_VALUE**. 2. Apply verification step 4 for each message. |
| Reference | DS\_CN\_0012, DS\_CN\_0028, DS\_CN\_0029 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_04] Accept Client ID different than zero

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to accept CAN Encapsulated SOME/IP messages even when its received Client ID is different than zero. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **WRONG\_CLIENT\_ID**: Arbitrary client ID used for the test. Must not be zero.  **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFrameOnEvent** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**.   Client ID is set to **WRONG\_CLIENT\_ID**.   1. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANMessageInfo** with parameter **messageID** set to0x050 (**BaseFrameOnEvent**). 2. DUT: Sends corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0540 (**clientServiceGetCANMessageInfo)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **signalValue1** is equal to **UINT8\_VALUE**.       * **signalValue2** is equal to **UINT32\_VALUE**. |
| Reference | DS\_CN\_0003 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_05] Accept Session ID different than zero

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to accept CAN Encapsulated SOME/IP messages even when its received Session ID is different than zero. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **WRONG\_SESSION\_ID**: Arbitrary session ID used for the test. Must not be zero.  **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFrameOnEvent** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**.   Session ID is set to **WRONG\_SESSION\_ID**.   1. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANMessageInfo** with parameter **messageID** set to0x050 (**BaseFrameOnEvent**). 2. DUT: Sends corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0540 (**clientServiceGetCANMessageInfo)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **signalValue1** is equal to **UINT8\_VALUE**.       * **signalValue2** is equal to **UINT32\_VALUE**. |
| Reference | DS\_CN\_0004 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_06] Filter message by CAN Message ID

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to filter out CAN Encapsulated SOME/IP messages by its message ID, when they are configurated to not be processed by the DUT. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFrameOnEventFiltered** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**. 2. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANMessageInfo** with parameter **messageID** set to0x052 (**BaseFrameOnEventFiltered**). 3. DUT: Sends corresponding SOME/IP Response message. 4. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0540 (**clientServiceGetCANMessageInfo)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **signalValue1** is equal to 0x00 (default value).       * **signalUpdates1** is equal to 0x00.       * **signalValue2** is equal to 0x0000 0000 (default value).       * **signalUpdates2** is equal to 0x00. |
| Reference | DS\_CN\_0030 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_07] Filter signal by given destination

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to filter out particular signals from a CAN Encapsulated SOME/IP message, when they are configurated to not be processed by the DUT. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFrameOnEventFiltered2** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**. 2. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANMessageInfo** with parameter **messageID** set to0x053 (**BaseFrameOnEventFiltered2**). 3. DUT: Sends corresponding SOME/IP Response message. 4. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0540 (**clientServiceGetCANMessageInfo)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **signalValue1** is equal to **UINT8\_VALUE**.       * **signalUpdates1** is equal to 0x01.       * **signalValue2** is equal to 0x0000 0000 (default value).       * **signalUpdates2** is equal to 0x00. |
| Reference | DS\_CN\_0032 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_08] Duplicated Value Check discards signal value

|  |  |
| --- | --- |
| Item | Description |
| Purpose | In case of using the Duplicated Value Check, the DUT is expected to discard a received signal value when it is identical to the previous value of this signal. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE\_1**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE\_2**: Arbitrary signal value for the test. Must be different than UINT32\_VALUE\_1. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **ExtendedFrameOnEvent** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE\_1**. 2. TESTER: Sends unicast CAN Encapsulated SOME/IP message **ExtendedFrameOnEvent** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE\_2**. 3. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANMessageInfo** with parameter **messageID** set to0x060 (**ExtendedFrameOnEvent**). 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0540 (**clientServiceGetCANMessageInfo)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **signalValue1** is equal to **UINT8\_VALUE**.       * **signalUpdates1** is equal to 0x01.       * **signalValue2** is equal to **UINT32\_VALUE\_2**.       * **signalUpdates2** is equal to 0x02. |
| Reference | DS\_CN\_0033 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_09] Timeout of cyclic message is assumed as error

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to check the interval of cyclically sent CAN Encapsulated SOME/IP messages and assume a communication error when a message is not received in time. In detail, all signals of the timed-out message shall be updated to their default values. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **T\_TIMEOUT**: Threshold for message timeout detection in milliseconds [ms]. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFramePeriodic** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**. 2. TESTER: Waits until a duration of **T\_TIMEOUT** is elapsed. 3. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANMessageInfo** with parameter **messageID** set to0x055 (**BaseFramePeriodic**). 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0540 (**clientServiceGetCANMessageInfo)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **isTimeoutOccurred** is true.       * **signalValue1** is equal to 0x00 (default value).       * **signalUpdates1** is equal to 0x02.       * **signalValue2** is equal to 0x0000 0000 (default value).       * **signalUpdates2** is equal to 0x02. |
| Reference | DS\_CN\_0037, DS\_CN\_0038, DS\_CN\_0039 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_10] Reset timeout timer for cyclic message

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to reset the timer for timeout detection of cyclic sent CAN Encapsulated SOME/IP messages each time, a new message has been received. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **N\_CYCLES** (UINT8): Number of message cycles used for the test. Must be at least 3. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFramePeriodic** for a number of **N\_CYCLES** considering the defined cycle time. Payload is set as follow:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**. 2. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANMessageInfo** with parameter **messageID** set to0x055 (**BaseFramePeriodic**). 3. DUT: Sends corresponding SOME/IP Response message. 4. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0540 (**clientServiceGetCANMessageInfo)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **isTimeoutOccurred** is false.       * **signalValue1** is equal to **UINT8\_VALUE**.       * **signalUpdates1** is equal to **N\_CYCLES**.       * **signalValue2** is equal to **UINT32\_VALUE**.       * **signalUpdates2** is equal to **N\_CYCLES**. |
| Reference | DS\_CN\_0031 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_11] No timeout while inactive network mode

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to not perform timeout detection for any CAN Encapsulated SOME/IP message while it is in inactive network mode. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **T\_TIMEOUT**: Threshold for message timeout detection in milliseconds [ms]. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFramePeriodic** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**. 2. TESTER: Sends SOME/IP Request-No-Return message for method **changeNetworkMode** with the following parameters:    * **mode** is set to0x01 (inactive).    * **duration** is set to **T\_TIMEOUT** x 2. 3. TESTER: Waits until a duration of **T\_TIMEOUT** x 2 is elapsed. 4. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANMessageInfo** with parameter **messageID** set to0x055 (**BaseFramePeriodic**). 5. DUT: Sends corresponding SOME/IP Response message. 6. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0540 (**clientServiceGetCANMessageInfo)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **isTimeoutOccurred** is false.       * **signalValue1** is equal to **UINT8\_VALUE**.       * **signalUpdates1** is equal to 0x01.       * **signalValue2** is equal to **UINT32\_VALUE**.       * **signalUpdates2** is equal to 0x01. |
| Reference | DS\_CN\_0040 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_12] No timeout while start offset delay

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to not perform timeout detection for any CAN Encapsulated SOME/IP message during its corresponding initial start offset delay. Therefore, the DUT has to check status of the sending ECU in order to determine whether start offset delay shall be considered or not. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_TIMEOUT**: Threshold for message timeout detection in milliseconds [ms]. |
| Test Procedure | 1. TESTER: Waits until a duration of **T\_TIMEOUT** x 2 is elapsed. 2. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANMessageInfo** with parameter **messageID** set to0x055 (**BaseFramePeriodic**). 3. DUT: Sends corresponding SOME/IP Response message. 4. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0540 (**clientServiceGetCANMessageInfo)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **isTimeoutOccurred** is false.       * **signalValue1** is equal to 0x00 (default value).       * **signalUpdates1** is equal to 0x00.       * **signalValue2** is equal to 0x0000 0000 (default value).       * **signalUpdates2** is equal to 0x00. |
| Reference | DS\_CN\_0040 |
| Notes | Checking status of sending ECU shall happen via appropriate Network Management (NM) messages. |

#### [SOMEIPCLT\_CAN\_13] No timeout while already reported

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to not perform further timeout detection for a CAN Encapsulated SOME/IP message while timeout of that message is already detected. When later at some point a respective message could be received, timeout detection shall be resumed again. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **T\_TIMEOUT**: Threshold for message timeout detection in milliseconds [ms]. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFramePeriodic** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**. 2. TESTER: Waits until a duration of **T\_TIMEOUT** is elapsed. 3. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANMessageInfo** with parameter **messageID** set to0x055 (**BaseFramePeriodic**). 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. 6. TESTER: Waits until a duration of **T\_TIMEOUT** is elapsed. 7. Repeat previous steps 3 to 5. 8. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFramePeriodic** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**. 9. TESTER: Waits until a duration of **T\_TIMEOUT** is elapsed. 10. Repeat previous steps 3 to 5. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0540 (**clientServiceGetCANMessageInfo)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **isTimeoutOccurred** is true.       * **signalUpdates1** is equal to 0x02.       * **signalUpdates2** is equal to 0x02. 2. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0540 (**clientServiceGetCANMessageInfo)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **isTimeoutOccurred** is false.       * **signalUpdates1** is equal to 0x02.       * **signalUpdates2** is equal to 0x02. 3. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0540 (**clientServiceGetCANMessageInfo)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **isTimeoutOccurred** is true.       * **signalUpdates1** is equal to 0x03.       * **signalUpdates2** is equal to 0x03. |
| Reference | DS\_CN\_0040 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_14] No timeout for “On Event” message

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to not perform timeout detection for any CAN Encapsulated SOME/IP message which is configured as “On Event” message. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **T\_TIMEOUT**: Threshold for message timeout detection in milliseconds [ms]. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFrameOnEvent** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**. 2. TESTER: Waits until a duration of **T\_TIMEOUT** is elapsed. 3. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANMessageInfo** with parameter **messageID** set to0x050 (**BaseFrameOnEvent**). 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0540 (**clientServiceGetCANMessageInfo)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **isTimeoutOccurred** is false.       * **signalValue1** is equal to **UINT8\_VALUE**.       * **signalUpdates1** is equal to 0x01.       * **signalValue2** is equal to **UINT32\_VALUE**.       * **signalUpdates2** is equal to 0x01. |
| Reference | DS\_CN\_0040 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_15] Increment error counters

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to increment error counters by one, when timeout of a CAN Encapsulated SOME/IP message has been detected.  In detail, error counter Global\_ERR\_COUNT shall be updated for any message and the individual error counter ERR\_COUNT only for messages of a particular CAN message ID. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test.  **UINT32\_VALUE**: Arbitrary signal value for the test.  **N\_CYCLES** (UINT8): Number of message cycles used for the test. Must be at least 3. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFramePeriodic** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**. 2. TESTER: Sends unicast CAN Encapsulated SOME/IP message **ExtendedFramePeriodic** for a number of **N\_CYCLES** considering the defined cycle time. Payload is set as follow:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**. 3. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANErrorCount** with parameter **messageID** set to0x055 (**BaseFramePeriodic**). 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. 6. Repeat previous steps 3 to 5 for message ID 0x0000 0061 (**ExtendedFramePeriodic**). |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0541 (**clientServiceGetCANErrorCount)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **globalErrorCount** is equal to 0x01.       * **errorCount** is equal to 0x01. 2. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0541 (**clientServiceGetCANErrorCount)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **globalErrorCount** is equal to 0x01.       * **errorCount** is equal to 0x00. |
| Reference | DS\_CN\_0043, DS\_CN\_0045 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_16] Error counter values not exceed 255

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to ensure that error counters do not exceed a maximum value of 255, even when more errors have been detected. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test.  **UINT32\_VALUE**: Arbitrary signal value for the test.  **T\_TIMEOUT**: Threshold for message timeout detection in milliseconds [ms]. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFramePeriodic** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**. 2. TESTER: Waits until a duration of **T\_TIMEOUT** is elapsed. 3. Repeat previous steps 1 to 2 for 256 times. 4. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANErrorCount** with parameter **messageID** set to0x055 (**BaseFramePeriodic**). 5. DUT: Sends corresponding SOME/IP Response message. 6. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0541 (**clientServiceGetCANErrorCount)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **globalErrorCount** is equal to 0xFF.       * **errorCount** is equal to 0xFF. |
| Reference | DS\_CN\_0046 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_17] Reset error counter if network mode changes to active

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to reset error counters to zero when its network mode has changed to active. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test.  **UINT32\_VALUE**: Arbitrary signal value for the test.  **T\_TIMEOUT**: Threshold for message timeout detection in milliseconds [ms].  **T\_NETWORK\_INACTIVE**: Duration in milliseconds [ms] how long the DUT shall stay in network mode inactive for the test. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFramePeriodic** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**. 2. TESTER: Waits until a duration of **T\_TIMEOUT** is elapsed. 3. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANErrorCount** with parameter **messageID** set to0x055 (**BaseFramePeriodic**). 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. 6. TESTER: Sends SOME/IP Request-No-Return message for method **changeNetworkMode** with the following parameters:    * **mode** is set to0x01 (inactive).    * **duration** is set to **T\_NETWORK\_INACTIVE**. 7. TESTER: Waits until duration of **T\_NETWORK\_INACTIVE** is elapsed. 8. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANErrorCount** with parameter **messageID** set to0x055 (**BaseFramePeriodic**). 9. DUT: Sends corresponding SOME/IP Response message. 10. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0541 (**clientServiceGetCANErrorCount)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **globalErrorCount** is equal to 0x01.       * **errorCount** is equal to 0x01. 2. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0541 (**clientServiceGetCANErrorCount)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **globalErrorCount** is equal to 0x00.       * **errorCount** is equal to 0x00. |
| Reference | DS\_CN\_0044 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_18] Reset error counter while network mode stays in error state

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to reset error counters to zero while its network mode stays in error state. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test.  **UINT32\_VALUE**: Arbitrary signal value for the test.  **T\_TIMEOUT**: Threshold for message timeout detection in milliseconds [ms].  **T\_NETWORK\_ERROR**: Duration in milliseconds [ms] how long the DUT shall stay in network mode error state for the test. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFramePeriodic** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**. 2. TESTER: Waits until a duration of **T\_TIMEOUT** is elapsed. 3. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANErrorCount** with parameter **messageID** set to0x055 (**BaseFramePeriodic**). 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. 6. TESTER: Sends SOME/IP Request-No-Return message for method **changeNetworkMode** with the following parameters:    * **mode** is set to0x02 (error state).    * **duration** is set to **T\_NETWORK\_ERROR**. 7. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANErrorCount** with parameter **messageID** set to0x055 (**BaseFramePeriodic**). 8. DUT: Sends corresponding SOME/IP Response message. 9. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0541 (**clientServiceGetCANErrorCount)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **globalErrorCount** is equal to 0x01.       * **errorCount** is equal to 0x01. 2. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0541 (**clientServiceGetCANErrorCount)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **globalErrorCount** is equal to 0x00.       * **errorCount** is equal to 0x00. |
| Reference | DS\_CN\_0044 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_19] No error counting while sending ECU is missing

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to not increment error counters when messages timed out that should have been sent by an ECU which is currently not available to the network. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **T\_TIMEOUT**: Threshold for timeout detection in milliseconds [ms]. |
| Test Procedure | 1. TESTER: Waits until the respective start offset delay of message **BaseFramePeriodic** is elapsed. 2. TESTER: Waits until a duration of **T\_TIMEOUT** is elapsed. 3. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANErrorCount** with parameter **messageID** set to0x055 (**BaseFramePeriodic**). 4. DUT: Sends corresponding SOME/IP Response message. 5. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0541 (**clientServiceGetCANErrorCount)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **globalErrorCount** is equal to 0x00.       * **errorCount** is equal to 0x00. |
| Reference | DS\_CN\_0042 |
| Notes | Checking whether an ECU is available or not, happens via appropriate Network Management (NM) messages. |

#### [SOMEIPCLT\_CAN\_20] E2E: Decode Profile 5 message

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to receive and decode CAN Encapsulated SOME/IP messages accordingly to the AUTOSAR 4.2.2 E2E Profile 5. |
| Preconditions | DUT is running and offering the Enhanced Testability Service  DUT supports E2E Protection.  CAN E2E Profile 5 instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFramePeriodicE2EProfile5** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**.   E2E Alive Counter is set to 0x00.  E2E Data ID is set to 0xF85A.  E2E Data Length is set to 0x05.   1. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANMessageInfo** with parameter **messageID** set to0x05A (**BaseFramePeriodicE2EProfile5**). 2. DUT: Sends corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0540 (**clientServiceGetCANMessageInfo)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **signalValue1** is equal to **UINT8\_VALUE**.       * **signalValue2** is equal to **UINT32\_VALUE**. |
| Reference | DS\_CN\_0047 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_21] E2E: Decode Profile 11 message

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to receive and decode CAN Encapsulated SOME/IP messages accordingly to the AUTOSAR 4.3.1 E2E Profile 11. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT supports E2E Protection.  CAN E2E Profile 11 instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFramePeriodicE2EProfile11** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**.   E2E Alive Counter is set to 0x0.  E2E Data ID is set to 0xF85B.  E2E Data Length is set to 0x05.   1. TESTER: Sends SOME/IP Request message for method **clientServiceGetCANMessageInfo** with parameter **messageID** set to0x05B (**BaseFramePeriodicE2EProfile11**). 2. DUT: Sends corresponding SOME/IP Response message. 3. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0540 (**clientServiceGetCANMessageInfo)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameters:       * **signalValue1** is equal to **UINT8\_VALUE**.       * **signalValue2** is equal to **UINT32\_VALUE**. |
| Reference | DS\_CN\_0047 |
| Notes |  |

#### [SOMEIPCLT\_CAN\_22] E2E: Update Alive Counter even on repeated value error

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to update its internal Alive Counter to the value which is received by appropriated CAN Encapsulated SOME/IP messages, even when both values are identical that is treated as repeated value error. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT supports E2E Protection.  CAN E2E Profile 11 instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFramePeriodicE2EProfile11** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**.   E2E Alive Counter is set to 0x0.  E2E Data ID is set to 0xF85B.  E2E Data Length is set to 0x05.   1. Repeat step 1. 2. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError**. 3. DUT: Sends corresponding SOME/IP Response message. 4. TESTER: Verify received SOME/IP Response message. 5. Repeat step 1. 6. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError**. 7. DUT: Sends corresponding SOME/IP Response message. 8. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameter:       * **errorCode** is equal to 0x0B (E\_E2E\_REPEATED). 2. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameter:       * **errorCode** is equal to 0x0B (E\_E2E\_REPEATED). |
| Reference | DS\_CN\_0050, DS\_CN\_0051 |
| Notes | Current value of the DUT’s Alive Counter is checked indirectly by trying to force an E\_E2E\_REPEATED error. Only when this error is reported by the DUT, the exact value of the Alive Counter can be confirmed. It must be equal to the one which has forced the error. |

#### [SOMEIPCLT\_CAN\_23] E2E: Update Alive Counter even on wrong sequence error

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to update its internal Alive Counter to the value which is received by appropriated CAN Encapsulated SOME/IP messages, even when the delta between both values is too big that is treated as wrong sequence error. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT supports E2E Protection.  CAN E2E Profile 11 instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFramePeriodicE2EProfile11** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**.   E2E Alive Counter is set to 0x0.  E2E Data ID is set to 0xF85B.  E2E Data Length is set to 0x05.   1. Repeat step 1 but with E2E Alive Counter set to 0xB. 2. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError**. 3. DUT: Sends corresponding SOME/IP Response message. 4. TESTER: Verify received SOME/IP Response message. 5. Repeat step 2. 6. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError**. 7. DUT: Sends corresponding SOME/IP Response message. 8. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameter:       * **errorCode** is equal to 0x0C (E\_E2E\_WRONG\_SEQUENCE). 2. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameter:       * **errorCode** is equal to 0x0B (E\_E2E\_REPEATED). |
| Reference | DS\_CN\_0050, DS\_CN\_0051 |
| Notes | Current value of the DUT’s Alive Counter is checked indirectly by trying to force an E\_E2E\_REPEATED error. Only when this error is reported by the DUT, the exact value of the Alive Counter can be confirmed. It must be equal to the one which has forced the error. |

#### [SOMEIPCLT\_CAN\_24] E2E: No Alive Counter update on CRC error

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to discard the new Alive Counter value of a received CAN Encapsulated SOME/IP message when a CRC error has been detected for this message. Current value of the DUT’s Alive Counter shall not be updated. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT supports E2E Protection.  CAN E2E Profile 11 instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **WRONG\_CRC8**: Arbitrary 8-bit E2E CRC value for the test. Must be different than the valid one. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFramePeriodicE2EProfile11** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**.   E2E Alive Counter is set to 0x0.  E2E Data ID is set to 0xF85B.  E2E Data Length is set to 0x05.   1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFramePeriodicE2EProfile11** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**.   E2E Alive Counter is set to 0x1.  E2E Data ID is set to 0xF85B.  E2E Data Length is set to 0x05.  E2E CRC is set to **WRONG\_CRC8**.   1. Repeat step 1. 2. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError**. 3. DUT: Sends corresponding SOME/IP Response message. 4. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameter:       * **errorCode** is equal to 0x0B (E\_E2E\_REPEATED). |
| Reference | DS\_CN\_0050, DS\_CN\_0051 |
| Notes | Current value of the DUT’s Alive Counter is checked indirectly by trying to force an E\_E2E\_REPEATED error. Only when this error is reported by the DUT, the exact value of the Alive Counter can be confirmed. It must be equal to the one which has forced the error. |

#### [SOMEIPCLT\_CAN\_25] E2E: Report Alive Counter errors only for “Periodic” messages

|  |  |
| --- | --- |
| Item | Description |
| Purpose | The DUT is expected to report Alive Counter related errors of received CAN Encapsulated SOME/IP messages only when that message is configured as “Periodic” message. In detail, the test simulates a wrong sequence error for an event triggered message and checks the DUT’s reaction. |
| Preconditions | DUT is running and offering the Enhanced Testability Service.  DUT supports E2E Protection.  CAN default instance must be activated as ETS client instance. |
| Test Setup | Whitebox Test (ETS) (see chapter 4.1.2) |
| Test variables | **UINT8\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal.  **UINT32\_VALUE**: Arbitrary signal value for the test. Must be different than default value of the respective signal. |
| Test Procedure | 1. TESTER: Sends unicast CAN Encapsulated SOME/IP message **BaseFrameOnEventE2EProfile11** with the following payload:    * **signalUINT8** is set to **UINT8\_VALUE**.    * **signalUINT32** is set to **UINT32\_VALUE**.   E2E Alive Counter is set to 0x0.  E2E Data ID is set to 0xF85C.  E2E Data Length is set to 0x05.   1. Repeat step 1 but with E2E Alive Counter set to 0xB. 2. TESTER: Sends SOME/IP Request message for method **clientServiceGetLastError**. 3. DUT: Sends corresponding SOME/IP Response message. 4. TESTER: Verify received SOME/IP Response message. |
| Verification | 1. SOME/IP Response is received before **TimeoutResponse** and contains:    1. Service ID is equal to **ETS\_ServiceID**.    2. Method ID is equal to 0x0521 (**clientServiceGetLastError)**.    3. Return Code is equal to 0x00 (E\_OK).    4. Payload provides the following parameter:       * **errorCode** is equal to 0x00 (E\_OK). |
| Reference | DS\_CN\_0051 |
| Notes |  |

# 참조

## HMC/KMC 간행물

표 10 HMC/KMC 간행물

|  |  |  |
| --- | --- | --- |
| **No.** | **Reference** | **Version** |
| [1] | ES96595-05 Ethernet IP Communication Design Specification |  |
| [2] | ES96595-11 Ethernet SOME/IP Communication Design Specification |  |

## 표준 간행물

표 11 표준 간행물

|  |  |  |
| --- | --- | --- |
| **No.** | **Reference** | **Version** |
| [1] | OPEN Alliance Automotive Ethernet ECU Test Specification | 2.0/  Aug 23, 2017 |