

| | |
|-----------------------------------|-------------------------------------------------------------------------|
| Document Title | SWS_ServiceDiscovery: Complete Change Documentation 4.3.0 - 4.3.1 |
| Document Owner | AUTOSAR |
| Document Responsibility | AUTOSAR |
| Document Identification No | 695 |

| | |
|---------------------------------|------------------|
| Document Status | Final |
| Part of AUTOSAR Standard | Classic Platform |
| Part of Standard Release | 4.3.1 |

Table of Contents

| | | |
|------|----------------------------------|----|
| 1 | SWS_ServiceDiscovery | 3 |
| 1.1 | Specification Item ECUC_SD_00096 | 3 |
| 1.2 | Specification Item ECUC_SD_00106 | 5 |
| 1.3 | Specification Item ECUC_SD_00118 | 7 |
| 1.4 | Specification Item SWS_SD_00107 | 9 |
| 1.5 | Specification Item SWS_SD_00132 | 10 |
| 1.6 | Specification Item SWS_SD_00293 | 12 |
| 1.7 | Specification Item SWS_SD_00343 | 13 |
| 1.8 | Specification Item SWS_SD_00362 | 15 |
| 1.9 | Specification Item SWS_SD_00369 | 16 |
| 1.10 | Specification Item SWS_SD_00407 | 18 |
| 1.11 | Specification Item SWS_SD_00410 | 19 |
| 1.12 | Specification Item SWS_SD_00446 | 21 |
| 1.13 | Specification Item SWS_SD_00453 | 21 |
| 1.14 | Specification Item SWS_SD_00454 | 24 |
| 1.15 | Specification Item SWS_SD_00455 | 26 |
| 1.16 | Specification Item SWS_SD_00469 | 28 |
| 1.17 | Specification Item SWS_SD_00471 | 29 |
| 1.18 | Specification Item SWS_SD_00473 | 31 |
| 1.19 | Specification Item SWS_SD_00481 | 32 |
| 1.20 | Specification Item SWS_SD_00723 | 34 |
| 1.21 | Specification Item SWS_SD_00724 | 35 |
| 1.22 | Specification Item SWS_SD_00725 | 37 |

1 SWS_ServiceDiscovery

1.1 Specification Item ECUC_SD_00096

Trace References:

none

Content:

| | | | |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|---------------------|
| Name | SdEventActivationRefSdEventHandlerUdp.SdEventActivationRef | | |
| Parent Container | SdEventHandlerUdp | | |
| Description | Reference to a SoAdRoutingGroup for activation of the data path for a subscribed client (start sending events after subscribe). This is usually equal to the SdEventActivationRef referenced by SdEventHandlerUdp | | |
| Multiplicity | 0..1 | | |
| Type | Symbolic name reference to [SoAdRoutingGroup] | | |
| Post-Build Variant Multiplicity | true | | |
| Post-Build Variant Value | true | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #68823: [Sd] Switching of EventHandler from Unicast to Multicast.

Problem description:

If the SdEventHandlerMulticastThreshold is set to say 3, and at the current moment two Consumed eventgroups (say A and B) are subscribed to an EventHandlerThese are being served by Unicast as per [SWS_SD_00455], the SubscribeEventGroupAck having previously being sent to these consumed eventgroups.

Now if a new SubscribeEventGroup (C) is received, the Event Handler will shift these three to multicast as per [SWS_SD_00455].

Will it also send a new SubscribeEventGroupAck to the Consumed eventgroups A and B to indicate that the events are now sent over multicast?

If so, what are the actions on the client side for the consumed eventgroups A and B?

Agreed solution:

~SWS_SD_00455

"SoAd_DisableSpecific Routing" -> "SoAd_DisableSpecificRouting"

~Figure 9.6

"SoAd_GetRemoteAddr(return, SoConId, IpAddrPtr)" ->

"SoAd_GetRemoteAddr(return, SdMulticastEventSoConRef, IpAddrPtr)"

~Figure 9.8

"SoAd_EnableRouting(SdEventHandlerTcp->SdEventActivationRef)" ->

"SoAd_EnableSpecificRouting(SdEventHandlerTcp->SdEventActivationRef, So-
ConId)"

~Alt [MulticastThreshold==0 || numOfSubs<MulticastThreshold]

"SoAd_EnableRouting(SdEventHandlerUdp->SdEventActivationRef)" ->

"SoAd_EnableSpecificRouting(SdEventHandlerUdp->SdEventActivationRef, So-
ConId)"

~Alt [numOfSubs=MulticastThreshold]

"[numOfSubs=MulticastThreshold]" -> "[numOfSubs==MulticastThreshold]"

"SoAd_EnableRouting(SdEventHandlerMulticast->SdEventActivationRef)" ->

"SoAd_EnableSpecificRouting(SdEventHandlerMulticast->SdEventActivationRef,
SdEventHandlerMulticast->SdMulticastEventSoConRef)"

"SoAd_DisableRouting(SdEventHandlerUdp->SdEventActivationRef)" ->

"SoAd_DisableSpecificRouting(SdEventHandlerUdp->SdEventActivationRef,
SoConId)"

-Alt [numOfSubs>=MulticastThreshold]

~ECUC_SD_00096

Add at end of description: "This is usually equal to the SdEventActivationRef
referenced by SdEventHandlerUdp".

~ECUC_SD_00106

Add at end of description: "This is usually equal to the SdConsumedEventGroupUd-
pActivationRef".

~ECUC_SD_00118

Change multiplicity of SdMulticastEventSoConRef from optional to mandatory (0..1
to 1).

-Last change on issue 68823 comment 44-

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 3 | 1 |

1.2 Specification Item ECUC_SD_00106

Trace References:

none

Content:

| | | | |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|---------------------|
| Name | SdConsumedEventGroupMulticastActivationRefSdConsumedEventGroup.SdConsumedEventGroupMulticastActivationRef | | |
| Parent Container | SdConsumedEventGroup | | |
| Description | <p>The reference of a Routing Group in order to activate and setup the Socket Connection for Multicast Events of this EventGroup. The multicast address from the received Multicast option is setup by SoAd_RequestIpAddrAssignment.</p> <p>The local address is the same as for the unicast events; thus, it was sent in the UDP Endpoint option of the Subscribe EventGroup entry.</p> <p><i>This is usually equal to the SdConsumedEventGroupUdpActivationRef.</i></p> | | |
| Multiplicity | 0..1 | | |
| Type | Symbolic name reference to [SoAdRoutingGroup] | | |
| Post-Build Variant Multiplicity | true | | |
| Post-Build Variant Value | true | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #68823: [Sd] Switching of EventHandler from Unicast to Multicast.

Problem description:

If the SdEventHandlerMulticastThreshold is set to say 3, and at the current moment two Consumed eventgroups (say A and B) are subscribed to an EventHandlerThese are being served by Unicast as per [SWS_SD_00455], the SubscribeEventGroupAck having previously being sent to these consumed eventgroups.

Now if a new SubscribeEventGroup (C) is received, the Event Handler will shift these three to multicast as per [SWS_SD_00455].

Will it also send a new SubscribeEventGroupAck to the Consumed eventgroups A and B to indicate that the events are now sent over multicast?

If so, what are the actions on the client side for the consumed eventgroups A and B?

Agreed solution:

~SWS_SD_00455

"SoAd_DisableSpecific Routing" -> "SoAd_DisableSpecificRouting"

~Figure 9.6

"SoAd_GetRemoteAddr(return, SoConId, IpAddrPtr)" ->
"SoAd_GetRemoteAddr(return, SdMulticastEventSoConRef, IpAddrPtr)"

~Figure 9.8

"SoAd_EnableRouting(SdEventHandlerTcp->SdEventActivationRef)" ->
"SoAd_EnableSpecificRouting(SdEventHandlerTcp->SdEventActivationRef, So-
ConId)"

~Alt [MulticastThreshold==0 || numOfSubs<MulticastThreshold]

"SoAd_EnableRouting(SdEventHandlerUdp->SdEventActivationRef)" ->
"SoAd_EnableSpecificRouting(SdEventHandlerUdp->SdEventActivationRef, So-
ConId)"

~Alt [numOfSubs=MulticastThreshold]

"[numOfSubs=MulticastThreshold]" -> "[numOfSubs==MulticastThreshold]"
"SoAd_EnableRouting(SdEventHandlerMulticast->SdEventActivationRef)" ->
"SoAd_EnableSpecificRouting(SdEventHandlerMulticast->SdEventActivationRef,
SdEventHandlerMulticast->SdMulticastEventSoConRef)"
"SoAd_DisableRouting(SdEventHandlerUdp->SdEventActivationRef)" ->
"SoAd_DisableSpecificRouting(SdEventHandlerUdp->SdEventActivationRef,
SoConId)"

-Alt [numOfSubs>=MulticastThreshold]

~ECUC_SD_00096

Add at end of description: "This is usually equal to the SdEventActivationRef referenced by SdEventHandlerUdp".

~ECUC_SD_00106

Add at end of description: "This is usually equal to the SdConsumedEventGroupUd-

pActivationRef".

~ECUC_SD_00118

Change multiplicity of SdMulticastEventSoConRef from optional to mandatory (0..1 to 1).

–Last change on issue 68823 comment 44–

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 3 | 1 |

1.3 Specification Item ECUC_SD_00118

Trace References:

none

Content:

| | | | |
|----------------------------------|-----------------------------------------------------------------------------------|---|---------------------|
| Name | SdMulticastEventSoConRefSdEventHandlerMulticast.SdMulticastEventSoConRef | | |
| Parent Container | SdEventHandlerMulticast | | |
| Description | Reference to the SoAdSocketConnection representing the multicast data path (UDP). | | |
| Multiplicity | 0..1 1 | | |
| Type | Symbolic name reference to [SoAdSocketConnection] | | |
| Post-Build Variant Multiplicity | true | | |
| Post-Build Variant Value | true | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #68823: [Sd] Switching of EventHandler from Unicast to Multicast.

Problem description:

If the SdEventHandlerMulticastThreshold is set to say 3, and at the current moment two Consumed eventgroups (say A and B) are subscribed to an EventHandlerThese are being served by Unicast as per [SWS_SD_00455], the Sub-

scribeEventGroupAck having previously being sent to these consumed eventgroups.

Now if a new SubscribeEventGroup (C) is received, the Event Handler will shift these three to multicast as per [SWS_SD_00455].

Will it also send a new SubscribeEventGroupAck to the Consumed eventgroups A and B to indicate that the events are now sent over multicast?

If so, what are the actions on the client side for the consumed eventgroups A and B?

Agreed solution:

~SWS_SD_00455

"SoAd_DisableSpecific Routing" -> "SoAd_DisableSpecificRouting"

~Figure 9.6

"SoAd_GetRemoteAddr(return, SoConId, IpAddrPtr)" ->
"SoAd_GetRemoteAddr(return, SdMulticastEventSoConRef, IpAddrPtr)"

~Figure 9.8

"SoAd_EnableRouting(SdEventHandlerTcp->SdEventActivationRef)" ->
"SoAd_EnableSpecificRouting(SdEventHandlerTcp->SdEventActivationRef, So-
ConId)"

~Alt [MulticastThreshhold==0 || numOfSubs<MulticastThreshhold]

"SoAd_EnableRouting(SdEventHandlerUdp->SdEventActivationRef)" ->
"SoAd_EnableSpecificRouting(SdEventHandlerUdp->SdEventActivationRef, So-
ConId)"

~Alt [numOfSubs=MulticastThreshhold]

"[numOfSubs=MulticastThreshhold]" -> "[numOfSubs==MulticastThreshhold]"
"SoAd_EnableRouting(SdEventHandlerMulticast->SdEventActivationRef)" ->
"SoAd_EnableSpecificRouting(SdEventHandlerMulticast->SdEventActivationRef,
SdEventHandlerMulticast->SdMulticastEventSoConRef)"
"SoAd_DisableRouting(SdEventHandlerUdp->SdEventActivationRef)" ->
"SoAd_DisableSpecificRouting(SdEventHandlerUdp->SdEventActivationRef,
SoConId)"

-Alt [numOfSubs>=MulticastThreshhold]

~ECUC_SD_00096

Add at end of description: "This is usually equal to the SdEventActivationRef referenced by SdEventHandlerUdp".

~ECUC_SD_00106

Add at end of description: "This is usually equal to the SdConsumedEventGroupUdpActivationRef".

~ECUC_SD_00118

Change multiplicity of SdMulticastEventSoConRef from optional to mandatory (0..1 to 1).

–Last change on issue 68823 comment 44–

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 3 | 1 |

1.4 Specification Item SWS_SD_00107

Trace References:

none

Content:

Development error values are of type uint8.

| Type or error | Relevance | Related error code | Value [hex] |
|---------------------------------------------|-------------|--------------------------------|-------------|
| SD has not been initialized | Development | SD_E_NOT_INITIALIZED UNINIT | 0x01 |
| Null pointer has been passed as an argument | Development | SD_E_PARAM_POINTER | 0x02 |
| Invalid mode request | Development | SD_E_INV_MODE | 0x03 |
| Invalid Id | Development | SD_E_INV_ID | 0x04 |
| Initialization failed | Development | SD_E_INIT_FAILED | 0x05 |

Table [REF] - Error classification (Development Errors)

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors

–Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***

SWS_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:

SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:

Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API
UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

—Last change on issue 59085 comment 88—

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 4 | 1 |

1.5 Specification Item SWS_SD_00132

Trace References:

none

Content:

If development error detection is enabled and the Service Discovery module has not been initialized using Sd_Init(), the Sd_MainFunction function shall raise the development error code SD_E_NOT_INITIALIZED UNINIT and the Sd_MainFunction function shall return without further action.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors
–Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***

SWS_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:

SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:

Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API
UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 4 | 1 |

1.6 Specification Item SWS_SD_00293

Trace References:

none

Content:

Services Non-SOME/IP-Services exist, that are not identified by a unique 16 Bit Service ID but a unique value of the key otherserv. These services use the Service ID 0xFFFFE and must always carry a configuration option with an otherserv record. ECUs receiving an entry with Service ID 0xFFFFE shall use the configuration option and the otherserv record within in order to identify the relevant Service or Eventgroup configuration item. **This means that two Service Instance with the same Service ID and Service Instance ID may exist as long as their otherserv record is different.**

The configuration option shall be built based on configuration parameters mentioned in SWS_SD_00292.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77251: [SD] Contradicting requirements on the usage of the otherserve configuration option

Problem description:

According to the Autosar Classic Platform the otherserv option shall be used to identify services with not unique ServiceIDs. (see SWS_SD_00293).

According to AUTOSAR_PRS_SOMEIPServiceDiscoveryProtocol the otherserv option shall be used exclusively for non-SOME/IP services.(see PRS_SOMEIPSD_00437)

–Last change on issue 77251 comment 1–

Agreed solution:

Modify SWS_SD_00293

from:

Services exist, that are not identified by a unique 16 Bit Service ID but a unique value of the key otherserv. These services use the Service ID 0xFFFE and must always carry a configuration option with an otherserv record. ECUs receiving an entry with Service ID 0xFFFE shall use the configuration option and the otherserv record within in order to identify the relevant Service or Eventgroup configuration item. This means that two Service Instance with the same Service ID and Service Instance ID may exist as long as their otherserv record is different. The configuration option shall be built based on configuration parameters mentioned in SWS_SD_00292.

to:

Non-SOME/IP-Services exist, that are not identified by a unique 16 Bit Service ID but a unique value of the key otherserv. These services use the Service ID 0xFFFE and must always carry a configuration option with an otherserv record. ECUs receiving an entry with Service ID 0xFFFE shall use the configuration option and the otherserv record within in order to identify the relevant Service or Eventgroup configuration item.

Modify PRS_SOMEIPSD_00437

Replace:

"contain at least an entry with key \"otherserv\""

by

"contain exactly one entry with key \"otherserv\""

–Last change on issue 77251 comment 10–

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 4 | 1 |

1.7 Specification Item SWS_SD_00343

Trace References:

none

Content:

If the Service Discovery Module receives a FindService Entry the following step (s) shall be performed **in the following order**:

- Send an "OfferService Entry" considering the appropriate delay (see chapter [REF]).

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73332: [SD] Improvement of SWS_ServiceDiscovery

Problem description:

During reading the SWS_ServiceDiscovery I discovered many minor issues and documented them in the attached files.

Agreed solution:

-[1]- [SWS_SD_00651] and [SWS_SD_00481] are redundant regarding the required goal:

[SWS_SD_00481]

The Service Discovery module shall minimize the amount of messages sent by combining multiple entries within one message whenever it is possible and not in conflict to the configuration. ()

[SWS_SD_00651]

The amount of separate Service Discovery messages shall be reduced, i.e.: Combine as much information as possible into one Service Discovery message before calling the Socket Adaptors transmit API. This means that when a entry is sent after waiting the appropriate delay (i.e. based on Request-Response-Delay) all other entries for this communication partner may be packed into the Service Discovery message as well.()

remove -SWS_SD_00481

-[2]- Replace the note below requirement SWS_SD_00494 by two explicit new requirements:

+SWS_SD_00xxx_1

If SdServerTimerRequestResponseMinDelay and SdServerTimerRequestResponseMaxDelay are set to the same value, this is the value of the delay.

If SdServerTimerRequestResponseMinDelay and SdServerTimerRequestResponseMaxDelay are set to 0, no delay shall be introduced.

+SWS_SD_00xxx_2

If SdClientTimerRequestResponseMinDelay and SdClientTimerRequestResponseMaxDelay are set to the same value, this is the value of the delay.

If SdClientTimerRequestResponseMinDelay and SdClientTimerRequestResponseMaxDelay are set to 0, no delay shall be introduced.

-[3]- Editorial [SWS_SD_00343]

If the Service Discovery Module receives a FindService Entry the following step(s) shall be performed in the following order:

Send an OfferService Entry considering the appropriate delay (see chapter 7.5.3).

()

=> Remove the string "in the following order", i.e.: If the Service Discovery Module receives a FindService Entry the following step shall be performed: [...]

-[4]- Editorial [SWS_SD_00453] and [SWS_SD_00454]:

"Only if this client was not subscribe before" => "Only if the client was not subscribed before" [...]

-[5]- Editorial [SWS_SD_00369]:

"After sending the maximum repetitions" => "After sending the maximum amount of repetitions" [...]

–Last change on issue 73332 comment 11–

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 1 | 1 |

1.8 Specification Item SWS_SD_00362

Trace References:

none

Content:

When the Initial Wait Phase is entered, the API SoAd_EnableSpecificRouting() shall be called with SdClientServiceActivationRef (see SdConsumedMethods) and the relevant Socket Connections for this Client Service Instance.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76523: [SD] SWS_SD_00362 has not been removed as described in RfC 73062

Problem description:

SWS_SD_00362 has not been removed as intended by RfC # 73062.

Functionalty of SWS_SD_00362 was moved to SWS_SD_00721.

–Last change on issue 76523 comment 1–

Agreed solution:

Remove SWS_SD_00362

–Last change on issue 76523 comment 1–

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 1 | 1 |

1.9 Specification Item SWS_SD_00369

Trace References:

none

Content:

After sending the maximum **amount of** repetitions (defined by SdClientTimerInitialFindRepetitionsMax) of FindService entries, the Repetition Phase shall be left and the Main Phase shall be entered.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73332: [SD] Improvement of SWS_ServiceDiscovery

Problem description:

During reading the SWS_ServiceDiscovery I discovered many minor issues and documented them in the attached files.

Agreed solution:

–[1]- [SWS_SD_00651] and [SWS_SD_00481] are redundant regarding the required goal:

[SWS_SD_00481]

The Service Discovery module shall minimize the amount of messages sent by

combining multiple entries within one message whenever it is possible and not in conflict to the configuration. ()

[SWS_SD_00651]

The amount of separate Service Discovery messages shall be reduced, i.e.: Combine as much information as possible into one Service Discovery message before calling the Socket Adaptors transmit API. This means that when a entry is sent after waiting the appropriate delay (i.e. based on Request-Response-Delay) all other entries for this communication partner may be packed into the Service Discovery message as well.()

remove -SWS_SD_00481

-[2]- Replace the note below requirement SWS_SD_00494 by two explicit new requirements:

+SWS_SD_00xxx_1

If SdServerTimerRequestResponseMinDelay and SdServerTimerRequestResponseMaxDelay are set to the same value, this is the value of the delay.

If SdServerTimerRequestResponseMinDelay and SdServerTimerRequestResponseMaxDelay are set to 0, no delay shall be introduced.

+SWS_SD_00xxx_2

If SdClientTimerRequestResponseMinDelay and SdClientTimerRequestResponseMaxDelay are set to the same value, this is the value of the delay.

If SdClientTimerRequestResponseMinDelay and SdClientTimerRequestResponseMaxDelay are set to 0, no delay shall be introduced.

-[3]- Editorial [SWS_SD_00343]

If the Service Discovery Module receives a FindService Entry the following step(s) shall be performed in the following order:

Send an OfferService Entry considering the appropriate delay (see chapter 7.5.3).

()

=> Remove the string "in the following order", i.e.: If the Service Discovery Module receives a FindService Entry the following step shall be performed: [...]

-[4]- Editorial [SWS_SD_00453] and [SWS_SD_00454]:

"Only if this client was not subscribe before" => "Only if the client was not

subscribed before" [...]

-[5]- Editorial [SWS_SD_00369]:

"After sending the maximum repetitions" => "After sending the maximum amount of repetitions" [...]

–Last change on issue 73332 comment 11–

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 1 | 1 |

1.10 Specification Item SWS_SD_00407

Trace References:

none

Content:

If development error detection is enabled and the Service Discovery module has not been initialized using Sd_Init(), the Sd_ServerServiceSetState function shall raise the development error code SD_E_NOT_INITIALIZED UNINIT and the Sd_ServerServiceSetState function shall return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors

–Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***

SWS_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:

SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:

Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API
UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***

Not affected. No configuration of runtime error reporting required (see SWS BSW
General).

–Last change on issue 59085 comment 88–

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 4 | 1 |

1.11 Specification Item SWS_SD_00410

Trace References:

none

Content:

If development error detection is enabled and the Service Discovery module has not been initialized using Sd_Init(), the Sd_ClientServiceSetState function shall raise the development error code SD_E_NOT_INITIALIZED UNINIT and the Sd_ClientServiceSetState function shall return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors

–Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***

SWS_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function Det_ReportRuntimeError

SWS_LinIf:

SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:

Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API
UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 4 | 1 |

1.12 Specification Item SWS_SD_00446

Trace References:

none

Content:

A reboot of the communication partner shall be detected based on consecutive Service Discovery messages (for communication partner; unicast and multicast separated) in the following two ways:

- Reboot Flag changes from '0' to '1' or
- Session ID ~~decreases~~ **does not increase**, while Reboot Flag stays '1'.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76513: [SD] Contradicting requirements for Service Discovery in Classic Platform and Foundation

Problem description:

SWS_SD_00446 is on contradiction with PRS_SOMEIPSD_00258.

According to the PRS Sd shall detect a reboot if the session ID stays the same (old.session_id>=new.session_id).

According to the SWS_SD Sd shall detect a reboot if the session Id decreases.

–Last change on issue 76513 comment 2–

Agreed solution:

~SWS_SD_00446

Session ID decreases, while Reboot Flag stays 1.

→

Session ID does not increase, while Reboot Flag stays 1.

–Last change on issue 76513 comment 19–

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 4 | 1 |

1.13 Specification Item SWS_SD_00453

Trace References:

none

Content:

If SdEventHandlerTCP is configured: For every SubscribeEventgroup entry of this Event Handler, the following shall be done:

- The relevant Routing Groups shall be identified by SdEventHandlerTcp.
- The relevant TCP Socket Connection of this client shall be identified using the Address/Port of Endpoint Option (UDP) referenced in the SubscribeEventgroup entry and the SdServerServiceTcpRef, or shall be set up, if not existed before.
- Check state of incoming TCP connection using SoAd_GetSoConMode. If mode is not SOAD_SOCON_ONLINE, answer using SubscribeEventgroupNack. Only if **this** **the** client was not **subscribe** **subscribed** before receiving the aforementioned entry:
 - SoAd_EnableSpecificRouting with SdEventActivationRef and the Socket Connection.
 - SoAd_IfSpecificRoutingGroupTransmit with SdEventTriggeringRef and the Socket Connection.
- Answer using SubscribeEventgroup entry.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73332: [SD] Improvement of SWS_ServiceDiscovery

Problem description:

During reading the SWS_ServiceDiscovery I discovered many minor issues and documented them in the attached files.

Agreed solution:

-[1]- [SWS_SD_00651] and [SWS_SD_00481] are redundant regarding the required goal:

[SWS_SD_00481]

The Service Discovery module shall minimize the amount of messages sent by combining multiple entries within one message whenever it is possible and not in conflict to the configuration. ()

[SWS_SD_00651]

The amount of separate Service Discovery messages shall be reduced, i.e.: Combine as much information as possible into one Service Discovery message before calling the Socket Adaptors transmit API. This means that when a entry is sent after waiting the appropriate delay (i.e. based on Request-Response-Delay) all other entries for this communication partner may be packed into the Service

Discovery message as well.()

remove -SWS_SD_00481

-[2]- Replace the note below requirement SWS_SD_00494 by two explicit new requirements:

+SWS_SD_00xxx_1

If SdServerTimerRequestResponseMinDelay and SdServerTimerRequestResponseMaxDelay are set to the same value, this is the value of the delay.

If SdServerTimerRequestResponseMinDelay and SdServerTimerRequestResponseMaxDelay are set to 0, no delay shall be introduced.

+SWS_SD_00xxx_2

If SdClientTimerRequestResponseMinDelay and SdClientTimerRequestResponseMaxDelay are set to the same value, this is the value of the delay.

If SdClientTimerRequestResponseMinDelay and SdClientTimerRequestResponseMaxDelay are set to 0, no delay shall be introduced.

-[3]- Editorial [SWS_SD_00343]

If the Service Discovery Module receives a FindService Entry the following step(s) shall be performed in the following order:

Send an OfferService Entry considering the appropriate delay (see chapter 7.5.3).

()

=> Remove the string "in the following order", i.e.: If the Service Discovery Module receives a FindService Entry the following step shall be performed: [...]

-[4]- Editorial [SWS_SD_00453] and [SWS_SD_00454]:

"Only if this client was not subscribe before" => "Only if the client was not subscribed before" [...]

-[5]- Editorial [SWS_SD_00369]:

"After sending the maximum repetitions" => "After sending the maximum amount of repetitions" [...]

—Last change on issue 73332 comment 11—

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 1 | 1 |

1.14 Specification Item SWS_SD_00454

Trace References:

none

Content:

If SdEventHandlerUDP is configured: For every SubscribeEventgroup entry of this EventHandler, the following shall be done:

- The relevant Routing Groups shall be identified by SdEventHandlerUdp.
- The relevant UDP Socket Connection of this client shall be identified using the Address/Port of Endpoint Option (UDP) referenced in the SubscribeEventgroup entry and the SdServerServiceUdpRef, or shall be set up (SoAd_SetUniqueRemoteAddr()), if not existed before.
 - If no Wildcard Socket Connection is left, SD_E_OUT_OF_RES shall be reported.
- Only if **this the** client was not **subscribe subscribed** before receiving this entry:
 - SoAd_EnableSpecificRouting with SdEventActivationRef and the Socket Connection depending on current number of subscribed clients and the SdEventHandlerMulticastThreshold.
 - SoAd_IfSpecificRoutingGroupTransmit with SdEventTriggeringRef and the Socket Connection.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73332: [SD] Improvement of SWS_ServiceDiscovery

Problem description:

During reading the SWS_ServiceDiscovery I discovered many minor issues and documented them in the attached files.

Agreed solution:

-[1]- [SWS_SD_00651] and [SWS_SD_00481] are redundant regarding the required goal:

[SWS_SD_00481]

The Service Discovery module shall minimize the amount of messages sent by combining multiple entries within one message whenever it is possible and not in conflict to the configuration. ()

[SWS_SD_00651]

The amount of separate Service Discovery messages shall be reduced, i.e.: Combine as much information as possible into one Service Discovery message before calling the Socket Adaptors transmit API. This means that when a entry is sent after waiting the appropriate delay (i.e. based on Request-Response-Delay) all other entries for this communication partner may be packed into the Service Discovery message as well.()

remove -SWS_SD_00481

-[2]- Replace the note below requirement SWS_SD_00494 by two explicit new requirements:

+SWS_SD_00xxx_1

If SdServerTimerRequestResponseMinDelay and SdServerTimerRequestResponseMaxDelay are set to the same value, this is the value of the delay.

If SdServerTimerRequestResponseMinDelay and SdServerTimerRequestResponseMaxDelay are set to 0, no delay shall be introduced.

+SWS_SD_00xxx_2

If SdClientTimerRequestResponseMinDelay and SdClientTimerRequestResponseMaxDelay are set to the same value, this is the value of the delay.

If SdClientTimerRequestResponseMinDelay and SdClientTimerRequestResponseMaxDelay are set to 0, no delay shall be introduced.

-[3]- Editorial [SWS_SD_00343]

If the Service Discovery Module receives a FindService Entry the following step(s) shall be performed in the following order:

Send an OfferService Entry considering the appropriate delay (see chapter 7.5.3).

()

=> Remove the string "in the following order", i.e.: If the Service Discovery Module receives a FindService Entry the following step shall be performed: [...]

-[4]- Editorial [SWS_SD_00453] and [SWS_SD_00454]:

"Only if this client was not subscribe before" => "Only if the client was not subscribed before" [...]

-[5]- Editorial [SWS_SD_00369]:

"After sending the maximum repetitions" => "After sending the maximum amount of repetitions" [...]

–Last change on issue 73332 comment 11–

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 1 | 1 |

1.15 Specification Item SWS_SD_00455

Trace References:

none

Content:

The SdEventHandlerMulticastThreshold shall be used to control when to enable/disable Unicast/Multicast by using SoAd_EnableSpecificRouting and SoAd_DisableSpecificRouting:

- If SdEventHandlerMulticastThreshold = 0: Setup Unicast to every subscribed client (Multicast always disabled).
- If SdEventHandlerMulticastThreshold = 1: Setup Multicast if one or more clients are subscribed (Unicast always disabled).
- If SdEventHandlerMulticastThreshold > 1:
 - Setup Unicast for all subscribed clients if number of subscribed clients < Sd EventHandlerMulticastThreshold,
 - else setup Multicast. Switch dynamically based on the number of subscribed clients:
 - With 0 clients: nothing enabled.
 - With clients < threshold: unicast for subscribed clients enabled. Multicast disabled.
 - With clients \geq threshold: multicast enabled. Unicast disabled.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #68823: [Sd] Switching of EventHandler from Unicast to Multicast.

Problem description:

If the SdEventHandlerMulticastThreshold is set to say 3, and at the current moment two Consumed eventgroups (say A and B) are subscribed to an EventHandler. These are being served by Unicast as per [SWS_SD_00455], the SubscribeEventGroupAck having previously been sent to these consumed eventgroups.

Now if a new SubscribeEventGroup (C) is received, the Event Handler will shift these three to multicast as per [SWS_SD_00455].

Will it also send a new SubscribeEventGroupAck to the Consumed eventgroups A and B to indicate that the events are now sent over multicast?

If so, what are the actions on the client side for the consumed eventgroups A and B?

Agreed solution:

~SWS_SD_00455

"SoAd_DisableSpecific Routing" -> "SoAd_DisableSpecificRouting"

~Figure 9.6

"SoAd_GetRemoteAddr(return, SoConId, IpAddrPtr)" ->

"SoAd_GetRemoteAddr(return, SdMulticastEventSoConRef, IpAddrPtr)"

~Figure 9.8

"SoAd_EnableRouting(SdEventHandlerTcp->SdEventActivationRef)" ->

"SoAd_EnableSpecificRouting(SdEventHandlerTcp->SdEventActivationRef, SoConId)"

~Alt [MulticastThreshhold==0 || numOfSubs<MulticastThreshhold]

"SoAd_EnableRouting(SdEventHandlerUdp->SdEventActivationRef)" ->

"SoAd_EnableSpecificRouting(SdEventHandlerUdp->SdEventActivationRef, SoConId)"

~Alt [numOfSubs=MulticastThreshhold]

"[numOfSubs=MulticastThreshhold]" -> "[numOfSubs==MulticastThreshhold]"

"SoAd_EnableRouting(SdEventHandlerMulticast->SdEventActivationRef)" ->

"SoAd_EnableSpecificRouting(SdEventHandlerMulticast->SdEventActivationRef, SdEventHandlerMulticast->SdMulticastEventSoConRef)"

"SoAd_DisableRouting(SdEventHandlerUdp->SdEventActivationRef)" ->

"SoAd_DisableSpecificRouting(SdEventHandlerUdp->SdEventActivationRef, SoConId)"

-Alt [numOfSubs>=MulticastThreshold]

~ECUC_SD_00096

Add at end of description: "This is usually equal to the SdEventActivationRef referenced by SdEventHandlerUdp".

~ECUC_SD_00106

Add at end of description: "This is usually equal to the SdConsumedEventGroupUdpActivationRef".

~ECUC_SD_00118

Change multiplicity of SdMulticastEventSoConRef from optional to mandatory (0..1 to 1).

–Last change on issue 68823 comment 44–

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 3 | 1 |

1.16 Specification Item SWS_SD_00469

Trace References:

none

Content:

If development error detection is enabled and the Service Discovery module has not been initialized using Sd_Init(), the Sd_ConsumedEventGroupSetState function shall raise the development error code SD_E_NOT_INITIALIZED UNINIT and the Sd_ConsumedEventGroupSetState function shall return E_NOT_OK.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors

–Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***

SWS_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:

SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:

Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API
UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

—Last change on issue 59085 comment 88—

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 4 | 1 |

1.17 Specification Item SWS_SD_00471

Trace References:

none

Content:

If development error detection is enabled and the Service Discovery module has not been initialized using `Sd_Init()`, the `Sd_LocalIpAddrAssignmentChg` function shall raise the development error code `SD_E_NOT_INITIALIZED UNINIT` and the `Sd_LocalIpAddrAssignmentChg` function shall return without further action.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors
–Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***

SWS_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function `Det_ReportRunTimeError`

SWS_LinIf:

SWS_LinIf_00359: add `Det_ReportRuntimeError`

SWS_UdpNm:

Replace `UDPNM_E_NO_INIT` with `UDPNM_E_UNINIT` in description of API `UdpNm_MainFunction_<Instance Id>` (SWS_UdpNm_00234)

*** ECUC XML ***

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

–Last change on issue 59085 comment 88–

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 4 | 1 |

1.18 Specification Item SWS_SD_00473

Trace References:

none

Content:

If development error detection is enabled and the Service Discovery module has not been initialized using Sd_Init(), the Sd_RxIndication function shall raise the development error code SD_E_NOT_INITIALIZED UNINIT and the Sd_RxIndication function shall return without further action.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

Problem description:

Inconsistencies in SWS with semantics of Default errors

–Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment
<https://www.autosar.org/bugzilla/attachment.cgi?id=4604>

Notes:

- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***

SWS_CanNm:

Chapter 8.6.1 Optional Interfaces:

Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:

SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:

Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***

Not affected. No configuration of runtime error reporting required (see SWS BSW General).

—Last change on issue 59085 comment 88—

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 4 | 1 |

1.19 Specification Item SWS_SD_00481

Trace References:

none

Content:

The Service Discovery module shall minimize the amount of messages sent by combining multiple entries within one message whenever it is possible and not in conflict to the configuration.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73332: [SD] Improvement of SWS_ServiceDiscovery

Problem description:

During reading the SWS_ServiceDiscovery I discovered many minor issues and documented them in the attached files.

Agreed solution:

-[1]- [SWS_SD_00651] and [SWS_SD_00481] are redundant regarding the required goal:

[SWS_SD_00481]

The Service Discovery module shall minimize the amount of messages sent by combining multiple entries within one message whenever it is possible and not in conflict to the configuration. ()

[SWS_SD_00651]

The amount of separate Service Discovery messages shall be reduced, i.e.: Combine as much information as possible into one Service Discovery message before calling the Socket Adaptors transmit API. This means that when a entry is sent after waiting the appropriate delay (i.e. based on Request-Response-Delay) all other entries for this communication partner may be packed into the Service Discovery message as well.()

remove -SWS_SD_00481

-[2]- Replace the note below requirement SWS_SD_00494 by two explicit new requirements:

+SWS_SD_00xxx_1

If SdServerTimerRequestResponseMinDelay and SdServerTimerRequestResponseMaxDelay are set to the same value, this is the value of the delay.

If SdServerTimerRequestResponseMinDelay and SdServerTimerRequestResponseMaxDelay are set to 0, no delay shall be introduced.

+SWS_SD_00xxx_2

If SdClientTimerRequestResponseMinDelay and SdClientTimerRequestResponseMaxDelay are set to the same value, this is the value of the delay.

If SdClientTimerRequestResponseMinDelay and SdClientTimerRequestResponseMaxDelay are set to 0, no delay shall be introduced.

-[3]- Editorial [SWS_SD_00343]

If the Service Discovery Module receives a FindService Entry the following step(s) shall be performed in the following order:

Send an OfferService Entry considering the appropriate delay (see chapter 7.5.3).

()

=> Remove the string "in the following order", i.e.: If the Service Discovery Module receives a FindService Entry the following step shall be performed: [...]

-[4]- Editorial [SWS_SD_00453] and [SWS_SD_00454]:

"Only if this client was not subscribe before" => "Only if the client was not subscribed before" [...]

-[5]- Editorial [SWS_SD_00369]:

"After sending the maximum repetitions" => "After sending the maximum amount of repetitions" [...]

–Last change on issue 73332 comment 11–

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 1 | 1 |

1.20 Specification Item SWS_SD_00723

Trace References:

none

Content:

During initialization of the SD module, the API SoAd_OpenSoCon() shall be called for all Socket Connections associated with SdInstanceTxPdu, SdInstanceUnicastRxPdu and SdInstanceMulticastRxPdu.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #75555: [Sd] Clarification of SD Control Path Connection

Problem description:

According SWS_SD_00606 & SWS_SD_00604 : SoAd_OpenSoCon should be done for all Socket Connections associated with this Server Service Instance.
Does "Server Associated" connection includes SD Control path that is Socket Connections for SD frames ?

If No then where this control path suppose to be opened for client as well as server services ?

Note :

According to Bugzilla 71026 AutoSoConEnable should be False for Tx SoCon , Rx Unicast SoCon & Rx Multicast SoCon.

Agreed solution:

+SWS_SD_00xxx after SWS_SD_00697

During initialization of the SD module, the API SoAd_OpenSoCon() shall be called for all Socket Connections associated with SdInstanceTxPdu, SdInstanceUnicastRxPdu and SdInstanceMulticastRxPdu.

Note: The SoAd module needs to be initialized before the SD module is initialized.

—Last change on issue 75555 comment 8—

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 1 | 1 |

1.21 Specification Item SWS_SD_00724

Trace References:

none

Content:

If SdServerTimerRequestResponseMinDelay and SdServerTimerRequestResponseMaxDelay are set to the same value, this value shall be used as delay.

If SdServerTimerRequestResponseMinDelay and SdServerTimerRequestResponseMaxDelay are set to 0, no delay shall be introduced.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73332: [SD] Improvement of SWS_ServiceDiscovery

Problem description:

During reading the SWS_ServiceDiscovery I discovered many minor issues and documented them in the attached files.

Agreed solution:

-[1]- [SWS_SD_00651] and [SWS_SD_00481] are redundant regarding the required goal:

[SWS_SD_00481]

The Service Discovery module shall minimize the amount of messages sent by combining multiple entries within one message whenever it is possible and not in conflict to the configuration. ()

[SWS_SD_00651]

The amount of separate Service Discovery messages shall be reduced, i.e.: Combine as much information as possible into one Service Discovery message before calling the Socket Adaptors transmit API. This means that when a entry is sent after waiting the appropriate delay (i.e. based on Request-Response-Delay) all other entries for this communication partner may be packed into the Service Discovery message as well.()

remove -SWS_SD_00481

-[2]- Replace the note below requirement SWS_SD_00494 by two explicit new requirements:

+SWS_SD_00xxx_1

If SdServerTimerRequestResponseMinDelay and SdServerTimerRequestResponseMaxDelay are set to the same value, this is the value of the delay.

If SdServerTimerRequestResponseMinDelay and SdServerTimerRequestResponseMaxDelay are set to 0, no delay shall be introduced.

+SWS_SD_00xxx_2

If SdClientTimerRequestResponseMinDelay and SdClientTimerRequestResponseMaxDelay are set to the same value, this is the value of the delay.

If SdClientTimerRequestResponseMinDelay and SdClientTimerRequestResponseMaxDelay are set to 0, no delay shall be introduced.

-[3]- Editorial [SWS_SD_00343]

If the Service Discovery Module receives a FindService Entry the following step(s) shall be performed in the following order:

Send an OfferService Entry considering the appropriate delay (see chapter 7.5.3).

()

=> Remove the string "in the following order", i.e.: If the Service Discovery Module receives a FindService Entry the following step shall be performed: [...]

-[4]- Editorial [SWS_SD_00453] and [SWS_SD_00454]:

"Only if this client was not subscribe before" => "Only if the client was not subscribed before" [...]

-[5]- Editorial [SWS_SD_00369]:

"After sending the maximum repetitions" => "After sending the maximum amount of repetitions" [...]

–Last change on issue 73332 comment 11–

BW-C-Level:

| Application | Specification | Bus |
|--------------------|----------------------|------------|
| 1 | 1 | 1 |

1.22 Specification Item SWS_SD_00725

Trace References:

none

Content:

If `SdClientTimerRequestResponseMinDelay` and `SdClientTimerRequestResponseMaxDelay` are set to the same value, this value shall be used as delay.

If `SdClientTimerRequestResponseMinDelay` and `SdClientTimerRequestResponseMaxDelay` are set to 0, no delay shall be introduced.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73332: [SD] Improvement of SWS_ServiceDiscovery

Problem description:

During reading the SWS_ServiceDiscovery I discovered many minor issues and documented them in the attached files.

Agreed solution:

-[1]- [SWS_SD_00651] and [SWS_SD_00481] are redundant regarding the required goal:

[SWS_SD_00481]

The Service Discovery module shall minimize the amount of messages sent by combining multiple entries within one message whenever it is possible and not in conflict to the configuration. ()

[SWS_SD_00651]

The amount of separate Service Discovery messages shall be reduced, i.e.: Combine as much information as possible into one Service Discovery message before calling the Socket Adaptors transmit API. This means that when a entry is sent after waiting the appropriate delay (i.e. based on Request-Response-Delay) all other entries for this communication partner may be packed into the Service Discovery message as well.()

remove -SWS_SD_00481

-[2]- Replace the note below requirement SWS_SD_00494 by two explicit new requirements:

+SWS_SD_00xxx_1

If SdServerTimerRequestResponseMinDelay and SdServerTimerRequestResponseMaxDelay are set to the same value, this is the value of the delay.

If SdServerTimerRequestResponseMinDelay and SdServerTimerRequestResponseMaxDelay are set to 0, no delay shall be introduced.

+SWS_SD_00xxx_2

If SdClientTimerRequestResponseMinDelay and SdClientTimerRequestResponseMaxDelay are set to the same value, this is the value of the delay.

If SdClientTimerRequestResponseMinDelay and SdClientTimerRequestResponseMaxDelay are set to 0, no delay shall be introduced.

-[3]- Editorial [SWS_SD_00343]

If the Service Discovery Module receives a FindService Entry the following step(s) shall be performed in the following order:

Send an OfferService Entry considering the appropriate delay (see chapter 7.5.3).

()

=> Remove the string "in the following order", i.e.: If the Service Discovery Module receives a FindService Entry the following step shall be performed: [...]

-[4]- Editorial [SWS_SD_00453] and [SWS_SD_00454]:

"Only if this client was not subscribe before" => "Only if the client was not subscribed before" [...]

-[5]- Editorial [SWS_SD_00369]:

"After sending the maximum repetitions" => "After sending the maximum amount of repetitions" [...]

–Last change on issue 73332 comment 11–

BW-C-Level:

| Application | Specification | Bus |
|-------------|---------------|-----|
| 1 | 1 | 1 |