

Document Title	Complete Change Documentation 1.4.0 - 1.5.0
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	885

Document Status	Final
Part of AUTOSAR Standard	Foundation
Part of Standard Release	1.5.0

Table of Contents

1	PRS_E2EProtocol	7
2	PRS_LogAndTraceProtocol	7
2.1	Specification Item PRS_Dlt_00635	7
2.2	Specification Item PRS_Dlt_00770	9
3	PRS_NetworkManagementProtocol	11
3.1	Specification Item PRS_Nm_00077	11
3.2	Specification Item PRS_Nm_00334	11
3.3	Specification Item PRS_Nm_00337	12
3.4	Specification Item PRS_Nm_00340	12
4	PRS_SOMEIPProtocol	13
4.1	Specification Item PRS_SOMEIP_00215	13
4.2	Specification Item PRS_SOMEIP_00223	16
4.3	Specification Item PRS_SOMEIP_00241	17
4.4	Specification Item PRS_SOMEIP_00242	20
4.5	Specification Item PRS_SOMEIP_00243	23
4.6	Specification Item PRS_SOMEIP_00244	26
4.7	Specification Item PRS_SOMEIP_00925	29
4.8	Specification Item PRS_SOMEIP_00941	30
5	PRS_SOMEIPServiceDiscoveryProtocol	32
5.1	Specification Item PRS_SOMEIPSD_00124	32
5.2	Specification Item PRS_SOMEIPSD_00125	37
5.3	Specification Item PRS_SOMEIPSD_00126	41
5.4	Specification Item PRS_SOMEIPSD_00127	45
5.5	Specification Item PRS_SOMEIPSD_00128	49
5.6	Specification Item PRS_SOMEIPSD_00129	54
5.7	Specification Item PRS_SOMEIPSD_00130	58
5.8	Specification Item PRS_SOMEIPSD_00231	63
5.9	Specification Item PRS_SOMEIPSD_00232	67
5.10	Specification Item PRS_SOMEIPSD_00233	71
5.11	Specification Item PRS_SOMEIPSD_00234	76
5.12	Specification Item PRS_SOMEIPSD_00235	80
5.13	Specification Item PRS_SOMEIPSD_00273	84
5.14	Specification Item PRS_SOMEIPSD_00275	89
5.15	Specification Item PRS_SOMEIPSD_00276	93
5.16	Specification Item PRS_SOMEIPSD_00307	97
5.17	Specification Item PRS_SOMEIPSD_00315	102
5.18	Specification Item PRS_SOMEIPSD_00326	106

5.19	Specification Item PRS_SOMEIPSD_00333	111
5.20	Specification Item PRS_SOMEIPSD_00544	116
5.21	Specification Item PRS_SOMEIPSD_00551	120
5.22	Specification Item PRS_SOMEIPSD_00552	121
5.23	Specification Item PRS_SOMEIPSD_00558	126
5.24	Specification Item PRS_SOMEIPSD_00559	127
5.25	Specification Item PRS_SOMEIPSD_00653	132
5.26	Specification Item PRS_SOMEIPSD_00710	133
5.27	Specification Item PRS_SOMEIPSD_00711	134
5.28	Specification Item PRS_SOMEIPSD_00713	135
5.29	Specification Item PRS_SOMEIPSD_00714	136
5.30	Specification Item PRS_SOMEIPSD_00803	138
5.31	Specification Item PRS_SOMEIPSD_00822	142
5.32	Specification Item PRS_SOMEIPSD_00823	143
5.33	Specification Item PRS_SOMEIPSD_00824	144
6	PRS_TimeSyncProtocol	146
7	RS_E2E	146
8	RS_FoundationDebugTraceProfile	146
9	RS_HealthMonitoring	146
10	RS_LogAndTrace	146
10.1	Specification Item RS_LT_00001	146
10.2	Specification Item RS_LT_00002	147
10.3	Specification Item RS_LT_00003	148
10.4	Specification Item RS_LT_00004	149
10.5	Specification Item RS_LT_00005	150
10.6	Specification Item RS_LT_00006	150
10.7	Specification Item RS_LT_00007	151
10.8	Specification Item RS_LT_00008	152
10.9	Specification Item RS_LT_00009	153
10.10	Specification Item RS_LT_00010	154
10.11	Specification Item RS_LT_00011	155
10.12	Specification Item RS_LT_00012	156
10.13	Specification Item RS_LT_00013	157
10.14	Specification Item RS_LT_00014	158
10.15	Specification Item RS_LT_00016	159
10.16	Specification Item RS_LT_00017	160
10.17	Specification Item RS_LT_00018	160
10.18	Specification Item RS_LT_00019	161
10.19	Specification Item RS_LT_00020	162

10.20	Specification Item RS_LT_00021	163
10.21	Specification Item RS_LT_00022	164
10.22	Specification Item RS_LT_00023	165
10.23	Specification Item RS_LT_00024	166
10.24	Specification Item RS_LT_00025	166
10.25	Specification Item RS_LT_00026	167
10.26	Specification Item RS_LT_00027	168
10.27	Specification Item RS_LT_00028	169
10.28	Specification Item RS_LT_00029	170
10.29	Specification Item RS_LT_00030	171
10.30	Specification Item RS_LT_00031	172
10.31	Specification Item RS_LT_00032	173
10.32	Specification Item RS_LT_00033	174
10.33	Specification Item RS_LT_00034	175
10.34	Specification Item RS_LT_00035	176
10.35	Specification Item RS_LT_00036	176
10.36	Specification Item RS_LT_00037	177
10.37	Specification Item RS_LT_00038	178
10.38	Specification Item RS_LT_00039	179
10.39	Specification Item RS_LT_00040	180
10.40	Specification Item RS_LT_00041	181
10.41	Specification Item RS_LT_00042	182
10.42	Specification Item RS_LT_00056	182
10.43	Specification Item RS_LT_00057	183
11	RS_Main	185
11.1	Specification Item RS_Main_00010	185
11.2	Specification Item RS_Main_00012	186
11.3	Specification Item RS_Main_00050	187
11.4	Specification Item RS_Main_00120	188
11.5	Specification Item RS_Main_00125	189
11.6	Specification Item RS_Main_00160	190
11.7	Specification Item RS_Main_00161	191
11.8	Specification Item RS_Main_00170	192
11.9	Specification Item RS_Main_00220	197
11.10	Specification Item RS_Main_00250	199
11.11	Specification Item RS_Main_00350	200
11.12	Specification Item RS_Main_00460	201
11.13	Specification Item RS_Main_00503	203
11.14	Specification Item RS_Main_00510	208
11.15	Specification Item RS_Main_00514	210
11.16	Specification Item RS_Main_00650	212
11.17	Specification Item RS_Main_00651	216

11.18	Specification Item RS_Main_01025	218
11.19	Specification Item RS_Main_01026	222
12	RS_Methodology	226
12.1	Specification Item RS_METH_00062	226
12.2	Specification Item RS_METH_00074	227
12.3	Specification Item RS_METH_00075	229
12.4	Specification Item RS_METH_00076	231
12.5	Specification Item RS_METH_00203	232
12.6	Specification Item RS_METH_00208	234
13	RS_NetworkManagement	235
13.1	Specification Item RS_Nm_00044	235
13.2	Specification Item RS_Nm_00148	236
13.3	Specification Item RS_Nm_02510	237
13.4	Specification Item RS_Nm_02529	237
13.5	Specification Item RS_Nm_02530	238
13.6	Specification Item RS_Nm_02531	239
13.7	Specification Item RS_Nm_02532	239
13.8	Specification Item RS_Nm_02533	240
13.9	Specification Item RS_Nm_02534	240
13.10	Specification Item RS_Nm_02535	241
14	RS_ProjectObjectives	242
15	RS_SOMEIPProtocol	242
16	RS_SOMEIPServiceDiscoveryProtocol	242
17	RS_TimeSync	242
18	SRS_Diagnostics	242
18.1	Specification Item SRS_Diag_04230	242
18.2	Specification Item SRS_Diag_04231	243
18.3	Specification Item SRS_Diag_04232	243
18.4	Specification Item SRS_Diag_04233	244
18.5	Specification Item SRS_Diag_04234	245
18.6	Specification Item SRS_Diag_04235	245
18.7	Specification Item SRS_Diag_04236	246
18.8	Specification Item SRS_Diag_04237	246
18.9	Specification Item SRS_Diag_04238	247
18.10	Specification Item SRS_Diag_04239	248
18.11	Specification Item SRS_Diag_04240	248



1 PRS_E2EProtocol

2 PRS_LogAndTraceProtocol

2.1 Specification Item PRS_Dlt_00635

Trace References:

RS_LT_00032

Content:

The following Dlt Commands using the following Services IDs shall be supported:

Service ID	Dlt Command Name	Description
0x01	SetLogLevel	Set the Log Level
0x02	SetTraceStatus	Enable/Disable Trace Messages
0x03	GetLogInfo	Returns the LogLevel for registered applications
0x04	GetDefaultLogLevel	Returns the LogLevel for wildcards
0x05	StoreConfiguration	Stores the current configuration non volatile
0x06	RestoreToFactoryDefault	Sets the configuration back to default
0x0A	SetMessageFiltering	Enable/Disable message filtering
0x11	SetDefaultLogLevel	Sets the LogLevel for wildcards
0x12	SetDefaultTraceStatus	Enable/Disable TraceMessages for wildcards
0x13	GetSoftwareVersion	Get the ECU software version
0x15	GetDefaultTraceStatus	Get the current TraceLevel for wildcards
0x17	GetLogChannelNames	Returns the LogChannel's name
0x1F	GetTraceStatus	Returns the current TraceStatus
0x20	SetLogChannelAssignment	Adds/ Removes the given LogChannel as output path
0x21	SetLogChannelThreshold	Sets the filter threshold for the given Log Channel
0x22	GetLogChannelThreshold	Returns the current LogLevel for a given LogChannel
0x23	BufferOverflowNotification	Report that a buffer overflow occurred
0x24	SyncTimeStamp	Reports synchronized absolute time

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #78317: [DLT] Add explicit time stamp exchange to enable synchronized tracing of multiple ECUs

Problem description:

To make usage of nodes synchronized using the StbM or another time synchronization technology, it is necessary to extend the exchange of time stamps in the DLT protocol by an exchange of explicit base time stamps.

Agreed solution:

The following change are to be done:

1. [PRS_Dlt_00635] Add a new line for the following Dlt command:

Service ID: 0x24

Dlt Command Name: SyncTimeStamp

Description: Reports synchronized absolute time

2. Add new chapter after 5.3.17-> 5.3.18 Synchronized TimeStamp

3. Add new requirement in new chapter (5.3.18) as follows:

Service name: SyncTimeStamp

Service ID [hex]: 0x24

Sync/Async: Synchronous

Reentrancy: Not Reentrant

Request Parameter

Number :none

Response parameter:

Number: 1

Name: status

Type: uint8

Description: 0==OK, 1==NOT_SUPPORTED, 2==ERROR

Number: 2

Name: synctimestamp

Type: TimeStamp

Description: structure contains time stamp which represents the synchronized absolute time starting from 1970-01-01.

Unit32 Nanoseconds part of the time

Unit 32 Seconds part of the time

Unit16 Seconds part of the time (MSB)

Description: The Dlt module sends this message to report the synchronized absolute time. The message will be sent once when message transmission is started and then every 10 minutes. If not supported, NOT SUPPORTED shall be reported. The message shall also be sent after a jump in the time is detected, e.g. leap time or synchronization status.

—Last change on issue 78317 comment 35—

BW-C-Level:

Application	Specification	Bus
1	1	1

2.2 Specification Item PRS_Dlt_00770

Trace References:

RS_LT_00037

Content:

Service name:		SyncTimeStamp	
Service ID [hex]		0x24	
:			
Sync/Async:		Synchronous	
Reentrancy:		Non Reentrant	
Request Parameter			
Number	Name	Type	Description
Response Parameter			
Number	Name	Type	Description
1	status	uint8	0 == OK 1 == NOT_SUPPORTED 2 == ERROR
2	synctimestamp	TimeStamp	structure contains time stamp which represents the synchronized absolute time starting from 1970-01-01. Unit32 Nanoseconds part of the time Unit 32 Seconds part of the time Unit16 Seconds part of the time (MSB)





Description:	The Dlt module sends this message to report the synchronized absolute time. The message will be sent once when message transmission is started and then every 10 minutes. If not supported, NOT SUPPORTED shall be reported. The message shall also be sent after a jump in the time is detected, e.g. leap time or synchronization status.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #78317: [DLT] Add explicit time stamp exchange to enable synchronized tracing of multiple ECUs

Problem description:

To make usage of nodes synchronized using the StbM or another time synchronization technology, it is necessary to extend the exchange of time stamps in the DLT protocol by an exchange of explicit base time stamps.

Agreed solution:

The following change are to be done:

1. [PRS_Dlt_00635] Add a new line for the following Dlt command:

Service ID: 0x24

Dlt Command Name: SyncTimeStamp

Description: Reports synchronized absolute time

2. Add new chapter after 5.3.17-> 5.3.18 Synchronized TimeStamp

3. Add new requirement in new chapter (5.3.18) as follows:

Service name: SyncTimeStamp

Service ID [hex]: 0x24

Sync/Async: Synchronous

Reentrancy: Not Reentrant

Request Parameter

Number :none

Response parameter:

Number: 1

Name: status

Type: uint8

Description: 0==OK, 1==NOT_SUPPORTED, 2==ERROR

Number: 2

Name: synctimestamp

Type: TimeStamp

Description: structure contains time stamp which represents the synchronized absolute time starting from 1970-01-01.

Unit32 Nanoseconds part of the time

Unit 32 Seconds part of the time

Unit16 Seconds part of the time (MSB)

Description: The Dlt module sends this message to report the synchronized absolute time. The message will be sent once when message transmission is started and then every 10 minutes. If not supported, NOT SUPPORTED shall be reported. The message shall also be sent after a jump in the time is detected, e.g. leap time or synchronization status.

–Last change on issue 78317 comment 35–

BW-C-Level:

Application	Specification	Bus
1	1	1

3 PRS_NetworkManagementProtocol

3.1 Specification Item PRS_Nm_00077

Trace References:

none

Content:

The length (in bytes) of the user data in a NM message shall be **defined** **configured** by [UserDataLength].

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

3.2 Specification Item PRS_Nm_00334

Trace References:

none

Content:

When the "Repeat Message" state is entered because of network request or repeat message request and configured number of immediate NM transmissions is greater than zero (see NmImmediateNmTransmissions), ~~the~~ **these immediate** NM messages shall be transmitted using Immediate NM Cycle Time (see NmImmediateNmCycleTime).

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

3.3 Specification Item PRS_Nm_00337

Trace References:

RS_Nm_02520

Content:

~~The~~ **If configured to do so the** NM algorithm shall only process **messages containing** PN request information if ~~it contains~~ **they contain** at least one bit set to 1 that corresponds to a PNC **which is** relevant for the ECU.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

3.4 Specification Item PRS_Nm_00340

Trace References:

RS_Nm_02518, RS_Nm_02520

Content:

If one PNC is not requested again (relevant PNC bit **is not** set to 1 **again**) within [PnReset Time] this PN shall be considered as "not requested".

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

4 PRS_SOMEIPProtocol

4.1 Specification Item PRS_SOMEIP_00215

Trace References:

RS_SOMEIP_00050

Content:

If length fields of static size are used (wire type 4), the size of the length field for arrays, structs and strings shall be identical and shall be greater than 0 in the configuration. Moreover, the size of the length field shall be configured for the top-level struct or method request/response. All arrays, structs and strings used within a struct or all arguments within a method shall inherit the size of the length field from the top-level definition.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80991: Serialization of structures with identifier and optional members always requires static definition of struct length field size

Problem description:

Name: Daniel Kerk
Phone:
Role: OEM

Description/Motivation:

For the serialization of structures with identifier and optional members (TLV serialization), the SOME/IP Protocol Specification currently specifies that the size of length fields for arrays, structs and strings shall be identical shall be greater than zero when length fields of static size are used, i.e. usage of wire type 4 (see PRS_SOMEIP_00215).

This implies that if length fields of dynamic size are used, the size of the length fields do not need to be defined statically.

This is true when the length field is preceded by a tag. The size of the length field for arrays, strings or structs as members of a parent struct can then be encoded in the wire type.

But there are cases where a length field is not preceded by a tag:
- the top-level struct (e.g. contained in a SOME/IP event)

- an array of structs, strings or array

The current specification implies that in case of usage of length fields of dynamic size, in the above mentioned cases, no length fields are added.

This is critical because when optional members come into play, this might lead to ambiguities.

Example a)

```
struct s
uint8 a; // optional
uint8 b; // optional
```

`s[] = 0xAA, 0xBB // one entry in array, both optional members exist`

will result in the following serialized byte stream:

```
vector len tlv-tag a tlv-tag b
0x00000006 0x0011 0xAA 0x0022 0xBB
```

According to the current specification, the following example will result in the same serialized byte stream:

Example b)

`s[] = 0xAA, 0xBB // two entries in array, only one optional member exists in each`

Without length fields, the deserializer cannot determine the boundaries of the structs within the array.

So even if length fields of dynamic size are used, a static length field size must be configured to cover the length fields which are not preceded by a tag.

Using length fields here will result in unambiguous serializations:

a)

```
vector len struct length tlv-tag a tlv-tag b
0x0000000A 0x00000006 0x0011 0xAA 0x0022 0xBB
```

b)

vector len struct length tlv-tag a struct length tlv-tag b
0x0000000E 0x00000003 0x0011 0xAA 0x00000003 0x0022 0xBB

Please also refer to AP-6496 in the AP JIRA.

This bug is closely related to the TLV concept incorporation, so we should try to fix it for FO_R1.5.

Agreed solution:

Rework and split PRS_SOMEIP_00215 as follows:

+**[PRS_SOMEIP_xxxx1]** The size of the length field for arrays, structs, unions and string shall be greater than 0 in the configuration. (RS_SOMEIP_00050)

Rationale: The TLV serialization requires the usage of length fields. When wire type 4 is used, the length field size must be statically configured. When wire types 5-7 (dynamic length field size) are used, the static configuration of the length field size must also be present since not all length fields are preceded by a tag, e.g. structs contained in an array or the top-level struct contained in a SOME/IP event. Not using length fields here would result in ambiguities.

+**[PRS_SOMEIP_xxxx2]** The configured size of the length field for arrays, structs, unions and strings shall be identical. (RS_SOMEIP_00050)

Rationale: In case of an unknown member or argument, the deserializer cannot determine the actual datatype of the member/argument when wire type 4 is used.

+**[PRS_SOMEIP_xxxx3]** The size of the length field shall be configured for the top-level struct or method request/response. All arrays, unions, structs and strings used within a struct or all arguments within a method shall inherit the size of the length field from the top-level definition. (RS_SOMEIP_00050)

Rationale: In case of an unknown member or argument, the deserializer needs to know the size of the length field when wire type 4 is used. The easiest way is that the size of the length field is then only defined at the top-level element.

+**[PRS_SOMEIP_xxxx4]** Overriding the size of the length field at a subordinate array, union, struct or string or at an individual method argument shall not be allowed. (RS_SOMEIP_00050)

~[PRS_SOMEIP_00215]

~Last change on issue 80991 comment 31~

BW-C-Level:

Application	Specification	Bus
1	1	1

4.2 Specification Item PRS_SOMEIP_00223

Trace References:

RS_SOMEIP_00050

Content:

The deserializer shall ignore **if** optional members/arguments **which** are not available in the serialized **byte** stream.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80339: Some statements in the document 'PRS SOMEIPProtocol' require further correction

Problem description:

During the document review two statements to be clarified were identified, listed below:

1. Section 4 Protocol Specification, [PRS_SOMEIP_00223], Meaning (and typo): The deserializer shall ignore (what?) if optional members/arguments are not available in the serialized byte stream.

2. Section 4 Protocol Specification, [PRS_SOMEIP_00754], Syntax: Double if statement in a single sentence should be avoided. Using conjunction statement (with and) is recommended.

Agreed solution:

~[PRS_SOMEIP_00223] DRAFT d The deserializer shall ignore optional members/arguments

which are not available in the serialized byte stream. c(RS_SOMEIP_00050)

~Last change on issue 80339 comment 7~

BW-C-Level:

Application	Specification	Bus
1	1	1

4.3 Specification Item PRS_SOMEIP_00241

Trace References:

RS_SOMEIP_00050

Content:

The size of the length field for arrays, structs, unions and string shall be greater than 0 in the configuration.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80991: Serialization of structures with identifier and optional members always requires static definition of struct length field size

Problem description:

Name: Daniel Kerk
Phone:
Role: OEM

Description/Motivation:

For the serialization of structures with identifier and optional members (TLV serialization), the SOME/IP Protocol Specification currently specifies that the size of length fields for arrays, structs and strings shall be identical shall be greater than zero when length fields of static size are used, i.e. usage of wire type 4 (see PRS_SOMEIP_00215).

This implies that if length fields of dynamic size are used, the size of the length fields do not need to be defined statically.

This is true when the length field is preceded by a tag. The size of the length field for arrays, strings or structs as members of a parent struct can then be encoded in the wire type.

But there are cases where a length field is not preceded by a tag:

- the top-level struct (e.g. contained in a SOME/IP event)
- an array of structs, strings or array

The current specification implies that in case of usage of length fields of dynamic size, in the above mentioned cases, no length fields are added.

This is critical because when optional members come into play, this might lead to ambiguities.

Example a)

```
struct s
uint8 a; // optional
uint8 b; // optional
```

`s[] = 0xAA, 0xBB` // one entry in array, both optional members exist

will result in the following serialized byte stream:

```
vector len tlv-tag a tlv-tag b
0x00000006 0x0011 0xAA 0x0022 0xBB
```

According to the current specification, the following example will result in the same serialized byte stream:

Example b)

`s[] = 0xAA, 0xBB` // two entries in array, only one optional member exists in each

Without length fields, the deserializer cannot determine the boundaries of the structs within the array.

So even if length fields of dynamic size are used, a static length field size must be configured to cover the length fields which are not preceded by a tag.

Using length fields here will result in unambiguous serializations:

a)

```
vector len struct length tlv-tag a tlv-tag b
0x0000000A 0x00000006 0x0011 0xAA 0x0022 0xBB
```

b)

```
vector len struct length tlv-tag a struct length tlv-tag b
```

0x0000000E 0x00000003 0x0011 0xAA 0x00000003 0x0022 0xBB

Please also refer to AP-6496 in the AP JIRA.

This bug is closely related to the TLV concept incorporation, so we should try to fix it for FO_R1.5.

Agreed solution:

Rework and split PRS_SOMEIP_00215 as follows:

+ [PRS_SOMEIP_xxxx1] The size of the length field for arrays, structs, unions and string shall be greater than 0 in the configuration. (RS_SOMEIP_00050)

Rationale: The TLV serialization requires the usage of length fields. When wire type 4 is used, the length field size must be statically configured. When wire types 5-7 (dynamic length field size) are used, the static configuration of the length field size must also be present since not all length fields are preceded by a tag, e.g. structs contained in an array or the top-level struct contained in a SOME/IP event. Not using length fields here would result in ambiguities.

+ [PRS_SOMEIP_xxxx2] The configured size of the length field for arrays, structs, unions and strings shall be identical. (RS_SOMEIP_00050)

Rationale: In case of an unknown member or argument, the deserializer cannot determine the actual datatype of the member/argument when wire type 4 is used.

+ [PRS_SOMEIP_xxxx3] The size of the length field shall be configured for the top-level struct or method request/response. All arrays, unions, structs and strings used within a struct or all arguments within a method shall inherit the size of the length field from the top-level definition. (RS_SOMEIP_00050)

Rationale: In case of an unknown member or argument, the deserializer needs to know the size of the length field when wire type 4 is used. The easiest way is that the size of the length field is then only defined at the top-level element.

+ [PRS_SOMEIP_xxxx4] Overriding the size of the length field at a subordinate array, union, struct or string or at an individual method argument shall not be allowed. (RS_SOMEIP_00050)

-[PRS_SOMEIP_00215]

–Last change on issue 80991 comment 31–

BW-C-Level:

Application	Specification	Bus
1	1	1

4.4 Specification Item PRS_SOMEIP_00242

Trace References:

RS_SOMEIP_00050

Content:

The configured size of the length field for arrays, structs, unions and strings shall be identical.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80991: Serialization of structures with identifier and optional members always requires static definition of struct length field size

Problem description:

Name: Daniel Kerk
Phone:
Role: OEM

Description/Motivation:

For the serialization of structures with identifier and optional members (TLV serialization), the SOME/IP Protocol Specification currently specifies that the size of length fields for arrays, structs and strings shall be identical shall be greater than zero when length fields of static size are used, i.e. usage of wire type 4 (see PRS_SOMEIP_00215).

This implies that if length fields of dynamic size are used, the size of the length fields do not need to be defined statically.

This is true when the length field is preceded by a tag. The size of the length field for arrays, strings or structs as members of a parent struct can then be encoded in the wire type.

But there are cases where a length field is not preceded by a tag:

- the top-level struct (e.g. contained in a SOME/IP event)
- an array of structs, strings or array

The current specification implies that in case of usage of length fields of dynamic size, in the above mentioned cases, no length fields are added.

This is critical because when optional members come into play, this might lead to ambiguities.

Example a)

```
struct s
uint8 a; // optional
uint8 b; // optional
```

`s[] = 0xAA, 0xBB` // one entry in array, both optional members exist

will result in the following serialized byte stream:

```
vector len tlv-tag a tlv-tag b
0x00000006 0x0011 0xAA 0x0022 0xBB
```

According to the current specification, the following example will result in the same serialized byte stream:

Example b)

`s[] = 0xAA, 0xBB` // two entries in array, only one optional member exists in each

Without length fields, the deserializer cannot determine the boundaries of the structs within the array.

So even if length fields of dynamic size are used, a static length field size must be configured to cover the length fields which are not preceded by a tag.

Using length fields here will result in unambiguous serializations:

a)
vector len struct length tlv-tag a tlv-tag b

0x0000000A 0x00000006 0x0011 0xAA 0x0022 0xBB

b)

vector len struct length tlv-tag a struct length tlv-tag b

0x0000000E 0x00000003 0x0011 0xAA 0x00000003 0x0022 0xBB

Please also refer to AP-6496 in the AP JIRA.

This bug is closely related to the TLV concept incorporation, so we should try to fix it for FO_R1.5.

Agreed solution:

Rework and split PRS_SOMEIP_00215 as follows:

+**[PRS_SOMEIP_xxxx1]** The size of the length field for arrays, structs, unions and string shall be greater than 0 in the configuration. (RS_SOMEIP_00050)

Rationale: The TLV serialization requires the usage of length fields. When wire type 4 is used, the length field size must be statically configured. When wire types 5-7 (dynamic length field size) are used, the static configuration of the length field size must also be present since not all length fields are preceded by a tag, e.g. structs contained in an array or the top-level struct contained in a SOME/IP event. Not using length fields here would result in ambiguities.

+**[PRS_SOMEIP_xxxx2]** The configured size of the length field for arrays, structs, unions and strings shall be identical. (RS_SOMEIP_00050)

Rationale: In case of an unknown member or argument, the deserializer cannot determine the actual datatype of the member/argument when wire type 4 is used.

+**[PRS_SOMEIP_xxxx3]** The size of the length field shall be configured for the top-level struct or method request/response. All arrays, unions, structs and strings used within a struct or all arguments within a method shall inherit the size of the length field from the top-level definition. (RS_SOMEIP_00050)

Rationale: In case of an unknown member or argument, the deserializer needs to know the size of the length field when wire type 4 is used. The easiest way is that the size of the length field is then only defined at the top-level element.

+**[PRS_SOMEIP_xxxx4]** Overriding the size of the length field at a subordinate array, union, struct or string or at an individual method argument shall not be

allowed. (RS_SOMEIP_00050)

-[PRS_SOMEIP_00215]

–Last change on issue 80991 comment 31–

BW-C-Level:

Application	Specification	Bus
1	1	1

4.5 Specification Item PRS_SOMEIP_00243

Trace References:

RS_SOMEIP_00050

Content:

The size of the length field shall be configured for the top-level struct or method request/response. All arrays, unions, structs and strings used within a struct or all arguments within a method shall inherit the size of the length field from the top-level definition.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80991: Serialization of structures with identifier and optional members always requires static definition of struct length field size

Problem description:

Name: Daniel Kerk

Phone:

Role: OEM

Description/Motivation:

For the serialization of structures with identifier and optional members (TLV serialization), the SOME/IP Protocol Specification currently specifies that the size of length fields for arrays, structs and strings shall be identical shall be greater than zero when length fields of static size are used, i.e. usage of wire type 4 (see PRS_SOMEIP_00215).

This implies that if length fields of dynamic size are used, the size of the length fields do not need to be defined statically.

This is true when the length field is preceded by a tag. The size of the length field for arrays, strings or structs as members of a parent struct can then be encoded in the wire type.

But there are cases where a length field is not preceded by a tag:

- the top-level struct (e.g. contained in a SOME/IP event)
- an array of structs, strings or array

The current specification implies that in case of usage of length fields of dynamic size, in the above mentioned cases, no length fields are added.

This is critical because when optional members come into play, this might lead to ambiguities.

Example a)

```
struct s
uint8 a; // optional
uint8 b; // optional
```

`s[] = 0xAA, 0xBB` // one entry in array, both optional members exist

will result in the following serialized byte stream:

```
vector len tlv-tag a tlv-tag b
0x00000006 0x0011 0xAA 0x0022 0xBB
```

According to the current specification, the following example will result in the same serialized byte stream:

Example b)

`s[] = 0xAA, 0xBB` // two entries in array, only one optional member exists in each

Without length fields, the deserializer cannot determine the boundaries of the structs within the array.

So even if length fields of dynamic size are used, a static length field size must be configured to cover the length fields which are not preceded by a tag.

Using length fields here will result in unambitious serializations:

a)

vector len struct length tlv-tag a tlv-tag b

0x0000000A 0x00000006 0x0011 0xAA 0x0022 0xBB

b)

vector len struct length tlv-tag a struct length tlv-tag b

0x0000000E 0x00000003 0x0011 0xAA 0x00000003 0x0022 0xBB

Please also refer to AP-6496 in the AP JIRA.

This bug is closely related to the TLV concept incorporation, so we should try to fix it for FO_R1.5.

Agreed solution:

Rework and split PRS_SOMEIP_00215 as follows:

+ [PRS_SOMEIP_xxxx1] The size of the length field for arrays, structs, unions and string shall be greater than 0 in the configuration. (RS_SOMEIP_00050)

Rationale: The TLV serialization requires the usage of length fields. When wire type 4 is used, the length field size must be statically configured. When wire types 5-7 (dynamic length field size) are used, the static configuration of the length field size must also be present since not all length fields are preceded by a tag, e.g. structs contained in an array or the top-level struct contained in a SOME/IP event. Not using length fields here would result in ambiguities.

+ [PRS_SOMEIP_xxxx2] The configured size of the length field for arrays, structs, unions and strings shall be identical. (RS_SOMEIP_00050)

Rationale: In case of an unknown member or argument, the deserializer cannot determine the actual datatype of the member/argument when wire type 4 is used.

+ [PRS_SOMEIP_xxxx3] The size of the length field shall be configured for the top-level struct or method request/response. All arrays, unions, structs and strings used within a struct or all arguments within a method shall inherit the size of the length field from the top-level definition. (RS_SOMEIP_00050)

Rationale: In case of an unknown member or argument, the deserializer needs to know the size of the length field when wire type 4 is used. The easiest way is that

the size of the length field is then only defined at the top-level element.

+[\[PRS_SOMEIP_xxxx4\]](#) Overriding the size of the length field at a subordinate array, union, struct or string or at an individual method argument shall not be allowed. (RS_SOMEIP_00050)

-[\[PRS_SOMEIP_00215\]](#)

—Last change on issue 80991 comment 31—

BW-C-Level:

Application	Specification	Bus
1	1	1

4.6 Specification Item [PRS_SOMEIP_00244](#)

Trace References:

[RS_SOMEIP_00050](#)

Content:

Overriding the size of the length field at a subordinate array, union, struct or string or at an individual method argument shall not be allowed.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80991: Serialization of structures with identifier and optional members always requires static definition of struct length field size

Problem description:

Name: Daniel Kerk

Phone:

Role: OEM

Description/Motivation:

For the serialization of structures with identifier and optional members (TLV serialization), the SOME/IP Protocol Specification currently specifies that the size of length fields for arrays, structs and strings shall be identical shall be greater than zero when length fields of static size are used, i.e. usage of wire type 4 (see [PRS_SOMEIP_00215](#)).

This implies that if length fields of dynamic size are used, the size of the length fields do not need to be defined statically.

This is true when the length field is preceded by a tag. The size of the length field for arrays, strings or structs as members of a parent struct can then be encoded in the wire type.

But there are cases where a length field is not preceded by a tag:

- the top-level struct (e.g. contained in a SOME/IP event)
- an array of structs, strings or array

The current specification implies that in case of usage of length fields of dynamic size, in the above mentioned cases, no length fields are added.

This is critical because when optional members come into play, this might lead to ambiguities.

Example a)

```
struct s
uint8 a; // optional
uint8 b; // optional
```

`s[] = 0xAA, 0xBB // one entry in array, both optional members exist`

will result in the following serialized byte stream:

```
vector len tlv-tag a tlv-tag b
0x00000006 0x0011 0xAA 0x0022 0xBB
```

According to the current specification, the following example will result in the same serialized byte stream:

Example b)

`s[] = 0xAA, 0xBB // two entries in array, only one optional member exists in each`

Without length fields, the deserializer cannot determine the boundaries of the structs within the array.

So even if length fields of dynamic size are used, a static length field size must be configured to cover the length fields which are not preceded by a tag.

Using length fields here will result in unambitious serializations:

a)

vector len struct length tlv-tag a tlv-tag b
0x0000000A 0x00000006 0x0011 0xAA 0x0022 0xBB

b)

vector len struct length tlv-tag a struct length tlv-tag b
0x0000000E 0x00000003 0x0011 0xAA 0x00000003 0x0022 0xBB

Please also refer to AP-6496 in the AP JIRA.

This bug is closely related to the TLV concept incorporation, so we should try to fix it for FO_R1.5.

Agreed solution:

Rework and split PRS_SOMEIP_00215 as follows:

+ [PRS_SOMEIP_xxxx1] The size of the length field for arrays, structs, unions and string shall be greater than 0 in the configuration. (RS_SOMEIP_00050)

Rationale: The TLV serialization requires the usage of length fields. When wire type 4 is used, the length field size must be statically configured. When wire types 5-7 (dynamic length field size) are used, the static configuration of the length field size must also be present since not all length fields are preceded by a tag, e.g. structs contained in an array or the top-level struct contained in a SOME/IP event. Not using length fields here would result in ambiguities.

+ [PRS_SOMEIP_xxxx2] The configured size of the length field for arrays, structs, unions and strings shall be identical. (RS_SOMEIP_00050)

Rationale: In case of an unknown member or argument, the deserializer cannot determine the actual datatype of the member/argument when wire type 4 is used.

+ [PRS_SOMEIP_xxxx3] The size of the length field shall be configured for the top-level struct or method request/response. All arrays, unions, structs and strings used within a struct or all arguments within a method shall inherit the size of the length field from the top-level definition. (RS_SOMEIP_00050)

Rationale: In case of an unknown member or argument, the deserializer needs to know the size of the length field when wire type 4 is used. The easiest way is that the size of the length field is then only defined at the top-level element.

+ [PRS_SOMEIP_xxxx4] Overriding the size of the length field at a subordinate array, union, struct or string or at an individual method argument shall not be allowed. (RS_SOMEIP_00050)

- [PRS_SOMEIP_00215]

–Last change on issue 80991 comment 31–

BW-C-Level:

Application	Specification	Bus
1	1	1

4.7 Specification Item PRS_SOMEIP_00925

Trace References:

RS_SOMEIP_00004

Content:

For the SOME/IP notification message the **client** **server** has to do the following for payload and header:

- Construct the payload
- Set the Message ID based on the event the server wants to send
- Set the Length field to 8 bytes (for the part of the SOME/IP header after the length field) + length of the serialized payload
- Set the Client ID to 0x00. Set the Session ID according to PRS_SOMEIP_00932, PRS_SOMEIP_00933, and PRS_SOMEIP_00521. In case of active Session Handling the Session ID shall be incremented upon each transmission.
- Set the Protocol Version according PRS_SOMEIP_00052
- Set the Interface Version according to the interface definition
- Set the Message Type to NOTIFICATION (i.e. 0x02)
- Set the Return Code to 0x00

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80340: One statement in the document 'PRS SOMEIPProtocol' requires further discussion

Problem description:

During the document review one statement to be clarified was identified:

Section 4 Protocol Specification, [PRS_SOMEIP_00925], Error: For the SOME/IP notification message the client has to do the following for payload and header.

It seems server would make more sense here.

A comprehensive feedback would be appreciated in order to go ahead with the document release.

Agreed solution:

[PRS_SOMEIP_00925] d For the SOME/IP notification message the server has to do

the following for payload and header:

–Last change on issue 80340 comment 2–

BW-C-Level:

Application	Specification	Bus
1	4	4

4.8 Specification Item PRS_SOMEIP_00941**Trace References:**

RS_SOMEIP_00027

Content:

In case of E2E communication protection being applied, the E2E header is placed after Return Code, depending on the chosen Offset value for the E2E header. The default Offset value is 64 bit, which puts the E2E header exactly between Return Code and Payload as shown in the Figure [REF fig_3a_SOME_2d_IP_2d_Header_E2E_2d_Implementation].

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #81921: Unclear which part of the SOME/IP Header is considered for E2E protection

Problem description:

Currently only in the Classic Platform it is clear that the payload and a specific part of the SOME/IP Header is used for the CRC calculation of the E2E protection. Which part of the SOME/IP header is used for the CRC is not explicitly specified but only implicitly by the architecture of the CP BSW.

For the Adaptive Platform it is currently not specified at all. (See AP-7187) But AP needs to also specify that to ensure compatibility of E2E protected communication between CP and AP, and even between AP instances of different vendors.

In AP-7187 it was decided that it would be best to add the specification which part of the SOME/IP Header is considered for E2E CRC calculation into the PRS SOME/IP.

For AP, the SWS CM should then refer to this specification.

Agreed solution:

In PRS_E2EProtocol, add a new section in chapter 9:

`\section{E2E and SOME/IP}`

For the combination of E2E communication protection with SOME/IP, there needs to be a common definition of the on-wire protocol. Depending on architecture properties, the implementing components need to be configured and used accordingly.

In general, all available E2E profiles can be used in combination with SOME/IP. However, they may have limitations, as for the maximum usable length of data, or being limited to fixed length messages.

The size of the E2E Header is dependent on the selected E2E profile.

`\begin{ARTrace}{PRS_E2E_USE_00236RS_E2E_08540}`

The E2E CRC shall be calculated over the following parts of the serialized SOME/IP message.

`\begin{enumerate}`

`\item Request ID (Client ID / Session ID) [32 bit]`

`\item Protocol Version [8 bit]`

`\item Interface Version [8 bit]`

`\item Message Type [8 bit]`

`\item Return Code [8 bit]`

`\item Payload [variable size]`

`\end{enumerate}`

\}\endARTrace

\}\beginARTracePRS_E2E_USE_00237RS_E2E_08540

The E2E header shall be placed after the Return Code depending on the chosen Offset value. The default Offset is 64 bit, which puts the E2E header exactly after the Return Code.

\}\endARTrace

In PRS_SOMEIIPProtocol, add below PRS_SOMEIP_00030 / figure fig:image002:

\}\beginARTracePRS_SOMEIP_00941RS_SOMEIP_00027

In case of E2E communication protection being applied, the E2E header is placed after Return Code, depending on the chosen Offset value for the E2E header. The default Offset value is 64 bit, which puts the E2E header exactly between Return Code and Payload as shown in the Figure~\}\refig:SOME-IP-Header_E2E-Implementation.

\}\endARTrace

\}\ARFigurechapters/99_Figures/SOME-IP-Header_E2E-

Implementation\}\textwidthfig:SOME-IP-Header_E2E-ImplementationSOME/IP
Header and E2E header Format

–Last change on issue 81921 comment 40–

BW-C-Level:

Application	Specification	Bus
1	4	4

5 PRS_SOMEIPServiceDiscoveryProtocol

5.1 Specification Item PRS_SOMEIPSD_00124

Trace References:

RS_SOMEIPSD_00019

Content:

For error handling of incoming SOME/IP-SD messages, Execute the checks described in (PRS_SOMEIPSD_00125, PRS_SOMEIPSD_00126, PRS_SOMEIPSD_00127, PRS_SOMEIPSD_00128, PRS_SOMEIPSD_00129, PRS_SOMEIPSD_00803,

PRS_SOMEIPSD_00130, PRS_SOMEIPSD_00131, PRS_SOMEIPSD_00132). If at least one of these checks fails, you need to:

- Answer with a Subscribe Eventgroup NACK, if the original entry was a Subscribe Eventgroup entry
- Ignore, if the original entry was not a Subscribe Eventgroup entry

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

* if number of opt1 equals 0, the Index 1st options also equals 0

* if number of opt2 equals 0, the Index 2nd options also equals 0
—Last change on issue 79206 comment 18—

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

- + * Discardable Flag [1 bit]: Specifies if the option can be discarded.
- + * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1.

+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

* The referenced options exist.

* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).

* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).

- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
- * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
- * The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup

entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

5.2 Specification Item PRS_SOMEIPSD_00125

Trace References:

RS_SOMEIPSD_00019

Content:

Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e. the message is at least 12 Bytes long. **If the check fails, the message shall be discarded without further actions.**

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupAck for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

- * if number of opt1 equals 0, the Index 1st options also equals 0

- * if number of opt2 equals 0, the Index 2nd options also equals 0

—Last change on issue 79206 comment 18—

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

- + * Discardable Flag [1 bit]: Specifies if the option can be discarded.

- + * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 0.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK en-

try, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.

- * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).

- * The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

5.3 Specification Item PRS_SOMEIPSD_00126

Trace References:

RS_SOMEIPSD_00019

Content:

Check if If the Service ID is known of a received entry is not known, the entry shall be ignored.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

- * if number of opt1 equals 0, the Index 1st options also equals 0

- * if number of opt2 equals 0, the Index 2nd options also equals 0

–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

- + * Discardable Flag [1 bit]: Specifies if the option can be discarded.

- + * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 0.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 0.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 0.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 0.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1.

- + * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.

- * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).

- * The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

5.4 Specification Item PRS_SOMEIPSD_00127

Trace References:

RS_SOMEIPSD_00019

Content:

Check if If the Instance ID of **this Service ID is known** a **received entry is not known**, the **entry shall be ignored**.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

* if number of opt1 equals 0, the Index 1st options also equals 0

* if number of opt2 equals 0, the Index 2nd options also equals 0

—Last change on issue 79206 comment 18—

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.

+ * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving

ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e. the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.

- * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).

- * The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

5.5 Specification Item PRS_SOMEIPSD_00128

Trace References:

RS_SOMEIPSD_00019

Content:

Check if If the Major Version of this Service Instance is known a received entry is not known, the entry shall be ignored.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

* if number of opt1 equals 0, the Index 1st options also equals 0

* if number of opt2 equals 0, the Index 2nd options also equals 0
—Last change on issue 79206 comment 18—

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

- + * Discardable Flag [1 bit]: Specifies if the option can be discarded.
- + * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1.

+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e. the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

* The referenced options exist.

* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).

* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).

- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
- * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
- * The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup

entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

5.6 Specification Item PRS_SOMEIPSD_00129

Trace References:

RS_SOMEIPSD_00019

Content:

Check if If the Eventgroup ID of the Service Instance with Major Version is known (only applicable for eventgroup entries) a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupAck for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

- * if number of opt1 equals 0, the Index 1st options also equals 0

- * if number of opt2 equals 0, the Index 2nd options also equals 0

—Last change on issue 79206 comment 18—

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

- + * Discardable Flag [1 bit]: Specifies if the option can be discarded.

- + * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 0.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK en-

try, the entry shall be processed, but the subscription shall not be considered as successful.

+`[PRS_SOMEIPSD_00xx1]` Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.

- * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).

- * The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

5.7 Specification Item PRS_SOMEIPSD_00130

Trace References:

RS_SOMEIPSD_00019

Content:

Check **if** the referenced Options **exist in the options array and are syntactically ok** of each **received entry**:

- **Length of Options Array is consistent** The referenced options exist.
- **if number of opt1 equals 0, the Index 1st options also equals 0** The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- **if number of opt2 equals 0, the Index 2nd options also equals 0** The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- **Option Type is known** There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- **Option Length is consistent** The Type of the referenced Option is known or the discardable flag is set to 1.

- Option The Type of the referenced Option is valid for entry allowed for the entry PRS_SOMEIPSD_00583 or discardable flag is set to 1.
- Endpoint Options with The Length of the referenced Option is consistent to the Type of the Option.
- An Endpoint Option has a valid L4-Protocol field.
- The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array

and are syntactically ok:

bullet 2 and 3

* if number of opt1 equals 0, the Index 1st options also equals 0

* if number of opt2 equals 0, the Index 2nd options also equals 0

–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.

+ * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).

- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
- * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).

* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

5.8 Specification Item PRS_SOMEIPSD_00231

Trace References:

RS_SOMEIPSD_00019

Content:

Options that are referenced by an entry shall be ignored if:

- The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
- The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
- The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.
and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

- * if number of opt1 equals 0, the Index 1st options also equals 0

- * if number of opt2 equals 0, the Index 2nd options also equals 0

–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

- + * Discardable Flag [1 bit]: Specifies if the option can be discarded.

- + * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the

receiver.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1.

+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+`[PRS_SOMEIPSD_00xx3]` If the checks in `[PRS_SOMEIPSD_00130]` fail for a received Offer entry, the entry shall be ignored.

+`[PRS_SOMEIPSD_00xx4]` If the checks in `[PRS_SOMEIPSD_00130]`, `[PRS_SOMEIPSD_00131]`, or `[PRS_SOMEIPSD_00132]` fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+`[PRS_SOMEIPSD_00xx5]` If the checks in `[PRS_SOMEIPSD_00130]` or `[PRS_SOMEIPSD_00132]` fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+`[PRS_SOMEIPSD_00xx1]` Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.

- * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).

- * The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

—Last change on issue 79206 comment 75—

BW-C-Level:

Application	Specification	Bus
1	4	4

5.9 Specification Item PRS_SOMEIPSD_00232

Trace References:

[RS_SOMEIPSD_00019](#)

Content:

If the checks in `PRS_SOMEIPSD_00130` fail for a received Find entry, the entry shall be ignored.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

* if number of opt1 equals 0, the Index 1st options also equals 0

* if number of opt2 equals 0, the Index 2nd options also equals 0

—Last change on issue 79206 comment 18—

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.

+ * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving

ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e. the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.

- * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).

- * The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

5.10 Specification Item PRS_SOMEIPSD_00233

Trace References:

RS_SOMEIPSD_00019

Content:

If the checks in PRS_SOMEIPSD_00130 fail for a received Offer entry, the entry shall be ignored.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

* if number of opt1 equals 0, the Index 1st options also equals 0

* if number of opt2 equals 0, the Index 2nd options also equals 0
—Last change on issue 79206 comment 18—

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

- + * Discardable Flag [1 bit]: Specifies if the option can be discarded.
- + * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1.

+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

* The referenced options exist.

* The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).

* The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).

- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
- * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
- * The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup

entry, though it is still allowed).
–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

5.11 Specification Item PRS_SOMEIPSD_00234

Trace References:

RS_SOMEIPSD_00019

Content:

If the checks in PRS_SOMEIPSD_00130, PRS_SOMEIPSD_00131, or PRS_SOMEIPSD_00132 fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupAck for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

- * if number of opt1 equals 0, the Index 1st options also equals 0

- * if number of opt2 equals 0, the Index 2nd options also equals 0

—Last change on issue 79206 comment 18—

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

- + * Discardable Flag [1 bit]: Specifies if the option can be discarded.

- + * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 0.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK en-

try, the entry shall be processed, but the subscription shall not be considered as successful.

+`[PRS_SOMEIPSD_00xx1]` Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.

- * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).

- * The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

5.12 Specification Item PRS_SOMEIPSD_00235

Trace References:

[RS_SOMEIPSD_00019](#)

Content:

If the checks in [PRS_SOMEIPSD_00130](#) or [PRS_SOMEIPSD_00132](#) fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

[PRS_SOMEIPSD_00130](#) is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

[SWS_SD_00661](#) contradicts [PRS_SOMEIPSD_00130](#) bullet "Option Type is known"

[SWS_SD_00662](#) contradicts [PRS_SOMEIPSD_00130](#) bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

- * if number of opt1 equals 0, the Index 1st options also equals 0

- * if number of opt2 equals 0, the Index 2nd options also equals 0

–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

- + * Discardable Flag [1 bit]: Specifies if the option can be discarded.

- + * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be

discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for

a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:

* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.

* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).

* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

—Last change on issue 79206 comment 75—

BW-C-Level:

Application	Specification	Bus
1	4	4

5.13 Specification Item PRS_SOMEIPSD_00273

Trace References:

RS_SOMEIPSD_00006

Content:

In order to identify the option type every option shall start with:

- Length [uint16]: Specifies the length of the option in Bytes.
- Type [uint8]: Specifying the type of the option.
- Discardable Flag [1 bit]: Specifies if the option can be discarded.

- Bit 1 to bit 7 are reserved and shall be 0.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

* if number of opt1 equals 0, the Index 1st options also equals 0

* if number of opt2 equals 0, the Index 2nd options also equals 0

—Last change on issue 79206 comment 18—

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

- + * Discardable Flag [1 bit]: Specifies if the option can be discarded.
- + * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e. the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to

PRS_SOMEIPSD_00583) or discardable flag is set to 1.

- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
- * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
- * The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

5.14 Specification Item PRS_SOMEIPSD_00275

Trace References:

RS_SOMEIPSD_00006

Content:

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as descibed in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

- * if number of opt1 equals 0, the Index 1st options also equals 0

- * if number of opt2 equals 0, the Index 2nd options also equals 0

—Last change on issue 79206 comment 18—

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

- + * Discardable Flag [1 bit]: Specifies if the option can be discarded.

- + * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 0.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 0.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
 - * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
 - * The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).
- Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

5.15 Specification Item PRS_SOMEIPSD_00276

Trace References:

RS_SOMEIPSD_00006

Content:

The format of the Configuration Option shall be as follows:

- Length [uint16]: Shall be set to the total number of bytes occupied by the configuration option, excluding the 16 bit length field and the 8 bit type flag.
- Type [uint8]: Shall be set to 0x01.
- **Reserved Discardable Flag** [uint8¹ bit]: Shall be set to 0x00. 1 if the Option can be discarded by the receiver.
- **Bit 1 to bit 7 are reserved and shall be 0.**
- ConfigurationString [dyn length]: Shall carry the configuration string.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is

known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

- * if number of opt1 equals 0, the Index 1st options also equals 0

- * if number of opt2 equals 0, the Index 2nd options also equals 0

–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

- + * Discardable Flag [1 bit]: Specifies if the option can be discarded.

- + * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1.

+ * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:

* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.

* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).

* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

—Last change on issue 79206 comment 75—

BW-C-Level:

Application	Specification	Bus
1	4	4

5.16 Specification Item PRS_SOMEIPSD_00307

Trace References:

RS_SOMEIPSD_00006, RS_SOMEIPSD_00010

Content:

The Format of the IPv4 Endpoint Option shall be as follows:

- Length [uint16]: Shall be set to 0x0009.

- Type [uint8]: Shall be set to 0x04.
- **Reserved Discardable Flag** [uint81 bit]: Shall be set to 0x00. 0.
- **Bit 1 to bit 7 are reserved and shall be 0.**
- IPv4-Address [uint32]: Shall transport the unicast IP-Address as four Bytes.
- Reserved [uint8]: Shall be set to 0x00.
- Transport Protocol (L4-Proto) [uint8]: Shall be set to the transport layer protocol (ISO/OSI layer 4) based on the IANA/IETF types (0x06: TCP, 0x11: UDP).
- Transport Protocol Port Number (L4-Port) [uint16]: Shall be set to the port of the layer 4 protocol.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD mes-

sages with option run length set to zero and option index not set to zero.
and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array
and are syntactically ok:

bullet 2 and 3

* if number of opt1 equals 0, the Index 1st options also equals 0

* if number of opt2 equals 0, the Index 2nd options also equals 0

–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.

+ * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving
ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the
receiver.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the
receiver.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.

- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the

receiver) and the discardable flag is set to 1.

* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).

* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

5.17 Specification Item PRS_SOMEIPSD_00315

Trace References:

RS_SOMEIPSD_00006, RS_SOMEIPSD_00010

Content:

The Format of the IPv6 Endpoint Option shall be as follows:

- Length [uint16]: Shall be set to 0x0015.
- Type [uint8]: Shall be set to 0x06.
- **Reserved Discardable Flag** [uint81 bit]: Shall be set to **0x00. 0**.
- **Bit 1 to bit 7 are reserved and shall be 0**.
- IPv6-Address [uint128]: Shall transport the unicast IP-Address as 16 Bytes.
- Reserved [uint8]: Shall be set to 0x00.
- Transport Protocol (L4-Proto) [uint8]: Shall be set to the transport layer protocol (ISO/OSI layer 4) based on the IANA/IETF types (0x06: TCP, 0x11: UDP).
- Transport Protocol Port Number (L4-Port) [uint16]: Shall be set to the transport layer port(e.g. 30490).

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

* if number of opt1 equals 0, the Index 1st options also equals 0

* if number of opt2 equals 0, the Index 2nd options also equals 0

—Last change on issue 79206 comment 18—

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.

+ * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e. the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP

option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
- * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
- * The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

—Last change on issue 79206 comment 75—

BW-C-Level:

Application	Specification	Bus
1	4	4

5.18 Specification Item PRS_SOMEIPSD_00326

Trace References:

RS_SOMEIPSD_00003

Content:

The Format of the IPv4 Endpoint Option shall be as follows:

- Length [uint16]: Shall be set to 0x0009.
- Type [uint8]: Shall be set to 0x14.
- **Reserved Discardable Flag [uint81 bit]:** Shall be set to **0x00. 0.**
- **Bit 1 to bit 7 are reserved and shall be 0.**
- IPv4-Address [uint32]: Shall transport the multicast IP-Address as four Bytes.
- Reserved [uint8]: Shall be set to 0x00.
- Transport Protocol (L4-Proto) [uint8]: Shall be set to the transport layer protocol (ISO/OSI layer 4) based on the IANA/IETF types (0x11: UDP).
- Transport Protocol Port Number (L4-Port) [uint16]: Shall be set to the port of the layer 4 protocol.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) op-

tion type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

- * if number of opt1 equals 0, the Index 1st options also equals 0

- * if number of opt2 equals 0, the Index 2nd options also equals 0

–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

- + * Discardable Flag [1 bit]: Specifies if the option can be discarded.

- + * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 0.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not

known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as

successful.

+`[PRS_SOMEIPSD_00xx1]` Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.

- * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).

- * The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

5.19 Specification Item PRS_SOMEIPSD_00333

Trace References:

RS_SOMEIPSD_00003

Content:

The Format of the IPv6 Multicast Option shall be as follows:

- Length [uint16]: Shall be set to 0x0015.
- Type [uint8]: Shall be set to 0x16.
- **Reserved Discardable Flag** [uint81 bit]: Shall be set to 0x00. 0.
- **Bit 1 to bit 7 are reserved and shall be 0.**
- IPv6-Address [uint128]: Shall transport the multicast IP-Address as 16 Bytes.
- Reserved [uint8]: Shall be set to 0x00.
- Transport Protocol (L4-Proto) [uint8]: Shall be set to the transport layer protocol (ISO/OSI layer 4) based on the IANA/IETF types (0x11: UDP).
- Transport Protocol Port Number (L4-Port) [uint16]: Shall be set to the port of the layer 4 protocol.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

* if number of opt1 equals 0, the Index 1st options also equals 0

* if number of opt2 equals 0, the Index 2nd options also equals 0

—Last change on issue 79206 comment 18—

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.

+ * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~ [PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~ [PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~ [PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~ [PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~ [PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~ [PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~ [PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~ [PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e. the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:

* The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.

* The option is redundant (i.e. another option of the same type and same content is referenced by this entry).

* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

5.20 Specification Item PRS_SOMEIPSD_00544

Trace References:

RS_SOMEIPSD_00011

Content:

The Format of the Load Balancing Option shall be as follows:

- Length [uint16]: Shall be set to 0x0005.
- Type [uint8]: Shall be set to 0x02.
- **Reserved Discardable Flag [uint81 bit]:** Shall be set to **0x00. 1 if the Option can be discarded by the receiver.**
- **Bit 1 to bit 7 are reserved and shall be 0.**
- Priority [uint16]: Carries the Priority of this instance. Lower value means higher priority.
- Weight [uint16]: Carries the Weight of this instance. Large value means higher probability to be chosen.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so

that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

- * if number of opt1 equals 0, the Index 1st options also equals 0

- * if number of opt2 equals 0, the Index 2nd options also equals 0

–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

- + * Discardable Flag [1 bit]: Specifies if the option can be discarded.

- + * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 0.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not

known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.

- * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).

- * The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

5.21 Specification Item PRS_SOMEIPSD_00551

Trace References:

RS_SOMEIPSD_00006, RS_SOMEIPSD_00010

Content:

The IPv4 SD Endpoint Option shall specify the IPv4-Address, the transport layer protocol (ISO/OSI layer 4) **used, and a Port Number and Port Number of the sender used for Service Discovery.**

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80342: [PRS_SOMEIPSD] Redundant requirements for SD endpoint option

Problem description:

During the document review one statement to be clarified was identified:

Section 4 Protocol Specification, [PRS_SOMEIP_00710].

It seems to be a redundant copy of [PRS_SOMEIP_00653], thus it should be removed.

A comprehensive feedback would be appreciated in order to go ahead with the document release.

Agreed solution:

- [PRS_SOMEIPSD_00710]
- [PRS_SOMEIPSD_00653]

~[PRS_SOMEIPSD_00558]

- The IPv6 SD Endpoint Option shall specify the IPv6-Address, the transport layer protocol (ISO/OSI layer 4) used, and its Port Number.
- + The IPv6 SD Endpoint Option shall specify the IPv6-Address, the transport layer protocol (ISO/OSI layer 4) and Port Number of the sender used for Service Discovery.

~[PRS_SOMEIPSD_00551]

- The IPv4 SD Endpoint Option shall specify the IPv4-Address, the transport layer protocol (ISO/OSI layer 4) used, and a Port Number.
- + The IPv4 SD Endpoint Option shall specify the IPv4-Address, the transport layer protocol (ISO/OSI layer 4) and Port Number of the sender used for Service Discovery.

—Last change on issue 80342 comment 9—

BW-C-Level:

Application	Specification	Bus
1	1	1

5.22 Specification Item PRS_SOMEIPSD_00552

Trace References:

RS_SOMEIPSD_00006, RS_SOMEIPSD_00010

Content:

The Format of the IPv4 SD Endpoint Option shall be as follows:

- Length [uint16]: Shall be set to 0x0015.
- Type [uint8]: Shall be set to 0x24.
- **Reserved Discardable Flag** [uint81 bit]: Shall be set to **0x00. 0.**
- **Bit 1 to bit 7 are reserved and shall be 0.**

- IPv4-Address [uint32]: Shall transport the unicast IP-Address of SOME/IP-SD as four Bytes.
- Reserved [uint8]: Shall be set to 0x00.
- Transport Protocol (L4-Proto) [uint8]: Shall be set to the transport layer protocol of SOME/IP-SD (currently: 0x11 UDP).
- Transport Protocol Port Number (L4-Port) [uint16]: Shall be set to the transport layer port of SOME/IP-SD (currently: 30490).

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array

and are syntactically ok:

bullet 2 and 3

* if number of opt1 equals 0, the Index 1st options also equals 0

* if number of opt2 equals 0, the Index 2nd options also equals 0

–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.

+ * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).

- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
- * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).

* The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

—Last change on issue 79206 comment 75—

BW-C-Level:

Application	Specification	Bus
1	4	4

5.23 Specification Item PRS_SOMEIPSD_00558

Trace References:

RS_SOMEIPSD_00006, RS_SOMEIPSD_00010

Content:

The IPv6 SD Endpoint Option shall specify the IPv6-Address, the transport layer protocol (ISO/OSI layer 4) **used, and its Port Number and Port Number of the sender used for Service Discovery.**

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80342: [PRS_SOMEIPSD] Redundant requirements for SD endpoint option

Problem description:

During the document review one statement to be clarified was identified:

Section 4 Protocol Specification, [PRS_SOMEIP_00710].

It seems to be a redundant copy of [PRS_SOMEIP_00653], thus it should be removed.

A comprehensive feedback would be appreciated in order to go ahead with the document release.

Agreed solution:

- [PRS_SOMEIPSD_00710]
- [PRS_SOMEIPSD_00653]

~[PRS_SOMEIPSD_00558]

- The IPv6 SD Endpoint Option shall specify the IPv6-Address, the transport layer protocol (ISO/OSI layer 4) used, and its Port Number.

+ The IPv6 SD Endpoint Option shall specify the IPv6-Address, the transport layer protocol (ISO/OSI layer 4) and Port Number of the sender used for Service Discovery.

~[PRS_SOMEIPSD_00551]

- The IPv4 SD Endpoint Option shall specify the IPv4-Address, the transport layer protocol (ISO/OSI layer 4) used, and a Port Number.

+ The IPv4 SD Endpoint Option shall specify the IPv4-Address, the transport layer protocol (ISO/OSI layer 4) and Port Number of the sender used for Service Discovery.

–Last change on issue 80342 comment 9–

BW-C-Level:

Application	Specification	Bus
1	1	1

5.24 Specification Item PRS_SOMEIPSD_00559

Trace References:

RS_SOMEIPSD_00006, RS_SOMEIPSD_00010

Content:

The Format of the IPv6 SD Endpoint Option shall be as follows:

- Length [uint16]: Shall be set to 0x0015.
- Type [uint8]: Shall be set to 0x26.
- **Reserved Discardable Flag [uint81 bit]:** Shall be set to **0x00. 0**.
- **Bit 1 to bit 7 are reserved and shall be 0.**
- IPv6-Address [uint128]: Shall transport the unicast IP-Address of SOME/IP-SD as 16 Bytes.
- Reserved [uint8]: Shall be set to 0x00.
- Transport Protocol (L4-Proto) [uint8]: Shall be set to the transport layer protocol of SOME/IP-SD (currently: 0x11 UDP).
- Transport Protocol Port Number (L4-Port) [uint16]: Shall be set to the transport layer port of SOME/IP-SD (currently: 30490).

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

* if number of opt1 equals 0, the Index 1st options also equals 0

* if number of opt2 equals 0, the Index 2nd options also equals 0

—Last change on issue 79206 comment 18—

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

+ * Discardable Flag [1 bit]: Specifies if the option can be discarded.

+ * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~ [PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~ [PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~ [PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~ [PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~ [PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~ [PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 0.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~ [PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.

+ * Discardable Flag [1 bit]: Shall be set to 1.

+ * Bit 1 to bit 7 are reserved and shall be 0.

~ [PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e. the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.

- * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).

- * The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

5.25 Specification Item PRS_SOMEIPSD_00653

Trace References:

RS_SOMEIPSD_00006, RS_SOMEIPSD_00010

Content:

The IPv6 SD Endpoint Option shall only be included if the SOME/IP-SD message is transported over IPv6.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80342: [PRS_SOMEIPSD] Redundant requirements for SD endpoint option

Problem description:

During the document review one statement to be clarified was identified:

Section 4 Protocol Specification, [PRS_SOMEIP_00710].

It seems to be a redundant copy of [PRS_SOMEIP_00653], thus it should be removed.

A comprehensive feedback would be appreciated in order to go ahead with the document release.

Agreed solution:

- [PRS_SOMEIPSD_00710]
- [PRS_SOMEIPSD_00653]

~[PRS_SOMEIPSD_00558]

- The IPv6 SD Endpoint Option shall specify the IPv6-Address, the transport layer protocol (ISO/OSI layer 4) used, and its Port Number.
- + The IPv6 SD Endpoint Option shall specify the IPv6-Address, the transport layer protocol (ISO/OSI layer 4) and Port Number of the sender used for Service Discovery.

~[PRS_SOMEIPSD_00551]

- The IPv4 SD Endpoint Option shall specify the IPv4-Address, the transport layer protocol (ISO/OSI layer 4) used, and a Port Number.
- + The IPv4 SD Endpoint Option shall specify the IPv4-Address, the transport layer protocol (ISO/OSI layer 4) and Port Number of the sender used for Service Discovery.

—Last change on issue 80342 comment 9—

BW-C-Level:

Application	Specification	Bus
1	1	1

5.26 Specification Item PRS_SOMEIPSD_00710

Trace References:

RS_SOMEIPSD_00006, RS_SOMEIPSD_00010

Content:

The IPv6 SD Endpoint Option shall only be included if the SOME/IP-SD message is transported over IPv6.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80342: [PRS_SOMEIPSD] Redundant requirements for SD endpoint option

Problem description:

During the document review one statement to be clarified was identified:

Section 4 Protocol Specification, [PRS_SOMEIP_00710].

It seems to be a redundant copy of [PRS_SOMEIP_00653], thus it should be removed.

A comprehensive feedback would be appreciated in order to go ahead with the document release.

Agreed solution:

- [PRS_SOMEIPSD_00710]
- [PRS_SOMEIPSD_00653]

~[PRS_SOMEIPSD_00558]

- The IPv6 SD Endpoint Option shall specify the IPv6-Address, the transport layer protocol (ISO/OSI layer 4) used, and its Port Number.

+ The IPv6 SD Endpoint Option shall specify the IPv6-Address, the transport layer protocol (ISO/OSI layer 4) and Port Number of the sender used for Service Discovery.

~[PRS_SOMEIPSD_00551]

- The IPv4 SD Endpoint Option shall specify the IPv4-Address, the transport layer protocol (ISO/OSI layer 4) used, and a Port Number.
- + The IPv4 SD Endpoint Option shall specify the IPv4-Address, the transport layer protocol (ISO/OSI layer 4) and Port Number of the sender used for Service Discovery.
- Last change on issue 80342 comment 9–

BW-C-Level:

Application	Specification	Bus
1	1	1

5.27 Specification Item PRS_SOMEIPSD_00711

Trace References:

RS_SOMEIPSD_00011

Content:

When looking for all service instances of a service (Service Instance set to 0xFFFF), the client shall choose the service instance with highest priority **that also matches client specific criteria**.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79832: [PRS SD] Clarify load balancing option usage

Problem description:

The current SOME/IP Service Discovery Protocol Specification does not specify how to proceed if a service using load balancing options is already in use (i.e. ongoing method calls and subscribed event groups) and an offer service with higher priority is received.

How shall a service be selected if one service instance uses a load balancing option and an other instance does not.

Agreed solution:

+ [PRS_SOMEIPSD_00xxx]

When looking for a specific service instances of a service (Service Instance set to any value other than 0xFFFF), the evaluation of the Load Balancing Option does not apply.

~ [PRS_SOMEIPSD_00711]

- When looking for all service instances of a service (Service Instance set to 0xFFFF), the client shall choose the service instance with highest priority.

+ When looking for all service instances of a service (Service Instance set to 0xFFFF), the client shall choose the service instance with highest priority that also matches client specific criteria.

Note: Client specific criteria may be applied by the client application when choosing one of the offered service instances. They are not defined in this specification, and could e.g. restrict the range of appropriate instance IDs.

~[PRS_SOMEIPSD_00713]

- In case if there is no Load Balancing Option available and several Service instance are offered, then the application logic shall perform the selection of the Service instance based on use-case specific criteria.

+ If an Offer Service entry references no Load Balancing option and several service instances are offered, the client shall handle the service instances without Load Balancing option as though they had the lowest priority.

–Last change on issue 79832 comment 45–

BW-C-Level:

Application	Specification	Bus
4	4	4

5.28 Specification Item PRS_SOMEIPSD_00713

Trace References:

RS_SOMEIPSD_00011

Content:

In case if there is If an Offer Service entry references no Load Balancing Option available and several Service instance option and several service instances are offered, then the application logic shall perform the selection of the Service instance based on use-case specific criteria the client shall handle the service instances without Load Balancing option as though they had the lowest priority.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79832: [PRS SD] Clarify load balancing option usage

Problem description:

The current SOME/IP Service Discovery Protocol Specification does not specify how to proceed if a service using load balancing options is already in use (i.e.

ongoing method calls and subscribed event groups) and an offer service with higher priority is received.

How shall a service be selected if one service instance uses a load balancing option and an other instance does not.

Agreed solution:

+ [PRS_SOMEIPSD_00xxx]

When looking for a specific service instances of a service (Service Instance set to any value other than 0xFFFF), the evaluation of the Load Balancing Option does not apply.

~ [PRS_SOMEIPSD_00711]

- When looking for all service instances of a service (Service Instance set to 0xFFFF), the client shall choose the service instance with highest priority.

+ When looking for all service instances of a service (Service Instance set to 0xFFFF), the client shall choose the service instance with highest priority that also matches client specific criteria.

Note: Client specific criteria may be applied by the client application when choosing one of the offered service instances. They are not defined in this specification, and could e.g. restrict the range of appropriate instance IDs.

~ [PRS_SOMEIPSD_00713]

- In case if there is no Load Balancing Option available and several Service instance are offered, then the application logic shall perform the selection of the Service instance based on use-case specific criteria.

+ If an Offer Service entry references no Load Balancing option and several service instances are offered, the client shall handle the service instances without Load Balancing option as though they had the lowest priority.

–Last change on issue 79832 comment 45–

BW-C-Level:

Application	Specification	Bus
4	4	4

5.29 Specification Item PRS_SOMEIPSD_00714

Trace References:

RS_SOMEIPSD_00011

Content:

When looking for a specific service instances of a service (Service Instance set to any value other than 0xFFFF), the evaluation of the Load Balancing Option does not apply.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79832: [PRS SD] Clarify load balancing option usage

Problem description:

The current SOME/IP Service Discovery Protocol Specification does not specify how to proceed if a service using load balancing options is already in use (i.e. ongoing method calls and subscribed event groups) and an offer service with higher priority is received.

How shall a service be selected if one service instance uses a load balancing option and an other instance does not.

Agreed solution:

+ [PRS_SOMEIPSD_00xxx]

When looking for a specific service instances of a service (Service Instance set to any value other than 0xFFFF), the evaluation of the Load Balancing Option does not apply.

~ [PRS_SOMEIPSD_00711]

- When looking for all service instances of a service (Service Instance set to 0xFFFF), the client shall choose the service instance with highest priority.

+ When looking for all service instances of a service (Service Instance set to 0xFFFF), the client shall choose the service instance with highest priority that also matches client specific criteria.

Note: Client specific criteria may be applied by the client application when choosing one of the offered service instances. They are not defined in this specification, and could e.g. restrict the range of appropriate instance IDs.

~ [PRS_SOMEIPSD_00713]

- In case if there is no Load Balancing Option available and several Service instance are offered, then the application logic shall perform the selection of the Service instance based on use-case specific criteria.

+ If an Offer Service entry references no Load Balancing option and several service instances are offered, the client shall handle the service instances without Load Balancing option as though they had the lowest priority.

—Last change on issue 79832 comment 45—

BW-C-Level:

Application	Specification	Bus
4	4	4

5.30 Specification Item PRS_SOMEIPSD_00803

Trace References:

RS_SOMEIPSD_00019

Content:

Check that If the length of the Entries Array has a valid size an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #79206: [PRS SD] Contradicting and incomplete requirements for Handling missing, redundant, and conflicting Options

Problem description:

PRS_SOMEIPSD_00130 is ambiguous/contradicting/incomplete in comparison to SWS "7.3.9.8 Handling missing, redundant, and conflicting Options"

SWS_SD_00661 contradicts PRS_SOMEIPSD_00130 bullet "Option Type is known"

SWS_SD_00662 contradicts PRS_SOMEIPSD_00130 bullet "Option is valid for entry"

SWS_SD_00662 is ambiguous in the meaning of redundant vs. valid/invalid as described in PRS.

If an option is not needed (redundant), I would consider it invalid as described in PRS_SOMEIPSD_00130. I would consider options that are not allowed as specified in PRS_SOMEIPSD_00583 as invalid. (E.g. a multicast endpoint option referenced from an offer is not allowed and invalid.)

Can anyone give an exhaustive list of options that are redundant and still valid, so that we can specify this in the PRS?

I propose to consider every entry that references a known (i.e. specified) option type, that is not allowed (refer to PRS_SOMEIPSD_00583) as invalid and ignore the entry or send a SubscribeEventgroupNack for SubscribeEventgroup.

Contradicting requirements:

[SWS_SD_00624] Implementations shall accept and process incoming SD messages with option run length set to zero and option index not set to zero.

and

PRS_SOMEIPSD_00130] Check if the referenced Options exist in the options array and are syntactically ok:

bullet 2 and 3

- * if number of opt1 equals 0, the Index 1st options also equals 0

- * if number of opt2 equals 0, the Index 2nd options also equals 0

–Last change on issue 79206 comment 18–

Agreed solution:

~[PRS_SOMEIPSD_00273]

add after Type (last bullet point)

- + * Discardable Flag [1 bit]: Specifies if the option can be discarded.

- + * Bit 1 to bit 7 are reserved and shall be 0.

+ [PRS_SOMEIPSD_00xxx] add after [PRS_SOMEIPSD_00274]

The discardable flag shall be set to 1 if the option can be discarded by a receiving ECU that does not support this option.

~[PRS_SOMEIPSD_00276]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00544]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 1 if the Option can be discarded by the receiver.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00307]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 0.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00315]

- * Reserved [uint8]: Shall be set to 0x00.

- + * Discardable Flag [1 bit]: Shall be set to 0.

- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00326]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00333]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 0.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00552]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

~[PRS_SOMEIPSD_00559]

- * Reserved [uint8]: Shall be set to 0x00.
- + * Discardable Flag [1 bit]: Shall be set to 1.
- + * Bit 1 to bit 7 are reserved and shall be 0.

-[PRS_SOMEIPSD_00124]

~[PRS_SOMEIPSD_00125] Check that at least enough bytes for an empty SOME/IP-SD message are present, i.e the message is at least 12 Bytes long. If the check fails, the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00803] If the length of the Entries Array has an invalid size (i.e. the entries array would exceed the message size), the message shall be discarded without further actions.

~[PRS_SOMEIPSD_00126] If the Service ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00127] If the Instance ID of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00128] If the Major Version of a received entry is not known, the entry shall be ignored.

~[PRS_SOMEIPSD_00129] If the Eventgroup ID of a received entry is not known, the entry shall be ignored. This is only applicable to eventgroup entries.

~[PRS_SOMEIPSD_00130] Check the referenced Options of each received

entry:

- * The referenced options exist.
- * The entry references all required options (e.g. a provided eventgroup that uses unicast requires a unicast endpoint option in a received Subscribe Eventgroup entry).
- * The entry only references supported options (e.g. a required eventgroup that does not support multicast data reception does not support multicast endpoint options in a Subscribe Eventgroup ACK entry).
- * There are no conflicts between the options referenced by an entry (i.e. two options of same type with contradicting content).
- * The Type of the referenced Option is known or the discardable flag is set to 1.
- * The Type of the referenced Option is allowed for the entry (refer to PRS_SOMEIPSD_00583) or discardable flag is set to 1.
- * The Length of the referenced Option is consistent to the Type of the Option.
- * An Endpoint Option has a valid L4-Protocol field.
- * The Option is valid (e.g. a multicast endpoint option shall use a multicast IP address).

Add after [PRS_SOMEIPSD_00130]:

Note: If an entry references an option that is known by the Service Discovery implementation but not required by the service (e.g. an Offer references a TCP and UDP option and the client uses only UDP, or a Subscribe Eventgroup entry references a UDP endpoint option but the server uses only multicast event transmission), the entry shall be processed.

Add the following new requiremenst after [PRS_SOMEIPSD_00132]:

+ [PRS_SOMEIPSD_00xx2] If the checks in [PRS_SOMEIPSD_00130] fail for a received Find entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx3] If the checks in [PRS_SOMEIPSD_00130] fail for a received Offer entry, the entry shall be ignored.

+ [PRS_SOMEIPSD_00xx4] If the checks in [PRS_SOMEIPSD_00130], [PRS_SOMEIPSD_00131], or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup entry, a Subscribe Eventgroup NACK entry shall be sent.

+ [PRS_SOMEIPSD_00xx5] If the checks in [PRS_SOMEIPSD_00130] or [PRS_SOMEIPSD_00132] fail for a received Subscribe Eventgroup ACK entry, the entry shall be processed, but the subscription shall not be considered as successful.

+ [PRS_SOMEIPSD_00xx1] Options that are referenced by an entry shall be

ignored if:

- * The Option Type is not known (i.e. not yet specified, or not supported by the receiver) and the discardable flag is set to 1.
- * The option is redundant (i.e. another option of the same type and same content is referenced by this entry).
- * The option is not required (e.g. a provided eventgroup that uses only multicast does not require a unicast endpoint option in a received Subscribe Eventgroup entry, though it is still allowed).

–Last change on issue 79206 comment 75–

BW-C-Level:

Application	Specification	Bus
1	4	4

5.31 Specification Item PRS_SOMEIPSD_00822

Trace References:

RS_SOMEIPSD_00015

Content:

If the server receives a Subscribe Eventgroup entry with the **Initial Data Requested Flag set to 0 and the** Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send **no** notifications/events (i.e. initial events)**immediately after sending the Subscribe Eventgroup Ack.**

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80067: [PRS SD] Contradicting and incomplete requirements for initial data request flag on server side

Problem description:

PRS_SOMEIPSD_00270, PRS_SOMEIPSD_00386, PRS_SOMEIPSD_00703 and PRS_SOMEIPSD_00463 specify when the initial data request flag shall be set to 1.

The implied semantic of the initial data request flag contradicts the specified reaction on the server as specified in PRS_SOMEIPSD_00822.

According to PRS_SOMEIPSD_00822 the initial data shall be sent independent of the request flag, making the request flag useless.

–Last change on issue 80067 comment 1–

Agreed solution:

~PRS_SOMEIPSD_00822

- If the server receives a Subscribe Eventgroup entry with the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send notifications/events (i.e. initial events) immediately after sending the Subscribe Eventgroup Ack.

+ If the server receives a Subscribe Eventgroup entry with the Initial Data Requested Flag set to 0 and the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send no notifications/events (i.e. initial events).

add PRS_SOMEIPSD_00xxx after PRS_SOMEIPSD_00822

+ If the server receives a Subscribe Eventgroup entry with the Initial Data Requested Flag set to 1 and the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send notifications/events (i.e. initial events) immediately after sending the Subscribe Eventgroup Ack.

-PRS_SOMEIPSD_00823

—Last change on issue 80067 comment 25—

BW-C-Level:

Application	Specification	Bus
1	1	1

5.32 Specification Item PRS_SOMEIPSD_00823

Trace References:

RS_SOMEIPSD_00015

Content:

The client shall repeat the Subscribe Eventgroup entry, if it did not receive the notifications/events in a configurable timeout independent of the setting of the Explicit Initial Data Control Flag.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80067: [PRS SD] Contradicting and incomplete requirements for initial data request flag on server side

Problem description:

PRS_SOMEIPSD_00270, PRS_SOMEIPSD_00386, PRS_SOMEIPSD_00703 and PRS_SOMEIPSD_00463 specify when the initial data request flag shall be set to 1.

The implied semantic of the initial data request flag contradicts the specified reaction on the server as specified in PRS_SOMEIPSD_00822.

According to PRS_SOMEIPSD_00822 the initial data shall be sent independent of the request flag, making the request flag useless.

–Last change on issue 80067 comment 1–

Agreed solution:

~PRS_SOMEIPSD_00822

- If the server receives a Subscribe Eventgroup entry with the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send notifications/events (i.e. initial events) immediately after sending the Subscribe Eventgroup Ack.

+ If the server receives a Subscribe Eventgroup entry with the Initial Data Requested Flag set to 0 and the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send no notifications/events (i.e. initial events).

add PRS_SOMEIPSD_00xxx after PRS_SOMEIPSD_00822

+ If the server receives a Subscribe Eventgroup entry with the Initial Data Requested Flag set to 1 and the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send notifications/events (i.e. initial events) immediately after sending the Subscribe Eventgroup Ack.

-PRS_SOMEIPSD_00823

–Last change on issue 80067 comment 25–

BW-C-Level:

Application	Specification	Bus
1	1	1

5.33 Specification Item PRS_SOMEIPSD_00824

Trace References:

[RS_SOMEIPSD_00015](#)

Content:

If the server receives a Subscribe Eventgroup entry with the Initial Data Requested Flag set to 1 and the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send notifications/events (i.e. initial events) immediately after sending the Subscribe Eventgroup Ack.

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80067: [PRS SD] Contradicting and incomplete requirements for initial data request flag on server side

Problem description:

PRS_SOMEIPSD_00270, PRS_SOMEIPSD_00386, PRS_SOMEIPSD_00703 and PRS_SOMEIPSD_00463 specify when the initial data request flag shall be set to 1.

The implied semantic of the initial data request flag contradicts the specified reaction on the server as specified in PRS_SOMEIPSD_00822.

According to PRS_SOMEIPSD_00822 the initial data shall be sent independent of the request flag, making the request flag useless.

–Last change on issue 80067 comment 1–

Agreed solution:

~PRS_SOMEIPSD_00822

- If the server receives a Subscribe Eventgroup entry with the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send notifications/events (i.e. initial events) immediately after sending the Subscribe Eventgroup Ack.

+ If the server receives a Subscribe Eventgroup entry with the Initial Data Requested Flag set to 0 and the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send no notifications/events (i.e. initial events).

add PRS_SOMEIPSD_00xxx after PRS_SOMEIPSD_00822

+ If the server receives a Subscribe Eventgroup entry with the Initial Data Requested Flag set to 1 and the Explicit Initial Data Control Flag (in the SOME/IP-SD header) set to 1, the server shall send notifications/events (i.e. initial events) immediately after sending the Subscribe Eventgroup Ack.

-PRS_SOMEIPSD_00823

–Last change on issue 80067 comment 25–

BW-C-Level:

Application	Specification	Bus
1	1	1

6 PRS_TimeSyncProtocol

7 RS_E2E

8 RS_FoundationDebugTraceProfile

9 RS_HealthMonitoring

10 RS_LogAndTrace

10.1 Specification Item RS_LT_00001

Trace References:

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	Valid
Description:	The LT module shall be a BSW module. It shall receive log and trace messages from several sources, like Applications, some BSW modules and the RTE. These messages may be transmitted over a communication interface to an external client.
Rationale:	Each Tier1 uses its own mechanisms to provide such a logging interface, using some internal or external debugging interfaces. The format of the logging content also differs from ECU to ECU. When testing several ECU's, many different tools and parsers are needed to get the right information out of the logs. A standard Diagnostic Logging Component with standardized logging content may help to reduce the testing efforts and enable new automated testing mechanisms. Also the number of tools could be reduced by a standard logging content and protocol.
AppliesTo:	FO
Use Case:	
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.2 Specification Item RS_LT_00002

Trace References:

RS_MAIN_00260, RS_MAIN_00420

Content:

Type:	Valid
Description:	A specified format shall be defined, which covers all requirements of log and trace. Some examples of the information stored in a log or trace message are the source, the context and the timestamp of the message to be able to filter the log and trace messages.
Rationale:	Since logging and tracing is an important mechanism for testability and proofing product quality, it is necessary to standardize the transmitted and stored data format. This is important for archiving, comparing and analyzing of log or trace messages. Also it may be possible to build common tools to interpret the incoming data.
AppliesTo:	FO
Use Case:	
Supporting Material:	see [REF] https://svn2.autosar.org/repos2/09_Work_Packages/13_WPII-1.1.3/01_Deliverables/AUTOSAR_Concept_DLT.ppt

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.3 Specification Item RS_LT_00003

Trace References:

RS_BRF_02144MAIN_00260, RS_BRF_01440 MAIN_00420, RS_MAIN_00260,
RS_MAIN_00011

Content:

Type:	Valid
Description:	LT shall offer a generic interface for Applications independent from the type of log and trace message.
Rationale:	To reduce the amount of interfaces a generic and message independent interface may be implemented.
AppliesTo:	FO
Use Case:	
Supporting Material:	see [REF]

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.4 Specification Item RS_LT_00004

Trace References:

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	valid
Description:	Application shall be aware of its log levels and trace status in order not to generate unnecessary log and trace messages. Log level and trace status should be checked by the Applications before sending the messages.
Rationale:	The generation of unnecessary load on the communication interface and for the system has to be avoided. The LT shall provide information about the actual configuration.
AppliesTo:	FO
Use Case:	
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.5 Specification Item RS_LT_00005

Trace References:

RS_BRF_02200MAIN_00260, RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	valid
Description:	The generation of the LT interface of each Application shall be enabled by the configuration.
Rationale:	Only the configured interfaces to the Applications, which are using log and trace, shall be generated.
AppliesTo:	FO
Use Case:	
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.6 Specification Item RS_LT_00006

Trace References:

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	Valid
Description:	The DET receives trace events from errors generated by BSW and Applications during debugging time. These events may be forwarded to the LT module.
Rationale:	To have an overview of all log, trace and error messages and to set all of them in the correct context, it is important to have all these messages and events in one list. Also it is not practicable to use more than one mechanism to report errors, logs and traces to an external client. So all these sources may be forwarded to the LT module.
AppliesTo:	CP
Use Case:	
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.7 Specification Item RS_LT_00007

Trace References:

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	Valid
Description:	The DEM may forward error events to the LT module.
Rationale:	It may be possible to get an overview of all error messages in an ECU. It may be possible to set them in the correct context with the error events reported by DEM. This makes an analysis of the reported errors more efficient and gives a correct picture of a failure situation.
AppliesTo:	CP
Use Case:	
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.8 Specification Item RS_LT_00008

Trace References:

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	Valid
Description:	RTE shall provide the possibility to trace the VFB.





Rationale:	In the future more and more Applications will be integrated in one ECU. As a consequence the communication between Application is done locally and not over an external traceable bus like CAN or Flexray. It is important to trace the internal communication over RTE/VFB.
AppliesTo:	CP
Use Case:	
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.9 Specification Item RS_LT_00009

Trace References:

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	valid
Description:	VFB-Tracing shall be provided to the RTE generation system. The LT module shall implement the handling of the RTE/VFB trace events.
Rationale:	To understand the communication between the Applications it is important to trace all function calls to the Applications except the RTE system function calls.





AppliesTo:	CP
Use Case:	–
Supporting Material:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.10 Specification Item RS_LT_00010

Trace References:

RS_BRF_01440 MAIN_00260, RS_BRF_02160 MAIN_00011, RS_MAIN_00260

Content:

Type:	valid
Description:	There shall be a global switch to turn on and off the RTE tracing of LT at configuration time and at run time.
Rationale:	To have a high level activation for all RTE tracing messages a global switch is needed.
AppliesTo:	CP
Use Case:	



**Supporting
Material:**

—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.11 Specification Item RS_LT_00011**Trace References:**

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	Valid
Description:	The LT may implement the handling of the RTE/VFB trace events.
Rationale:	To understand the communication between the Applications it is important to trace the function calls of an Application.
AppliesTo:	CP
Use Case:	—
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.12 Specification Item RS_LT_00012

Trace References:

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	Valid
Description:	LT may provide a solution to trace events linked to implicit communication mechanism. The implicit communication may be traced at runnable invocation and termination.
Rationale:	From the point of view of an Application no extra routine for receiving or sending signals (Sender Receiver Communication) with implicit sending/receiving is called.
AppliesTo:	
Use Case:	
Supporting Material:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.13 Specification Item RS_LT_00013

Trace References:

RS_BRF_01440 MAIN_00260, RS_BRF_01240 MAIN_00011, RS_MAIN_00100, RS_MAIN_00010

Content:

Type:	Valid
Description:	The log and trace messages shall be packet oriented. Each packet shall have one or several headers and a payload.
Rationale:	LT shall not care about the under laying transport protocol. LT shall define the log or trace message format.
AppliesTo:	FO
Use Case:	—
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.14 Specification Item RS_LT_00014

Trace References:

RS_BRF_01440MAIN_00260, RS_BRF_01240 MAIN_00011, RS_MAIN_00100,
RS_MAIN_00010

Content:

Type:	Valid
Description:	The log and trace messages shall be in binary format. This is necessary for reducing the bandwidth and the CPU consumption.
Rationale:	Scope of the LT component is to log and trace events without perturbing the ECU behavior. CPU consumption and bandwidth occupation may be reduced by using a binary format for the protocol.
Use Case:	—
Applies to:	FO
Dependencies:	—
Conflicts	None
Supporting Material	—

Type:	Valid
Description:	The log and trace messages shall be in binary format. This is necessary for reducing the bandwidth and the CPU consumption.
Rationale:	Scope of the LT component is to log and trace events without perturbing the ECU behavior. CPU consumption and bandwidth occupation may be reduced by using a binary format for the protocol.
AppliesTo:	FO
Use Case:	—
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.15 Specification Item RS_LT_00016

Trace References:

RS_BRF_01440MAIN_00260, RS_BRF_01624 MAIN_00011, RS_MAIN_00430

Content:

Type:	Valid
Description:	The Endian format shall be considered in the message specification. The message shall signal which Endianness mode is actually used.
Rationale:	Big and Little Endianness of data representation has mixed occurrences in automotive ECUs. The message format shall be able to deal with both. To reduce CPU load it shall be possible to transport the data in the corresponding format of the ECU.
AppliesTo:	FO
Use Case:	
Supporting Material:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.16 Specification Item RS_LT_00017

Trace References:

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	Valid
Description:	Each log and trace message may contain a timestamp, which shall be added to the message during reception of the message in the LT module.
Rationale:	For comparing and analyzing complex systems a timestamp is important. The time shall be in a defined granularity.
AppliesTo:	FO
Use Case:	—
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.17 Specification Item RS_LT_00018

Trace References:

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	Valid
Description:	Every time a message from an Application is received by the LT module or other software module, it shall increment its global messages counter and set the counter in the message.
Rationale:	It is important to know if a message is lost. A receiving external client may then detect if messages are lost (e.g.because of a buffer overflow in LT or network loss).
AppliesTo:	FO
Use Case:	—
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.18 Specification Item RS_LT_00019

Trace References:

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	Valid
Description:	Log levels shall be for example error, fatal, debug, information or verbose.
Rationale:	Standard for logging tools
AppliesTo:	FO
Use Case:	–
Supporting Material:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.19 Specification Item RS_LT_00020

Trace References:

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	Valid
Description:	The log and trace message may contain a parameter, which represents the source of the log and trace message.
Rationale:	The source can be identified by the Session ID of the port interface (port defined argument value) which was used to send the message from Application to LT.





AppliesTo:	FO
Use Case:	—
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.20 Specification Item RS_LT_00021**Trace References:**

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	Valid
Description:	There shall be additional parameters describing the group of the log and trace message.
Rationale:	Identifying some logically groups of log and trace to define some filtering action on the external client.
AppliesTo:	FO
Use Case:	—



**Supporting
Material:**

—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.21 Specification Item RS_LT_00022**Trace References:**

RS_BRF_01024 MAIN_00500, RS_MAIN_00220

Content:

Type:	Valid
Description:	Each ECU may have its unique ECU ID, which may be a parameter of each log and trace message.
Rationale:	An external client receives tracing from different ECU's so it's important to know the source.
AppliesTo:	FO
Use Case:	
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.22 Specification Item RS_LT_00023

Trace References:

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	valid
Description:	In the payload the parameters of each log and trace message from the Application shall be transported. It may be possible to transport more than one parameter.
Rationale:	–
AppliesTo:	FO
Use Case:	–
Supporting Material:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.23 Specification Item RS_LT_00024

Trace References:

RS_MAIN_00260, RS_MAIN_00011

Content:

Type:	Valid
Description:	In most cases the parameter shall be transmitted in the raw format. This means that a separate file shall be provided containing the information about the transmitted parameters.
Rationale:	The non-verbose mode is for saving memory and bandwidth, because on the ECU only the ID's of the log and trace messages shall be stored.
AppliesTo:	FO
Use Case:	
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.24 Specification Item RS_LT_00025

Trace References:

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	Valid
Description:	Because log and trace messages often consists of texts, there shall be the possibility to transmit text strings. It may be possible to replace the given text by an ID which may be replaced by the text in the external client (Compare verbose - non-verbose mode).
Rationale:	In small ECU strings can not be stored because of their memory usage. For this reasons these strings may be replaced by a message ID.
AppliesTo:	FO
Use Case:	—
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.25 Specification Item RS_LT_00026**Trace References:**

RS_MAIN_00260, RS_MAIN_00011

Content:

Type:	valid
Description:	The data in non-verbose mode need to be described by an extra file. This file may be in a XML format.
Rationale:	In small ECUs memory is very limited, so temporarily buffered data shall be very small.
AppliesTo:	FO
Use Case:	–
Supporting Material:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.26 Specification Item RS_LT_00027

Trace References:

RS_BRF_01024 MAIN_00500, RS_MAIN_00220

Content:

Type:	Valid
Description:	Each message may have a unique identifier significant for identifying the source of the tracing. The ID of a specific message shall be permanent over the developing phase and shall be unique for a given ECU.





Rationale:	Easy identifying the right information in the log and trace message is very important. Unique message IDs helps doing this.
AppliesTo:	FO
Use Case:	–
Supporting Material:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.27 Specification Item RS_LT_00028

Trace References:

RS_BRF_02160 MAIN_00260

Content:

Type:	valid
Description:	A control message shall be implemented to permit the external client to evaluate the round trip time. The external client shall use this message to synchronize the individual log and trace messages from the different ECUs.
Rationale:	The log and trace time information may enable a user to understand what happens if there is a correlation among the recorded events. It is important to be able to refer all the events logged to an absolute time scale.





AppliesTo:	FO
Use Case:	
Supporting Material:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.28 Specification Item RS_LT_00029**Trace References:**

RS_BRF_02008MAIN_00170, RS_BRF_02208 MAIN_00260, RS_MAIN_00170

Content:

Type:	valid
Description:	In production phase, the communication module (see RS_LT_00034) shall be disabled by default. In production phase LT shall use the security mechanisms provided by DCM to handle the access to the log and trace messages. A mechanism shall be implemented to enable the communication module for testing phase permanently.
Rationale:	Only authorized accesses to LT shall be allowed during production phase to avoid misuse.





AppliesTo:	FO
Use Case:	
Supporting Material:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.29 Specification Item RS_LT_00030

Trace References:

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	Valid
Description:	LT may shape the traffic bandwidth, depending on which transport interface is used for communication. The traffic shaping shall be done at runtime.
Rationale:	Since LT shall be used also in production phase, it is important that the traffic produced by the LT component will not cause an overload of the network and the ECU resources.
AppliesTo:	FO
Use Case:	



**Supporting
Material:**

—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.30 Specification Item RS_LT_00031**Trace References:**

RS_BRF_02200 MAIN_00260

Content:

Type:	Valid
Description:	LT shall be configurable at runtime. LT shall have for this propose runtime configurable variables which are initialized with the corresponding configuration parameters. It shall be possible to query the LT runtime configuration. The configuration variables shall be writeable and readable over with some control .mechanism triggered remotely.
Rationale:	To adapt LT behavior to ECU and network resources it is important to have the capability of configuring it.
AppliesTo:	FO
Use Case:	



**Supporting
Material:**

—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.31 Specification Item RS_LT_00032**Trace References:**

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	valid
Description:	LT shall support control messages to set the trace status and log level of each source of each ECU.
Rationale:	—
AppliesTo:	FO
Use Case:	
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.32 Specification Item RS_LT_00033

Trace References:

RS_BRF_02168 MAIN_00260, RS_MAIN_00011

Content:

Type:	Valid
Description:	It shall be possible to get a list of all Application IDs and context IDs with corresponding trace status and log levels which are registered to the LT module. Each Application or BSW module which wants to produce trace or log messages shall register to the LT module.
Rationale:	Users need to know what information can be traced or logged in order to monitor ECU activity.
AppliesTo:	FO
Use Case:	
Supporting Material:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.33 Specification Item RS_LT_00034

Trace References:

RS_BRF_01440MAIN_00260, RS_BRF_01056 MAIN_00011, RS_MAIN_00220

Content:

Type:	valid
Description:	At debugging phase a communication interface with a high bandwidth is needed. This communication interface is out of scope of the AUTOSAR specification. To use this communication interface with LT a generic API shall be defined. The user shall be able to implement a board specific communication module to communicate over a board specific communication interface.
Rationale:	—
AppliesTo:	FO
Use Case:	—
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.34 Specification Item RS_LT_00035

Trace References:

RS_BRF_02144 MAIN_00260, RS_BRF_01440 MAIN_00420, RS_BRF_01056
MAIN_00260, RS_MAIN_00011, RS_MAIN_00220

Content:

Type:	Valid
Description:	DCM shall provide an interface for LT to send and receive data over the diagnostic service. Log and trace messages are sent over this service and control messages for LT are received.
Rationale:	LT needs an interface to send log and trace message out of the ECU. DCM provides a bus independent access to the ECU over standardized diagnostics. This is available during production phase and provides a secured session control.
AppliesTo:	CP
Use Case:	
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.35 Specification Item RS_LT_00036

Trace References:

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	Valid
Description:	The LT may provide a buffer for storing log and trace messages. This may be a standard C-initialized buffer which shall be read out after init of LT.
Rationale:	This requirement shall guarantee that the LT module can provide its services to other BSW modules also before the complete initialization of all BSW modules.
AppliesTo:	FO
Use Case:	
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.36 Specification Item RS_LT_00037

Trace References:

RS_BRF_02184 MAIN_00260

Content:

Type:	Valid
Description:	If the communication module is not enabled or no external client for dumping log and trace message is connected, log and trace messages shall be stored locally in a memory buffer. The size of the buffer shall be set at configuration time.





Rationale:	To prevent a lack of communication e.g. at startup or if no external client is connected, the LT shall store some messages to be dumped as soon as communication channel is available.
AppliesTo:	FO
Use Case:	
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.37 Specification Item RS_LT_00038

Trace References:

RS_BRF_02184 MAIN_00260, RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	valid
Description:	LT shall permit to turn off or on traces or to set log levels for registered Application IDs and context IDs at runtime.
Rationale:	To gain efficiency in bandwidth and ECU resource usage a mechanism shall be provided to tune the transmission from Applications to LT.





AppliesTo:	FO
Use Case:	
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.38 Specification Item RS_LT_00039**Trace References:**

RS_BRF_01440MAIN_00260, RS_BRF_01824 MAIN_00011, RS_MAIN_00440

Content:

Type:	Valid
Description:	Some of the log Log and trace configurations like specific log levels Trace configuration shall be stored persistent.
Rationale:	The LT can be configured at runtime. Some of this L&T configuration data has to be stored to guarantee that at every startup the configuration would be the same.
AppliesTo:	FO
Use Case:	—





**Supporting
Material:**

–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.39 Specification Item RS_LT_00040

Trace References:

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	valid
Description:	In order to adapt network traffic to bandwidth availability, the LT component may be able to filter log and trace messages. A policy shall be defined to distinguish messages by different log and trace levels.
Rationale:	–
AppliesTo:	FO
Use Case:	–
Supporting Material:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.40 Specification Item RS_LT_00041

Trace References:

RS_BRF_01440 MAIN_00260, RS_MAIN_00011

Content:

Type:	valid
Description:	Only one SW module per ECU shall gather the log and trace messages centrally.
Rationale:	None
AppliesTo:	FO
Use Case:	None
Supporting Material:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
–Last change on issue 80283 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

10.41 Specification Item RS_LT_00042

Trace References:

RS_BRF_01016 MAIN_00330

Content:

Type:	valid
Description:	The Log and trace SW component may shall be also part of the system during production phase. It is needed for use cases like test drives and remote diagnosis.
Rationale:	None
AppliesTo:	FO
Use Case:	
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.42 Specification Item RS_LT_00056

Trace References:

RS_MAIN_00260, RS_MAIN_00011

Content:

Type:	valid
Description:	The log and trace messages shall contain a description of the parameters in verbose mode. No additional description file needs to be provided in this case.
Rationale:	In the verbose mode all parameters of the payload contain a self-description of their own type (e.g. integer/float, bit length) and some additional information (e.g. variable name, unit, etc).
AppliesTo:	FO
Use Case:	
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80283: [RS_LogAndTrace] Remove references to RS_BSF_* items

Problem description:

The RS_BSF_* documents are marked as deprecated and our requirements should not trace back to these documents but instead should trace back to RS_Main.

Agreed solution:

Proposed solution, please refer to <https://bugzilla.autosar.org/attachment.cgi?id=4792>
—Last change on issue 80283 comment 3—

BW-C-Level:

Application	Specification	Bus
1	1	1

10.43 Specification Item RS_LT_00057

Trace References:

RS_Main_00400

Content:

Type:	Valid
Description:	It shall be possible to trace the communication between applications.
Rationale:	More and more applications are integrated on one ECU. As a consequence the communication between applications is done locally and not over an external traceable bus like CAN.
AppliesTo:	FO
Use Case:	Trace the communication between applications.
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80277: [DLT] Log and Trace communication flow traceability requirement

Problem description:

In order to debug or analyze the communication between Adaptive Applications, the system integrator could enable the tracing of a given Adaptive Application during compile time (if the tracing feature is enabled/supported by the Adaptive Application in question).

Agreed solution:

Add a new requirement after chapter 5.2.1.1.3 [RS_LT_00049] Providing Logging Information:

5.2.1.1.4 [RS_LT_00057] Providing Tracing Information:

Type: Valid

Description: It shall be possible to trace the communication between applications.

Rationale: More and more applications are integrated on one ECU. As a consequence the communication between applications is done locally and not over an external traceable bus like CAN

Use Case: Trace the communication between applications

Applies to: AP, CP

—Last change on issue 80277 comment 12—

BW-C-Level:

Application	Specification	Bus
1	1	1

11 RS_Main

11.1 Specification Item RS_Main_00010

Trace References:

RS_PO_00005

Content:

Type:	Valid
Description:	AUTOSAR Platforms shall support commonly used safety mechanisms. This includes but is not limited to:
Rationale:	Facilitate the development of safety related systems by using AUTOSAR platforms. Platforms designed for the support of safety related systems are needed for safety related ECUs like digital engine control units and electronic power steering systems.
AppliesTo:	AP, CP
Use Case:	Driver temporarily/partially passes responsibility for driving task to vehicle (ADAS/HAD)
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #81369: RS Main - requirements header format not aligned

Problem description:

The header format of the requirements in the RS_Main document is not aligned. Some of them end with a dot and some not. This should be changed to either all with or all without dot.

Example:

[RS_Main_00050] AUTOSAR shall provide an Execution Framework towards applications to implement concurrent application internal control flows.

RS_Main_00106] AUTOSAR shall provide the possibility to extend the software with new SWCs without recompiling the platform foundation

Agreed solution:

Solution: Remove all dots at the end of RS_Main requirements
—Last change on issue 81369 comment 1—

BW-C-Level:

Application	Specification	Bus
1	1	1

11.2 Specification Item RS_Main_00012

Trace References:

RS_PO_00005, RS_PO_00009

Content:

Type:	Valid
Description:	Availability of the Adaptive Platform is defined as the probability that platform services will operate satisfactorily at a given point in time. It excludes both preventive and corrective maintenance downtime. Availability is limited to normal runtime behavior (excluding e.g. software update)
Rationale:	Facilitate the development of highly available systems by using AUTOSAR platforms.
AppliesTo:	AP, CP
Use Case:	Highly available systems are required for automated driving applications.
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #81369: RS Main - requirements header format not aligned

Problem description:

The header format of the requirements in the RS_Main document is not aligned. Some of them end with a dot and some not. This should be changed to either all with or all without dot.

Example:

[RS_Main_00050] AUTOSAR shall provide an Execution Framework towards applications to implement concurrent application internal control flows.

RS_Main_00106] AUTOSAR shall provide the possibility to extend the software with new SWCs without recompiling the platform foundation

Agreed solution:

Solution: Remove all dots at the end of RS_Main requirements
–Last change on issue 81369 comment 1–

BW-C-Level:

Application	Specification	Bus
1	1	1

11.3 Specification Item RS_Main_00050

Trace References:

RS_PO_00006, RS_PO_00004

Content:

Type:	Valid
Description:	AUTOSAR shall provide an Execution Framework towards applications to implement concurrent application internal control flows.
Rationale:	The execution framework must manage numerous running SWCs and handle their independent control flows.
AppliesTo:	AP
Use Case:	The execution framework starts several SWCs in an ordered manner.
Supporting Material:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #81369: RS Main - requirements header format not aligned

Problem description:

The header format of the requirements in the RS_Main document is not aligned. Some of them end with a dot and some not. This should be changed to either all with or all without dot.

Example:

[RS_Main_00050] AUTOSAR shall provide an Execution Framework towards applications to implement concurrent application internal control flows.

RS_Main_00106] AUTOSAR shall provide the possibility to extend the software with new SWCs without recompiling the platform foundation

Agreed solution:

Solution: Remove all dots at the end of RS_Main requirements
–Last change on issue 81369 comment 1–

BW-C-Level:

Application	Specification	Bus
1	1	1

11.4 Specification Item RS_Main_00120

Trace References:

RS_PO_00008

Content:

Type:	Valid
Description:	AUTOSAR shall provide specified test cases and the essential test methodology to ensure interoperability on application (RTE side) and bus level for BSW on ICC1 level (Black Box Test). These specified test cases and its related methodology shall be developed to test implementations of AUTOSAR basic software.
Rationale:	Acceptance tests are strongly needed to provide evidence that a product complies with the AUTOSAR specification i.e. to ensure a certain behavior of the regarded elements at the interfaces to application and communication busses.
Use Case:	Integration of the infrastructure SW into a specific ECU, bring it into the E/E-architecture without backlashes on the system. Example from real world: Integration of BSW stack (ICC1 level) to applications and the ECU infrastructure without difficulties. Support test of any ICC implementations (from ICC1 to ICC3). Reuse of the same test specification even when the ICC3 specification details change
Applies to:	TC
Dependencies:	–
Supporting Material:	–
Tested Items:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #81369: RS Main - requirements header format not aligned

Problem description:

The header format of the requirements in the RS_Main document is not aligned. Some of them end with a dot and some not. This should be changed to either all

with or all without dot.

Example:

[RS_Main_00050] AUTOSAR shall provide an Execution Framework towards applications to implement concurrent application internal control flows.

RS_Main_00106] AUTOSAR shall provide the possibility to extend the software with new SWCs without recompiling the platform foundation

Agreed solution:

Solution: Remove all dots at the end of RS_Main requirements
–Last change on issue 81369 comment 1–

BW-C-Level:

Application	Specification	Bus
1	1	1

11.5 Specification Item RS_Main_00125

Trace References:

RS_PO_00007, RS_PO_00008

Content:

Type:	Valid
Description:	Acceptance tests shall provide a reference to measure maturity.
Rationale:	An existing test specification provides verification for requirements that are available with the AUTOSAR software standard. A common set of test cases as a reference enables the verification in the software implementation.
Use Case:	Reuse of standard tests during the qualification process of BSW implementation.
Applies to:	TC
Dependencies:	–
Supporting Material:	–
Tested Items:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #81369: RS Main - requirements header format not aligned

Problem description:

The header format of the requirements in the RS_Main document is not aligned. Some of them end with a dot and some not. This should be changed to either all with or all without dot.

Example:

[RS_Main_00050] AUTOSAR shall provide an Execution Framework towards applications to implement concurrent application internal control flows.

RS_Main_00106] AUTOSAR shall provide the possibility to extend the software with new SWCs without recompiling the platform foundation

Agreed solution:

Solution: Remove all dots at the end of RS_Main requirements
–Last change on issue 81369 comment 1–

BW-C-Level:

Application	Specification	Bus
1	1	1

11.6 Specification Item RS_Main_00160

Trace References:

RS_PO_00007

Content:

Type:	Valid
Description:	Well-defined interfaces are the key for exchangeability, reusability and the basis for the freedom of interference.
Rationale:	Principle: "divide and conquer" which is a key success factor in the development of large systems.
AppliesTo:	AP, CP
Use Case:	Development of large interconnected software systems with a high degree of reuse, such as driving assistance systems.
Supporting Material:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #81369: RS Main - requirements header format not aligned

Problem description:

The header format of the requirements in the RS_Main document is not aligned. Some of them end with a dot and some not. This should be changed to either all with or all without dot.

Example:

[RS_Main_00050] AUTOSAR shall provide an Execution Framework towards applications to implement concurrent application internal control flows.

RS_Main_00106] AUTOSAR shall provide the possibility to extend the software with new SWCs without recompiling the platform foundation

Agreed solution:

Solution: Remove all dots at the end of RS_Main requirements

–Last change on issue 81369 comment 1–

BW-C-Level:

Application	Specification	Bus
1	1	1

11.7 Specification Item RS_Main_00161

Trace References:

RS_PO_00007

Content:

Type:	Valid
Description:	A comprehensive software model shall capture all elements of a distributed software system. The definition of functionality shall be independent from the final assignment to platforms and ECUs.
Rationale:	The allocation of software to platforms shall be a subsequent step after the overall definition of functionality.
AppliesTo:	AP, CP, FO
Use Case:	Development of large software systems being deployed on mixed platforms.





**Supporting
Material:**

–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #81369: RS Main - requirements header format not aligned

Problem description:

The header format of the requirements in the RS_Main document is not aligned. Some of them end with a dot and some not. This should be changed to either all with or all without dot.

Example:

[RS_Main_00050] AUTOSAR shall provide an Execution Framework towards applications to implement concurrent application internal control flows.

RS_Main_00106] AUTOSAR shall provide the possibility to extend the software with new SWCs without recompiling the platform foundation

Agreed solution:

Solution: Remove all dots at the end of RS_Main requirements
–Last change on issue 81369 comment 1–

BW-C-Level:

Application	Specification	Bus
1	1	1

11.8 Specification Item RS_Main_00170

Trace References:

RS_PO_00004, RS_PO_00005, RS_PO_00009

Content:

Type:	Valid
Description:	AUTOSAR shall provide secured access to ECU , (e.g. data and services by user secure authentication), including standardized up- and download of data and software external ECU users. For this mechanisms and methods shall access control decisions need to be defined/enforced.
Rationale:	The update Secure access and upgrade feasibility provided by AUTOSAR includes technical challenges (e.g. standardized up-/download protocol, partly update of the software) and authentication mechanisms (eare required for prevention of unauthorized access.g. how to authorize the user).
AppliesTo:	AP, CP
Use Case:	Download of dedicated Software Components in ECU.Secure V2X connection
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80615: Umbrella RfC for RS_Main working list R18-10 / FO1.5.0

Problem description:

This RfC is used to collect the RfCs currently discussed in the PL working list:

https://svn.autosar.org/repos/work/26_Standards/00_FO_R1/01_Sources/Elaboration/RS_Main/F

RfCs are discussed in the working list to achieve an agreement. Duplicate RfCs or resolved RfCs targeting single changes will be linked to this RfC.

For the release all agreed changes from the working list are updated to this RfC and implemented.

Agreed solution:

[RS_Main_00170] AUTOSAR shall provide secure access to ECU

Description: AUTOSAR shall provide secured access to ECU by secure authentication of external ECU users. For this mechanisms access control decisions need to be enforced.

Rationale: Secure access and authentication mechanisms are required for prevention of unauthorized access.

Use Case: Secure V2X connection

Applies to: AP, CP

Dependencies: To fulfill this requirement it is also necessary that the environment that is not standardized by AUTOSAR (e.g. bootloader) matches the same security requirements.

Supporting Material:—

(RS_PO_00004, RS_PO_00005, RS_PO_00009)

[new requirement] AUTOSAR shall support up - and download of data and software

Type: valid

Description: AUTOSAR shall support standardized up- and download of data and software. For all kind of data exchange between off- and onboard artifacts mechanisms and methods shall be defined. These mechanisms and methods shall support common protocols used for data-transfer. Partial updates of the software shall be supported. Independent access control rules and policies apply.

Rationale: Up- and download of data and software is required for software updates using standardized mechanisms.

Use Case: Download of dedicated Software Components in ECU.

Applies to: AP

Dependencies: –

Supporting Material: –

(RS_PO_00004, RS_PO_00005, RS_PO_00009)

remove [RS_Main_00220]

[RS_Main_00460] AUTOSAR shall standardize methods to organize mode management on Application, ECU and System level

Description: AUTOSAR shall provide a method to configure mode management mechanisms for Application Software to control or react on modes of the ECU or vehicle.

Rationale: The behavior of Application Software highly depends on the overall mode of the ECU. Therefore the method of mode management has to be standardized to achieve the same behavior if Application Software is allocated on another ECU.

[RS_Main_00503] AUTOSAR shall support change of communication and application software at runtime.

Description: Advanced systems require dynamic allocation of AUTOSAR Applications and adaptations of the communication topology after development and production at life-time of the system AUTOSAR shall provide a technical possibility which provides these Software changes at runtime.

Rationale: Advanced driving assistance functions have to be updated (e.g. after development or production).

Use-Case: Update of Application Software or update of configuration over the air

Applies to: AP

Dependencies: –

–Last change on issue 80615 comment 8–

BW-C-Level:

Application	Specification	Bus
1	1	1

- RfC #82029: [CONC_636] [MS4] [FO] Concept incorporation

Problem description:

This RfC is derived from AUTOSAR concept:

CONC_636 "Security Extensions" [1] [2]

It is intended to track the incorporation of the concept's technical solution into the AUTOSAR CP, FO and AP standard as draft since there is no external validation in 2018.

For each impacted document, a copy& paste ready solution is provided by the concept and a respective implementation task is created. In case of questions, please contact the concept owner:

Eduard Metzker (eduard.metzker@vector.com)

For SWS_AdaptiveDiagnostics a Jira Ticket will be created.

For CP a separate Bugzilla Ticket will be created

- C1 - "Security Event Memory (SEM)" is released as draft.
- C2 - "Key Management" is released as valid.
- C4 - "STBM with Secure Time" is released as draft.
- C5 - "Diagnostic Policy Manager" is released as valid.
- C6 - "Improvement of Certificate Handling" is released as valid.
- C7 - "Abstract pre-definition of Crypto Items in System Template" is released as valid.

Best regards, QA

[1] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_633_FormalModelQuery/Wo

[2] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_633_FormalModelQuery/Wo

Agreed solution:

Implement the CONC_636 in your document according to [1] an [2].

[1] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_636_SecurityExtensions/Wo

[2] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_636_SecurityExtensions/Wo

- C1 - "Security Event Memory (SEM)" is released as draft.
- C2 - "Key Management" is released as valid.
- C4 - "STBM with Secure Time" is released as draft.
- C5 - "Diagnostic Policy Manager" is released as valid.
- C6 - "Improvement of Certificate Handling" is released as valid.
- C7 - "Abstract pre-definition of Crypto Items in System Template" is released as valid.

=====

RS_Main

5.1 Main requirements

TR_Glossary:

7.3.1 C1: Security Event Memory

SRS_Diagnostic:

7.3.4 C5: Diagnostic Policy Manager

In case of questions, please contact the concept owner:
Eduard Metzker (eduard.metzker@vector.com)

Example to mark requirements as "draft":

1) For RS and SRS: Please set the field "Type" to "draft"

2) For SWS: Please use the annotation: [SWS_<shortname>_<XXXXX>] DRAFT
<Text> ((S)RS_ID)

3) For EXP/TR: Please add a sentence to chapter "Limitations" and list the used concept parts and provide a sentence that these concept parts are "draft"

If possible a bit more elaborated description of what is draft would be great. Just some additional words or sentences what the concept contains and what this means for your document. Maybe there even is some specific chapter where things coming from the concept are bundled to which you could refer in the limitations chapter, stating that everything that is described there is implemented as draft in SWS and RS documents.

—Last change on issue 82029 comment 5—

BW-C-Level:

Application	Specification	Bus
-	-	-

11.9 Specification Item RS_Main_00220

Trace References:

RS_PO_00001, RS_PO_00004

Content:

Type:	Valid
Description:	The specification of functional interfaces of AUTOSAR shall be specified in standard C90.
Rationale:	A useful reduction of programming languages to current programming languages reduces the impacts on AUTOSAR definitions and specifications due to logical and/or technical differences of different programming languages.
AppliesTo:	CP, FO
Use Case:	AUTOSAR implementation in C, C++.
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80615: Umbrella RfC for RS_Main working list R18-10 / FO1.5.0

Problem description:

This RfC is used to collect the RfCs currently discussed in the PL working list:

https://svn.autosar.org/repos/work/26_Standards/00_FO_R1/01_Sources/Elaboration/RS_Main/F

RfCs are discussed in the working list to achieve an agreement. Duplicate RfCs or resolved RfCs targeting single changes will be linked to this RfC.

For the release all agreed changes from the working list are updated to this RfC and implemented.

Agreed solution:

[RS_Main_00170] AUTOSAR shall provide secure access to ECU

Description: AUTOSAR shall provide secured access to ECU by secure authentication of external ECU users. For this mechanisms access control decisions need to be enforced.

Rationale: Secure access and authentication mechanisms are required for prevention of unauthorized access.

Use Case: Secure V2X connection

Applies to: AP, CP

Dependencies: To fulfill this requirement it is also necessary that the environment that is not standardized by AUTOSAR (e.g. bootloader) matches the same security requirements.

Supporting Material:–

(RS_PO_00004, RS_PO_00005, RS_PO_00009)

[new requirement] AUTOSAR shall support up - and download of data and software

Type: valid

Description: AUTOSAR shall support standardized up- and download of data and software. For all kind of data exchange between off- and onboard artifacts mechanisms and methods shall be defined. These mechanisms and methods shall support common protocols used for data-transfer. Partial updates of the software shall be supported. Independent access control rules and policies apply.

Rationale: Up- and download of data and software is required for software updates using standardized mechanisms.

Use Case: Download of dedicated Software Components in ECU.

Applies to: AP

Dependencies: –

Supporting Material: –

(RS_PO_00004, RS_PO_00005, RS_PO_00009)

remove [RS_Main_00220]

[RS_Main_00460] AUTOSAR shall standardize methods to organize mode management on Application, ECU and System level

Description: AUTOSAR shall provide a method to configure mode management mechanisms for Application Software to control or react on modes of the ECU or vehicle.

Rationale: The behavior of Application Software highly depends on the overall mode of the ECU. Therefore the method of mode management has to be standardized to achieve the same behavior if Application Software is allocated on another ECU.

[RS_Main_00503] AUTOSAR shall support change of communication and application software at runtime.

Description: Advanced systems require dynamic allocation of AUTOSAR Applications and adaptations of the communication topology after development and production at life-time of the system AUTOSAR shall provide a technical possibility which provides these Software changes at runtime.

Rationale: Advanced driving assistance functions have to be updated (e.g. after development or production).

Use-Case: Update of Application Software or update of configuration over the air

Applies to: AP

Dependencies: –

–Last change on issue 80615 comment 8–

BW-C-Level:

Application	Specification	Bus
1	1	1

11.10 Specification Item RS_Main_00250

Trace References:

RS_PO_00007

Content:

Type:	Valid
Description:	The definition and description of roles and activities in the design methodology should support a work-share model.
Rationale:	As AUTOSAR enables work-share on different positions and activities it shall provide a common understanding of roles and activities.
AppliesTo:	AP, CP, FO
Use Case:	Share activities like AUTOSAR configuration and Application Software partitioning between software integrator and software architect.
Supporting Material:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #81369: RS Main - requirements header format not aligned

Problem description:

The header format of the requirements in the RS_Main document is not aligned. Some of them end with a dot and some not. This should be changed to either all with or all without dot.

Example:

[RS_Main_00050] AUTOSAR shall provide an Execution Framework towards applications to implement concurrent application internal control flows.

RS_Main_00106] AUTOSAR shall provide the possibility to extend the software with new SWCs without recompiling the platform foundation

Agreed solution:

Solution: Remove all dots at the end of RS_Main requirements
–Last change on issue 81369 comment 1–

BW-C-Level:

Application	Specification	Bus
1	1	1

11.11 Specification Item RS_Main_00350

Trace References:

RS_PO_00005, RS_PO_00009

Content:

Type:	Valid
Description:	To achieve safety-related properties an adequate software architectural design and implementation matching the safety requirements is required and has to be demonstrated. Such demonstration can be done by safety analyses, therefore.
Rationale:	In the context of the safety-related developments a confirmation that design and implementation are adequately safe is required.
AppliesTo:	AP, CP, FO
Use Case:	–
Supporting Material:	ISO26262

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #81369: RS Main - requirements header format not aligned

Problem description:

The header format of the requirements in the RS_Main document is not aligned. Some of them end with a dot and some not. This should be changed to either all with or all without dot.

Example:

[RS_Main_00050] AUTOSAR shall provide an Execution Framework towards applications to implement concurrent application internal control flows.

RS_Main_00106] AUTOSAR shall provide the possibility to extend the software with new SWCs without recompiling the platform foundation

Agreed solution:

Solution: Remove all dots at the end of RS_Main requirements
–Last change on issue 81369 comment 1–

BW-C-Level:

Application	Specification	Bus
1	1	1

11.12 Specification Item RS_Main_00460

Trace References:

RS_PO_00001, RS_PO_00004

Content:

Type:	Valid
Description:	The AUTOSAR Basic Software shall provide a method to configure mode management mechanisms for Application Software to control or react on modes of the ECU / or vehicle.
Rationale:	The behavior of Application Software highly depends on the overall mode of the ECUand/or the System. Therefore the overall method of mode management has to be standardized to achieve the same behavior if Application Software is transferred from one ECU to allocated on another ECUor from one System to another System.
AppliesTo:	AP, CP
Use Case:	Degradation of application functionality in certain power modes.
Supporting Material:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80615: Umbrella RfC for RS_Main working list R18-10 / FO1.5.0

Problem description:

This RfC is used to collect the RfCs currently discussed in the PL working list:

https://svn.autosar.org/repos/work/26_Standards/00_FO_R1/01_Sources/Elaboration/RS_Main/F

RfCs are discussed in the working list to achieve an agreement. Duplicate RfCs or resolved RfCs targeting single changes will be linked to this RfC.

For the release all agreed changes from the working list are updated to this RfC and implemented.

Agreed solution:

[RS_Main_00170] AUTOSAR shall provide secure access to ECU

Description: AUTOSAR shall provide secured access to ECU by secure authentication of external ECU users. For this mechanisms access control decisions need to be enforced.

Rationale: Secure access and authentication mechanisms are required for prevention of unauthorized access.

Use Case: Secure V2X connection

Applies to: AP, CP

Dependencies: To fulfill this requirement it is also necessary that the environment that is not standardized by AUTOSAR (e.g. bootloader) matches the same security requirements.

Supporting Material:—

(RS_PO_00004, RS_PO_00005, RS_PO_00009)

[new requirement] AUTOSAR shall support up - and download of data and software

Type: valid

Description: AUTOSAR shall support standardized up- and download of data and software. For all kind of data exchange between off- and onboard artifacts mechanisms and methods shall be defined. These mechanisms and methods shall support common protocols used for data-transfer. Partial updates of the software shall be supported. Independent access control rules and policies apply.

Rationale: Up- and download of data and software is required for software updates using standardized mechanisms.

Use Case: Download of dedicated Software Components in ECU.

Applies to: AP

Dependencies: —

Supporting Material: —

(RS_PO_00004, RS_PO_00005, RS_PO_00009)

remove [RS_Main_00220]

[RS_Main_00460] AUTOSAR shall standardize methods to organize mode management on Application, ECU and System level

Description: AUTOSAR shall provide a method to configure mode management mechanisms for Application Software to control or react on modes of the ECU or vehicle.

Rationale: The behavior of Application Software highly depends on the overall mode of the ECU. Therefore the method of mode management has to be standardized to achieve the same behavior if Application Software is allocated on another ECU.

[RS_Main_00503] AUTOSAR shall support change of communication and application software at runtime.

Description: Advanced systems require dynamic allocation of AUTOSAR Applications and adaptations of the communication topology after development and production at life-time of the system AUTOSAR shall provide a technical possibility which provides these Software changes at runtime.

Rationale: Advanced driving assistance functions have to be updated (e.g. after development or production).

Use-Case: Update of Application Software or update of configuration over the air

Applies to: AP

Dependencies: –

–Last change on issue 80615 comment 8–

BW-C-Level:

Application	Specification	Bus
1	1	1

11.13 Specification Item RS_Main_00503

Trace References:

RS_PO_00005

Content:

Type:	Valid
Description:	Advanced systems require dynamic allocation of AUTOSAR Applications and adaptations of the communication topology after development and production , but at life-time of the system . AUTOSAR shall provide a platform, technical possibility which provides these featuresSoftware changes at runtime.





Rationale:	Advanced driving assistance functions have to be updated over the air (e.This might include changes to the scheduling and the communication behavior of the function and has to be supported by the platformg. after development or production).
AppliesTo:	AP
Use Case:	OEM provides safety Update of Application Software or security related software for installation onto vehicleupdate of configuration over the air
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80615: Umbrella RfC for RS_Main working list R18-10 / FO1.5.0

Problem description:

This RfC is used to collect the RfCs currently discussed in the PL working list:

https://svn.autosar.org/repos/work/26_Standards/00_FO_R1/01_Sources/Elaboration/RS_Main/F

RfCs are discussed in the working list to achieve an agreement. Duplicate RfCs or resolved RfCs targeting single changes will be linked to this RfC.

For the release all agreed changes from the working list are updated to this RfC and implemented.

Agreed solution:

[RS_Main_00170] AUTOSAR shall provide secure access to ECU

Description: AUTOSAR shall provide secured access to ECU by secure authentication of external ECU users. For this mechanisms access control decisions need to be enforced.

Rationale: Secure access and authentication mechanisms are required for prevention of unauthorized access.

Use Case: Secure V2X connection

Applies to: AP, CP

Dependencies: To fulfill this requirement it is also necessary that the environment that is not standardized by AUTOSAR (e.g. bootloader) matches the same security requirements.

Supporting Material:—

(RS_PO_00004, RS_PO_00005, RS_PO_00009)

[new requirement] AUTOSAR shall support up - and download of data and software

Type: valid

Description: AUTOSAR shall support standardized up- and download of data and software. For all kind of data exchange between off- and onboard artifacts mechanisms and methods shall be defined. These mechanisms and methods shall support common protocols used for data-transfer. Partial updates of the software shall be supported. Independent access control rules and policies apply.

Rationale: Up- and download of data and software is required for software updates using standardized mechanisms.

Use Case: Download of dedicated Software Components in ECU.

Applies to: AP

Dependencies: –

Supporting Material: –

(RS_PO_00004, RS_PO_00005, RS_PO_00009)

remove [RS_Main_00220]

[RS_Main_00460] AUTOSAR shall standardize methods to organize mode management on Application, ECU and System level

Description: AUTOSAR shall provide a method to configure mode management mechanisms for Application Software to control or react on modes of the ECU or vehicle.

Rationale: The behavior of Application Software highly depends on the overall mode of the ECU. Therefore the method of mode management has to be standardized to achieve the same behavior if Application Software is allocated on another ECU.

[RS_Main_00503] AUTOSAR shall support change of communication and application software at runtime.

Description: Advanced systems require dynamic allocation of AUTOSAR Applications and adaptations of the communication topology after development and production at life-time of the system AUTOSAR shall provide a technical possibility which provides these Software changes at runtime.

Rationale: Advanced driving assistance functions have to be updated (e.g. after development or production).

Use-Case: Update of Application Software or update of configuration over the air

Applies to: AP

Dependencies: –

–Last change on issue 80615 comment 8–

BW-C-Level:

Application	Specification	Bus
1	1	1

- RfC #81962: Restructure RS_Main based on PL Requirement Level Review

Problem description:

Based on the review from PL team which proposes to split RS_Main into functional and non-functional requirements and identifies stable AUTOSAR requirements the following structure is proposed:

1. functional requirements
2. non-functional requirements
3. Unclear requirement type on RS_Main Level
4. Platform Level Candidates

Chapter 3 will contain requirements with e.g. mixed functional and non-functional parts which have to be evaluated further after FO R1.5.0

Chapter 4 will contain possible candidates for the Platform Level which have to be evaluated further after FO R.1.5.0

Agreed solution:

Restructure requirements based on PL review and change structure to:

1. functional requirements
2. non-functional requirements
3. Unclear requirement type on RS_Main Level
4. Platform Level Candidates

Preliminary List of RS_Main Requirements in chapter 1 and 2:

[RS_Main_00001] AUTOSAR shall provide a software platform for embedded real-time systems

[RS_Main_00010] AUTOSAR shall support the development of safety related systems.

[RS_Main_00012] AUTOSAR shall provide a software platform to support the development of highly available systems.

[RS_Main_00030] AUTOSAR shall support development processes for safety related systems

[RS_Main_00060] AUTOSAR shall provide a standardized software interface for communication between Applications

[RS_Main_00080] AUTOSAR shall provide means to describe a component model for Application Software

[RS_Main_00120] AUTOSAR shall provide means to assure interoperability of AUTOSAR implementations (ICC1 level) on application level (RTE) and bus level.

[RS_Main_00127] AUTOSAR shall provide generic test cases

[RS_Main_00130] AUTOSAR shall provide an abstraction from hardware

[RS_Main_00140] AUTOSAR shall provide network independent communication mechanisms for applications

[RS_Main_00160] AUTOSAR shall provide means to describe interfaces of the entire system.

[RS_Main_00161] AUTOSAR shall provide a unified way to describe software systems deployed to Adaptive and / or Classic platforms.

[RS_Main_00190] AUTOSAR shall support standardized interoperability with non-AUTOSAR software

[RS_Main_00200] AUTOSAR specifications shall allow resource efficient implementations

[RS_Main_00230] AUTOSAR shall support network topologies including gateways

[RS_Main_00250] AUTOSAR methodology shall provide a predefinition of typical roles and activities.

[RS_Main_00170] AUTOSAR shall provide secure access to ECU

[RS_Main_00260] AUTOSAR shall provide diagnostics means during runtime, for production and services purposes

[RS_Main_00280] AUTOSAR shall support standardized automotive communication protocols

[RS_Main_00300] AUTOSAR shall provide data exchange formats to support work-share in large inter and intra company development groups

[RS_Main_00320] AUTOSAR shall provide formats to specify system development

[RS_Main_00340] AUTOSAR shall support the continuous timing requirement analysis

[RS_Main_00350] AUTOSAR specifications shall be analyzable and support according methods to demonstrate the achievement of safety related properties.

[RS_Main_00360] AUTOSAR shall support variant management

[RS_Main_00460] AUTOSAR shall standardize methods to organize mode management on Application, ECU and System level

[RS_Main_00480] AUTOSAR shall support the test of implementations

[RS_Main_00491] AUTOSAR shall provide means for logging

[RS_Main_00500] AUTOSAR shall provide naming conventions

[RS_Main_00501] AUTOSAR shall support redundancy concepts

[RS_Main_00514] AUTOSAR shall support the development of secure systems

[RS_Main_01001] AUTOSAR shall support intra ECU communication

–Last change on issue 81962 comment 6–

BW-C-Level:

Application	Specification	Bus
1	1	1

11.14 Specification Item RS_Main_00510

Trace References:

RS_PO_00004, RS_PO_00005, RS_PO_00008

Content:

Type:	Valid
Description:	AUTOSAR shall provide means to check data authenticity, data integrity, optionally confidentiality and data freshness in inter ECU communication.
Rationale:	Dependable systems rely on authentic and trustworthy exchange of information between ECUs. Protecting and assuring data authenticity, data integrity and data freshness in inter ECU communication allows for the development of secure and safe systems by using the AUTOSAR platform. Confidentiality ensures privacy of data.
AppliesTo:	AP, CP
Use Case:	Protection of on-board communication against manipulation or eavesdropping.
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #81908: [CONC_645] [MS4] [FO] Concept incorporation

Problem description:

This RfC is derived from AUTOSAR concept:

CONC_645 "TLS" [1] [2]

It is intended to track the incorporation of the concept's technical solution into the AUTOSAR CP and FO standard as draft since there is no external validation in 2018.

There is a PL decision and also coordinated by WP-X-VAL that the CONC_645 will be released as draft in R4.4.0/18-10 and the concept will be validated 2019 by an external validation.

For each impacted document, a copy& paste ready solution is provided by the concept and a respective implementation task is created. In case of questions, please contact the concept owner:

Armin Happel (armin.happel@vector.com)

For the CP part a further Bugzilla Ticket will be created since difference Bugzilla

projects.

Best regards, QA

[1] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_645_TransportLayerSecurity

[2] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_645_TransportLayerSecurity

Agreed solution:

Incorporate the CONC_645 as "draft" in your document according to [1] an [2].

RS_Main:

Changing RS_Main_00510 in chapter 4.1.21

See https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_645_TransportLayerSecurity

TR_Glossary:

https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_645_TransportLayerSecurity/W

Example to mark requirements as "draft":

1) For RS and SRS: Please set the field "Type" to "draft"

2) For SWS: Please use the annotation: [SWS_<shortname>_<XXXXX>] DRAFT
<Text> ((S)RS_ID)

3) For EXP/TR: Please add a sentence to chapter "Limitations" and list the used concept parts and provide a sentence that these concpet parts are "draft"

If possible a bit more elaborated description of what is draft would be great. Just some additional words or sentences what the concept contains and what this means for your document. Maybe there even is some specific chapter where things coming from the concept are bundled to which you could refer in the limitations chapter, stating that everything that is described there is implemented as draft in SWS and RS documents.

—Last change on issue 81908 comment 5—

BW-C-Level:

Application	Specification	Bus
-	-	-

11.15 Specification Item RS_Main_00514

Trace References:

RS_PO_00005

Content:

Type:	Valid
Description:	AUTOSAR shall specify an automotive security approach defining security measures mechanisms and properties.
Rationale:	Protection of data against misuse is important when storing sensitive data (e.g. personal data) The security properties which shall be supported by AUTOSAR include authenticity, credit card information) in the car confidentiality, integrity and non-repudiation.
AppliesTo:	AP, CP
Use Case:	Customer acquires license to use Protect after sales feature or activation and data in his vehicle usage via appropriate security mechanisms.
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #82029: [CONC_636] [MS4] [FO] Concept incorporation

Problem description:

This RfC is derived from AUTOSAR concept:

CONC_636 "Security Extensions" [1] [2]

It is intended to track the incorporation of the concept's technical solution into the AUTOSAR CP, FO and AP standard as draft since there is no external validation in 2018.

For each impacted document, a copy& paste ready solution is provided by the concept and a respective implementation task is created. In case of questions, please contact the concept owner:

Eduard Metzker (eduard.metzker@vector.com)

For SWS_AdaptiveDiagnostics a Jira Ticket will be created.

For CP a separate Bugzilla Ticket will be created

- C1 - "Security Event Memory (SEM)" is released as draft.

- C2 - "Key Management" is released as valid.
- C4 - "STBM with Secure Time" is released as draft.
- C5 - "Diagnostic Policy Manager" is released as valid.
- C6 - "Improvement of Certificate Handling" is released as valid.
- C7 - "Abstract pre-definition of Crypto Items in System Template" is released as valid.

Best regards, QA

[1] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_633_FormalModelQuery/Wo

[2] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_633_FormalModelQuery/Wo

Agreed solution:

Implement the CONC_636 in your document according to [1] an [2].

[1] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_636_SecurityExtensions/Wo

[2] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_636_SecurityExtensions/Wo

- C1 - "Security Event Memory (SEM)" is released as draft.
- C2 - "Key Management" is released as valid.
- C4 - "STBM with Secure Time" is released as draft.
- C5 - "Diagnostic Policy Manager" is released as valid.
- C6 - "Improvement of Certificate Handling" is released as valid.
- C7 - "Abstract pre-definition of Crypto Items in System Template" is released as valid.

=====

RS_Main

5.1 Main requirements

TR_Glossary:

7.3.1 C1: Security Event Memory

SRS_Diagnostic:

7.3.4 C5: Diagnostic Policy Manager

In case of questions, please contact the concept owner:
Eduard Metzker (eduard.metzker@vector.com)

Example to mark requirements as "draft":

1) For RS and SRS: Please set the field "Type" to "draft"

2) For SWS: Please use the annotation: [SWS_<shortname>_<XXXXX>] DRAFT
<Text> ((S)RS_ID)

3) For EXP/TR: Please add a sentence to chapter "Limitations" and list the used concept parts and provide a sentence that these concept parts are "draft"

If possible a bit more elaborated description of what is draft would be great. Just some additional words or sentences what the concept contains and what this means for your document. Maybe there even is some specific chapter where things coming from the concept are bundled to which you could refer in the limitations chapter, stating that everything that is described there is implemented as draft in SWS and RS documents.

—Last change on issue 82029 comment 5—

BW-C-Level:

Application	Specification	Bus
-	-	-

11.16 Specification Item RS_Main_00650

Trace References:

RS_PO_00004, RS_PO_00005, RS_PO_00008

Content:

Type:	Valid
Description:	AUTOSAR shall support standardized up- and download of data and software. For all kind of data exchange between off- and onboard artifacts mechanisms and methods shall be defined. These mechanisms and methods shall support common protocols used for data-transfer. Partial updates of the software shall be supported. Independent access control rules and policies apply.
Rationale:	Up- and download of data and software is required for software updates using standardized mechanisms.
AppliesTo:	AP
Use Case:	Download of dedicated Software Components in ECU.
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80615: Umbrella RfC for RS_Main working list R18-10 / FO1.5.0

Problem description:

This RfC is used to collect the RfCs currently discussed in the PL working list:

https://svn.autosar.org/repos/work/26_Standards/00_FO_R1/01_Sources/Elaboration/RS_Main/F

RfCs are discussed in the working list to achieve an agreement. Duplicate RfCs or resolved RfCs targeting single changes will be linked to this RfC.

For the release all agreed changes from the working list are updated to this RfC and implemented.

Agreed solution:

[RS_Main_00170] AUTOSAR shall provide secure access to ECU

Description: AUTOSAR shall provide secured access to ECU by secure authentication of external ECU users. For this mechanisms access control decisions need to be enforced.

Rationale: Secure access and authentication mechanisms are required for prevention of unauthorized access.

Use Case: Secure V2X connection

Applies to: AP, CP

Dependencies: To fulfill this requirement it is also necessary that the environment that is not standardized by AUTOSAR (e.g. bootloader) matches the same security requirements.

Supporting Material:–

(RS_PO_00004, RS_PO_00005, RS_PO_00009)

[new requirement] AUTOSAR shall support up - and download of data and software

Type: valid

Description: AUTOSAR shall support standardized up- and download of data and software. For all kind of data exchange between off- and onboard artifacts mechanisms and methods shall be defined. These mechanisms and methods shall support common protocols used for data-transfer. Partial updates of the software shall be supported. Independent access control rules and policies apply.

Rationale: Up- and download of data and software is required for software updates using standardized mechanisms.

Use Case: Download of dedicated Software Components in ECU.

Applies to: AP

Dependencies: –

Supporting Material: –

(RS_PO_00004, RS_PO_00005, RS_PO_00009)

remove [RS_Main_00220]

[RS_Main_00460] AUTOSAR shall standardize methods to organize mode management on Application, ECU and System level

Description: AUTOSAR shall provide a method to configure mode management mechanisms for Application Software to control or react on modes of the ECU or vehicle.

Rationale: The behavior of Application Software highly depends on the overall mode of the ECU. Therefore the method of mode management has to be standardized to achieve the same behavior if Application Software is allocated on another ECU.

[RS_Main_00503] AUTOSAR shall support change of communication and application software at runtime.

Description: Advanced systems require dynamic allocation of AUTOSAR Applications and adaptations of the communication topology after development and production at life-time of the system AUTOSAR shall provide a technical possibility which provides these Software changes at runtime.

Rationale: Advanced driving assistance functions have to be updated (e.g. after development or production).

Use-Case: Update of Application Software or update of configuration over the air

Applies to: AP

Dependencies: –

–Last change on issue 80615 comment 8–

BW-C-Level:

Application	Specification	Bus
1	1	1

- RfC #81962: Restructure RS_Main based on PL Requirement Level Review

Problem description:

Based on the review from PL team which proposes to split RS_Main into functional and non-functional requirements and identifies stable AUTOSAR requirements the following structure is proposed:

1. functional requirements
2. non-functional requirements
3. Unclear requirement type on RS_Main Level
4. Platform Level Candidates

Chapter 3 will contain requirements with e.g. mixed functional and non-functional

parts which have to be evaluated further after FO R1.5.0

Chapter 4 will contain possible candidates for the Platform Leven which have to be evaluated further after FO R.1.5.0

Agreed solution:

Restructure requirements based on PL review and change structure to:

1. functional requirements
2. non-functional requirements
3. Unclear requirement type on RS_Main Level
4. Platform Level Candidates

Preliminary List of RS_Main Requirements in chapter 1 and 2:

[RS_Main_00001] AUTOSAR shall provide a software platform for embedded real-time systems

[RS_Main_00010] AUTOSAR shall support the development of safety related systems.

[RS_Main_00012] AUTOSAR shall provide a software platform to support the development of highly available systems.

[RS_Main_00030] AUTOSAR shall support development processes for safety related systems

[RS_Main_00060] AUTOSAR shall provide a standardized software interface for communication between Applications

[RS_Main_00080] AUTOSAR shall provide means to describe a component model for Application Software

[RS_Main_00120] AUTOSAR shall provide means to assure interoperability of AUTOSAR implementations (ICC1 level) on application level (RTE) and bus level.

[RS_Main_00127] AUTOSAR shall provide generic test cases

[RS_Main_00130] AUTOSAR shall provide an abstraction from hardware

[RS_Main_00140] AUTOSAR shall provide network independent communication mechanisms for applications

[RS_Main_00160] AUTOSAR shall provide means to describe interfaces of the entire system.

[RS_Main_00161] AUTOSAR shall provide a unified way to describe software systems deployed to Adaptive and / or Classic platforms.

[RS_Main_00190] AUTOSAR shall support standardized interoperability with non-AUTOSAR software

[RS_Main_00200] AUTOSAR specifications shall allow resource efficient implementations

[RS_Main_00230] AUTOSAR shall support network topologies including gateways

[RS_Main_00250] AUTOSAR methodology shall provide a predefinition of typical roles and activities.

[RS_Main_00170] AUTOSAR shall provide secure access to ECU

[RS_Main_00260] AUTOSAR shall provide diagnostics means during runtime, for production and services purposes
 [RS_Main_00280] AUTOSAR shall support standardized automotive communication protocols
 [RS_Main_00300] AUTOSAR shall provide data exchange formats to support work-share in large inter and intra company development groups
 [RS_Main_00320] AUTOSAR shall provide formats to specify system development
 [RS_Main_00340] AUTOSAR shall support the continuous timing requirement analysis
 [RS_Main_00350] AUTOSAR specifications shall be analyzable and support according methods to demonstrate the achievement of safety related properties.
 [RS_Main_00360] AUTOSAR shall support variant management
 [RS_Main_00460] AUTOSAR shall standardize methods to organize mode management on Application, ECU and System level
 [RS_Main_00480] AUTOSAR shall support the test of implementations
 [RS_Main_00491] AUTOSAR shall provide means for logging
 [RS_Main_00500] AUTOSAR shall provide naming conventions
 [RS_Main_00501] AUTOSAR shall support redundancy concepts
 [RS_Main_00514] AUTOSAR shall support the development of secure systems
 [RS_Main_01001] AUTOSAR shall support intra ECU communication
 –Last change on issue 81962 comment 6–

BW-C-Level:

Application	Specification	Bus
1	1	1

11.17 Specification Item RS_Main_00651

Trace References:

[RS_PO_00005](#), [RS_PO_00007](#), [RS_PO_00008](#)

Content:

Type:	Draft
-------	-------





Description:	AUTOSAR shall support direct forwarding of CAN, LIN, and FlexRay frames from multiple source buses to a destination CAN or FlexRay bus, to Ethernet, or to a proprietary network (CDD). The destination can be a diagnostic connector (CAN(-FD) or Ethernet/DiP), an intermediate bus, or a CDD that represents a special bus or a bus controlled by a non-AUTOSAR application. The following forwardings shall be supported:
Rationale:	It is not always possible or sometimes just too complicated to connect an analysis tool directly to an internal network. Forwarding of internal communication to a diagnostic connector allows for observation of internal communication using an external tester.
AppliesTo:	CP
Use Case:	Debugging of internal networks without direct access from an analysis tool.
Supporting Material:	Concept 634 "Bus Mirroring"

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #81967: [CONC_634] [MS4] [FO] Concept incorporation

Problem description:

This RfC is derived from AUTOSAR concept:

CONC_634 "BusMirroring" [1] [2]

It is intended to track the incorporation of the concept's technical solution into the AUTOSAR CP standard .

For each impacted document, a copy& paste ready solution is provided by the concept and a respective implementation task is created. In case of questions, please contact the concept owner:

Martin Schlodder (martin.schlodder@vector.com)

For CP part a separate CP RfC will be created.

Best regards, QA

[1] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_634_BusMirroring/Working/A

[2] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_634_BusMirroring/Working/P

Agreed solution:

Agreed solution is described in chapter 7 of the concept document:

https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_634_BusMirroring/Working/AUT

TR_Glossary

https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_634_BusMirroring/Working/AUT

=====

RS_Main

https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_634_BusMirroring/Working/AUT

BW-C-Level:

Application	Specification	Bus
1	1	1

11.18 Specification Item RS_Main_01025

Trace References:

[RS_PO_00007](#)

Content:

Type:	Draft
Description:	Each AUTOSAR module shall provide a standardized method and interface to enable debugging the software of AUTOSAR systems with awareness of the AUTOSAR architecture. Each part of the AUTOSAR software shall provide methods of obtaining internal state information to be used by debuggers.
Rationale:	Debugging tools need internal information to visualize the state of the software. Components and modules implementing this requirement shall provide the necessary state information that can be used by internal and external tools.
AppliesTo:	AP, CP
Use Case:	Debugging the software.
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80615: Umbrella RfC for RS_Main working list R18-10 / FO1.5.0

Problem description:

This RfC is used to collect the RfCs currently discussed in the PL working list:

https://svn.autosar.org/repos/work/26_Standards/00_FO_R1/01_Sources/Elaboration/RS_Main/F

RfCs are discussed in the working list to achieve an agreement. Duplicate RfCs or resolved RfCs targeting single changes will be linked to this RfC.

For the release all agreed changes from the working list are updated to this RfC and implemented.

Agreed solution:

[RS_Main_00170] AUTOSAR shall provide secure access to ECU

Description: AUTOSAR shall provide secured access to ECU by secure authentication of external ECU users. For this mechanisms access control decisions need to be enforced.

Rationale: Secure access and authentication mechanisms are required for prevention of unauthorized access.

Use Case: Secure V2X connection

Applies to: AP, CP

Dependencies: To fulfill this requirement it is also necessary that the environment that is not standardized by AUTOSAR (e.g. bootloader) matches the same security requirements.

Supporting Material:–

(RS_PO_00004, RS_PO_00005, RS_PO_00009)

[new requirement] AUTOSAR shall support up - and download of data and software

Type: valid

Description: AUTOSAR shall support standardized up- and download of data and software. For all kind of data exchange between off- and onboard artifacts mechanisms and methods shall be defined. These mechanisms and methods shall support common protocols used for data-transfer. Partial updates of the software shall be supported. Independent access control rules and policies apply.

Rationale: Up- and download of data and software is required for software updates using standardized mechanisms.

Use Case: Download of dedicated Software Components in ECU.

Applies to: AP

Dependencies: –

Supporting Material: –

(RS_PO_00004, RS_PO_00005, RS_PO_00009)

remove [RS_Main_00220]

[RS_Main_00460] AUTOSAR shall standardize methods to organize mode management on Application, ECU and System level

Description: AUTOSAR shall provide a method to configure mode management mechanisms for Application Software to control or react on modes of the ECU or vehicle.

Rationale: The behavior of Application Software highly depends on the overall mode of the ECU. Therefore the method of mode management has to be standardized to achieve the same behavior if Application Software is allocated on another ECU.

[RS_Main_00503] AUTOSAR shall support change of communication and application software at runtime.

Description: Advanced systems require dynamic allocation of AUTOSAR Applications and adaptations of the communication topology after development and production at life-time of the system AUTOSAR shall provide a technical possibility which provides these Software changes at runtime.

Rationale: Advanced driving assistance functions have to be updated (e.g. after development or production).

Use-Case: Update of Application Software or update of configuration over the air

Applies to: AP

Dependencies: –

–Last change on issue 80615 comment 8–

BW-C-Level:

Application	Specification	Bus
1	1	1

- RfC #82000: [CONC_628] [MS4] [FO] Concept incorporation

Problem description:

This RfC is derived from AUTOSAR concept:

CONC_628 "ARTI" [1] [2]

It is intended to track the incorporation of the concept's technical solution into the AUTOSAR CP and AP standard as draft since there is no external validation in 2018.

There is a PL decision and also coordinated by WP-X-VAL that the CONC_628 will be released as draft in R4.4.0 and the concept will be validated 2019 by an external validation.

For each impacted document, a copy& paste ready solution is provided by the concept and a respective implementation task is created. In case of questions,

please contact the concept owner:

Peter Gliwa (peter.gliwa@gliwa.com) or Rudolf Dienstbeck
(Rudolf.Dienstbeck@Lauterbach.com)

For CP part a separate Bugzilla issue will be created.

For RS_AdaptivePlatformDebugTraceProfile a Jira Ticket will be created.

New deliverables:

RS_AdaptivePlatformDebugTraceProfile

RS_ClassicPlatformDebugTraceProfile

RS_FoundationDebugTraceProfile

SWS_ClassicPlatformARTI

MOD_ARTI

Best regards, QA

[1] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_628_ARTI/Working/AUTOSA

[2] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_628_ARTI/Documents/NewD

Agreed solution:

Implement the CONC_633 as "draft" in your document according to [1] an [2].

[1] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_628_ARTI/Working/AUTOSA

[2] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_628_ARTI/Documents/NewD

In case of questions, please contact the concept owner:

Peter Gliwa (peter.gliwa@gliwa.com) or Rudolf Dienstbeck
(Rudolf.Dienstbeck@Lauterbach.com)

Example to mark requirements as "draft":

1) For RS and SRS: Please set the field "Type" to "draft"

2) For SWS: Please use the annotation: [SWS_<shortname>_<XXXXX>] DRAFT
<Text> ((S)RS_ID)

3) For EXP/TR: Please add a sentence to chapter "Limitations" and list the used concept parts and provide a sentence that these concept parts are "draft"

If possible a bit more elaborated description of what is draft would be great. Just some additional words or sentences what the concept contains and what this means for your document. Maybe there even is some specific chapter where things

coming from the concept are bundled to which you could refer in the limitations chapter, stating that everything that is described there is implemented as draft in SWS and RS documents.

–Last change on issue 82000 comment 4–

BW-C-Level:

Application	Specification	Bus
-	-	-

11.19 Specification Item RS_Main_01026

Trace References:

RS_PO_00007

Content:

Type:	Draft
Description:	Each AUTOSAR module shall provide a standardized method and interface to enable tracing and profiling the software of AUTOSAR systems with awareness of the AUTOSAR architecture. Each part of the AUTOSAR software shall provide methods of obtaining event information to be used by trace analysis tools.
Rationale:	Tracing and timing analysis tools need internal information to visualize and inspect the run-time behavior of the software. Components and modules implementing this requirement shall provide the necessary details and hooks that can be used by tools.
AppliesTo:	AP, CP
Use Case:	Run-time tracing the software, profiling, timing measurement.
Supporting Material:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80615: Umbrella RfC for RS_Main working list R18-10 / FO1.5.0

Problem description:

This RfC is used to collect the RfCs currently discussed in the PL working list:

https://svn.autosar.org/repos/work/26_Standards/00_FO_R1/01_Sources/Elaboration/RS_Main/F

RfCs are discussed in the working list to achieve an agreement. Duplicate RfCs or resolved RfCs targeting single changes will be linked to this RfC.

For the release all agreed changes from the working list are updated to this RfC and implemented.

Agreed solution:

[RS_Main_00170] AUTOSAR shall provide secure access to ECU

Description: AUTOSAR shall provide secured access to ECU by secure authentication of external ECU users. For this mechanisms access control decisions need to be enforced.

Rationale: Secure access and authentication mechanisms are required for prevention of unauthorized access.

Use Case: Secure V2X connection

Applies to: AP, CP

Dependencies: To fulfill this requirement it is also necessary that the environment that is not standardized by AUTOSAR (e.g. bootloader) matches the same security requirements.

Supporting Material:–

(RS_PO_00004, RS_PO_00005, RS_PO_00009)

[new requirement] AUTOSAR shall support up - and download of data and software

Type: valid

Description: AUTOSAR shall support standardized up- and download of data and software. For all kind of data exchange between off- and onboard artifacts mechanisms and methods shall be defined. These mechanisms and methods shall support common protocols used for data-transfer. Partial updates of the software shall be supported. Independent access control rules and policies apply.

Rationale: Up- and download of data and software is required for software updates using standardized mechanisms.

Use Case: Download of dedicated Software Components in ECU.

Applies to: AP

Dependencies: –

Supporting Material: –

(RS_PO_00004, RS_PO_00005, RS_PO_00009)

remove [RS_Main_00220]

[RS_Main_00460] AUTOSAR shall standardize methods to organize mode management on Application, ECU and System level

Description: AUTOSAR shall provide a method to configure mode management mechanisms for Application Software to control or react on modes of the ECU or vehicle.

Rationale: The behavior of Application Software highly depends on the overall mode of the ECU. Therefore the method of mode management has to be standardized to achieve the same behavior if Application Software is allocated on another ECU.

[RS_Main_00503] AUTOSAR shall support change of communication and application software at runtime.

Description: Advanced systems require dynamic allocation of AUTOSAR Applications and adaptations of the communication topology after development and production at life-time of the system AUTOSAR shall provide a technical possibility which provides these Software changes at runtime.

Rationale: Advanced driving assistance functions have to be updated (e.g. after development or production).

Use-Case: Update of Application Software or update of configuration over the air

Applies to: AP

Dependencies: –

–Last change on issue 80615 comment 8–

BW-C-Level:

Application	Specification	Bus
1	1	1

- RfC #82000: [CONC_628] [MS4] [FO] Concept incorporation

Problem description:

This RfC is derived from AUTOSAR concept:

CONC_628 "ARTI" [1] [2]

It is intended to track the incorporation of the concept's technical solution into the AUTOSAR CP and AP standard as draft since there is no external validation in 2018.

There is a PL decision and also coordinated by WP-X-VAL that the CONC_628 will be released as draft in R4.4.0 and the concept will be validated 2019 by an external validation.

For each impacted document, a copy& paste ready solution is provided by the concept and a respective implementation task is created. In case of questions, please contact the concept owner:

Peter Gliwa (peter.gliwa@gliwa.com) or Rudolf Dienstbeck (Rudolf.Dienstbeck@Lauterbach.com)

For CP part a separate Bugzilla issue will be created.

For RS_AdaptivePlatformDebugTraceProfile a Jira Ticket will be created.

New deliverables:

RS_AdaptivePlatformDebugTraceProfile
RS_ClassicPlatformDebugTraceProfile
RS_FoundationDebugTraceProfile
SWS_ClassicPlatformARTI
MOD_ARTI

Best regards, QA

[1] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_628_ARTI/Working/AUTOSA

[2] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_628_ARTI/Documents/NewD

Agreed solution:

Implement the CONC_633 as "draft" in your document according to [1] an [2].

[1] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_628_ARTI/Working/AUTOSA

[2] https://svn.autosar.org/repos/work/25_Concepts/Sources/CONC_628_ARTI/Documents/NewD

In case of questions, please contact the concept owner:

Peter Gliwa (peter.gliwa@gliwa.com) or Rudolf Dienstbeck
(Rudolf.Dienstbeck@Lauterbach.com)

Example to mark requirements as "draft":

1) For RS and SRS: Please set the field "Type" to "draft"
2) For SWS: Please use the annotation: [SWS_<shortname>_<XXXXX>] DRAFT
<Text> ((S)RS_ID)

3) For EXP/TR: Please add a sentence to chapter "Limitations" and list the used concept parts and provide a sentence that these concept parts are "draft"

If possible a bit more elaborated description of what is draft would be great. Just some additional words or sentences what the concept contains and what this means for your document. Maybe there even is some specific chapter where things coming from the concept are bundled to which you could refer in the limitations chapter, stating that everything that is described there is implemented as draft in SWS and RS documents.

—Last change on issue 82000 comment 4—

BW-C-Level:

Application	Specification	Bus
-	-	-

12 RS_Methodology

12.1 Specification Item RS_METH_00062

Trace References:

RS_Main_00080, RS_Main_00360

Content:

Type:	valid
Description:	The AUTOSAR methodology shall support allow system development with different combinations of configuration classes variant handling support.
Rationale:	Configuration The configuration of parameters can be performed in different process steps : pre-compile, link time, and post-build at different times.
AppliesTo:	CP, AP
Use Case:	OEM configuration of (post-build) data after a release from a Tier1 Tier 1 supplier.
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80500: Variant handling requirements shall be made available to CP and AP

Problem description:

The variant handling requirements of the current document FO_RS_Methodology_362 shall be made available to both platforms, the Classic Platform as well as the Adaptive Platform.

Agreed solution:

1) Move the sub-section 'Variant Handling Requirements' from the section 'Requirements for the Classic Platform' to the section 'General Requirements', including all enclosed requirements.

2) Changes of RS_METH_00062:

Description:

The AUTOSAR methodology shall allow system development with variant handling support.

Rationale:

The configuration of parameters can be performed in different process steps at different times.

Use Case:

OEM configuration of (post-build) data after a release from a Tier 1 supplier.

Handling information related to different configuration items (units for version control).

Applies to:

CP,AP

3) Change of RS_METH_00074:

Applies to:

CP,AP

4) Change of RS_METH_00075:

Applies to:

CP,AP

5) Change of RS_METH_00076:

Applies to:

CP,AP

BW-C-Level:

Application	Specification	Bus
1	1	1

12.2 Specification Item RS_METH_00074

Trace References:

RS_Main_00360

Content:

Type:	valid
Description:	The AUTOSAR Methodology shall specify particular points in the workflow on which variation can be resolved.
Rationale:	Need for a stable reference on Binding times.
AppliesTo:	CP, AP
Use Case:	During the development of an AUTOSAR System and ECU, specific variants need to be created, and eventual chosen, e.g pre-compile, or post-build.
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80500: Variant handling requirements shall be made available to CP and AP

Problem description:

The variant handling requirements of the current document FO_RS_Methodology_362 shall be made available to both platforms, the Classic Platform as well as the Adaptive Platform.

Agreed solution:

1) Move the sub-section 'Variant Handling Requirements' from the section 'Requirements for the Classic Platform' to the section 'General Requirements', including all enclosed requirements.

2) Changes of RS_METH_00062:

Description:

The AUTOSAR methodology shall allow system development with variant handling support.

Rationale:

The configuration of parameters can be performed in different process steps at different times.

Use Case:

OEM configuration of (post-build) data after a release from a Tier 1 supplier.
Handling information related to different configuration items (units for version control).

Applies to:

CP,AP

3) Change of RS_METH_00074:

Applies to:
CP,AP

4) Change of RS_METH_00075:

Applies to:
CP,AP

5) Change of RS_METH_00076:

Applies to:
CP,AP

BW-C-Level:

Application	Specification	Bus
1	1	1

12.3 Specification Item RS_METH_00075

Trace References:

RS_Main_00360

Content:

Type:	valid
Description:	The AUTOSAR Methodology shall specify particular tasks/activities in which variation will be resolved.
Rationale:	Need for clarification of methodology of variants.
AppliesTo:	CP, AP
Use Case:	If two software components provide the same interface in different variants of the system, a task is needed to select the one provider to resolve that system variant.
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80500: Variant handling requirements shall be made available to CP and AP

Problem description:

The variant handling requirements of the current document FO_RS_Methodology_362 shall be made available to both platforms, the Classic Platform as well as the Adaptive Platform.

Agreed solution:

1) Move the sub-section 'Variant Handling Requirements' from the section 'Requirements for the Classic Platform' to the section 'General Requirements', including all enclosed requirements.

2) Changes of RS_METH_00062:

Description:

The AUTOSAR methodology shall allow system development with variant handling support.

Rationale:

The configuration of parameters can be performed in different process steps at different times.

Use Case:

OEM configuration of (post-build) data after a release from a Tier 1 supplier.

Handling information related to different configuration items (units for version control).

Applies to:

CP,AP

3) Change of RS_METH_00074:

Applies to:

CP,AP

4) Change of RS_METH_00075:

Applies to:

CP,AP

5) Change of RS_METH_00076:

Applies to:

CP,AP

BW-C-Level:

Application	Specification	Bus
1	1	1

12.4 Specification Item RS_METH_00076

Trace References:

RS_Main_00360

Content:

Type:	valid
Description:	AUTOSAR Methodology shall specify particular work products to maintain the values of variant selectors.
Rationale:	This makes it clear where the values for variant selectors are stored and maintained.
AppliesTo:	CP, AP
Use Case:	The possible variants are known up front: they are created at a certain time and owned as a work product, and finally consumed when the variant is selected.
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80500: Variant handling requirements shall be made available to CP and AP

Problem description:

The variant handling requirements of the current document FO_RS_Methodology_362 shall be made available to both platforms, the Classic Platform as well as the Adaptive Platform.

Agreed solution:

1) Move the sub-section 'Variant Handling Requirements' from the section 'Requirements for the Classic Platform' to the section 'General Requirements', including all enclosed requirements.

2) Changes of RS_METH_00062:

Description:

The AUTOSAR methodology shall allow system development with variant handling support.

Rationale:

The configuration of parameters can be performed in different process steps at different times.

Use Case:

OEM configuration of (post-build) data after a release from a Tier 1 supplier.

Handling information related to different configuration items (units for version control).

Applies to:

CP,AP

3) Change of RS_METH_00074:**Applies to:**

CP,AP

4) Change of RS_METH_00075:**Applies to:**

CP,AP

5) Change of RS_METH_00076:**Applies to:**

CP,AP

BW-C-Level:

Application	Specification	Bus
1	1	1

12.5 Specification Item RS_METH_00203

Trace References:

RS_Main_00503

Content:

Type:	valid
Description:	The manifest contains all necessary information that is needed in order to integrate applications onto the Adaptive Platform. The methodology shall explain how this information will be collected, for the machine, the service instances as well as for the application itself, and later on how the manifest will be used for configuration purposes.
Rationale:	Methodology consistency using the Manifest Specification
AppliesTo:	AP
Use Case:	The Application Execution Manifest is used for describing all process related aspects of an executable.
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80502: Make TR_METH_00208 also a requirement for AP

Problem description:

The requirement RS_METH_00208 shall also be made available to the Adaptive Platform (AP).

Agreed solution:

1) Move RS_METH_00208 to section 'General Requirements/Main Requirements'

2) Changes of RS_METH_00208:

Header:

The methodology shall support the data exchange between different stakeholders

Description:

AUTOSAR shall define an exchange format for data exchange between different stakeholders during the development of AUTOSAR-based vehicles.

The exchange format shall cover the description of software, network topology and network communication.

Dependencies:

RS_METH_00203

Applies to:

CP,AP

Use case:

An OEM provides a software interface description and network communication description and delivers it to a supplier for the development and integration.

3) Changes to RS_METH_00203:

Rename Application Manifest to Execution Manifest

–Last change on issue 80502 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

12.6 Specification Item RS_METH_00208

Trace References:

RS_Main_00300, RS_Main_00150

Content:

Type:	valid
Description:	The AUTOSAR templates include numerous features shall define an exchange format for data exchange between different stakeholders during the system design, the software component, the ECU configuration etcdevelopment of AUTOSAR-based vehicles. The methodology exchange format shall clearly describe cover the activities to use/modify these description of software, network topology and which activities require them to be completed before commencingnetwork communication.
Rationale:	Usage of AUTOSAR templates in the development process
AppliesTo:	CP, AP
Use Case:	Usage of the SW-C template: An existing OEM provides a software component is implemented interface description and is network communication description and delivers it to be integrated into a suppliers ECU. A description of supplier for the component is needed in order to correctly integrate that component into the ECUdevelopment and integration.
Supporting Material:	–

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- RfC #80502: Make TR_METH_00208 also a requirement for AP

Problem description:

The requirement RS_METH_00208 shall also be made available to the Adaptive Platform (AP).

Agreed solution:

- 1) Move RS_METH_00208 to section 'General Requirements/Main Requirements'
- 2) Changes of RS_METH_00208:

Header:

The methodology shall support the data exchange between different stakeholders

Description:

AUTOSAR shall define an exchange format for data exchange between different stakeholders during the development of AUTOSAR-based vehicles.
The exchange format shall cover the description of software, network topology and network communication.

Dependencies:

RS_METH_00203

Applies to:

CP,AP

Use case:

An OEM provides a software interface description and network communication description and delivers it to a supplier for the development and integration.

- 3) Changes to RS_METH_00203:

Rename Application Manifest to Execution Manifest

–Last change on issue 80502 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

13 RS_NetworkManagement

13.1 Specification Item RS_Nm_00044

Trace References:

RS_Main_00420

Content:

Type:	valid
Description:	Network management mechanisms for each supported protocol shall be realized using a limited number of predefined NM states and NM transitions. The events triggering the transitions between states and the actions taken on these transitions may be protocol specific. A bus sleep mode shall be supported for each protocol. NM shall be executable on asynchronous communication systems (e.g. CAN) as well as on synchronous communication systems (e.g. FlexRay), and also on any other types of communication systems which are in the scope of Autosar.
Rationale:	In today's cars, multiple different communication systems are implemented. For energy consumption, all ECUs have to be able to switch into a low power mode. Therefore, network management is necessary for all communication systems. To facilitate understanding, NM shall be constructed from a common set of state definitions.
AppliesTo:	AP, CP
Use Case:	ECU with CAN and FlexRay, Ethernet
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

13.2 Specification Item RS_Nm_00148**Trace References:**

RS_Main_00140

Content:

Type:	valid
Description:	The specification and implementation shall be split-up into a communication system independent and communication system dependent parts (the communication system dependent parts shall be based on the communication system abstraction).
Rationale:	Re-use





AppliesTo:	CP
Use Case:	CAN NM Software Architecture (AUTOSAR SC decision from Apr 25th, 2006).
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

13.3 Specification Item RS_Nm_02510

Trace References:

RS_Main_00420

Content:

Type:	valid
Description:	For CAN NM it shall be optionally possible that the NM message transmission confirmation is generated at the transmission request to the CAN Interface layer.
Rationale:	If the bus access is completely regulated through an offline system design tool, the actual transmit confirmation to inform the Nm about a successful transmission can be regarded as redundant. Since the maximum arbitration time is known it is acceptable to immediately raise the confirmation at the transmission request time.
AppliesTo:	CP
Use Case:	Usage of CAN NM in a deterministic bus system.
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

13.4 Specification Item RS_Nm_02529

Trace References:

RS_Main_00420

Content:

Type:	valid
Description:	If partial networking is used, the ECU shall secure that the first message on the bus is the wakeup frame. This requirement will be implemented in CanIf.
Rationale:	If all ECUs on the bus use partial networking, they use the CAN transceiver with the partial networking extensions. These transceivers only wake up after receiving the Wakeup Frame.
AppliesTo:	CP
Use Case:	—
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

13.5 Specification Item RS_Nm_02530

Trace References:

RS_Main_00420

Content:

Type:	valid
Description:	CanIf shall provide an optional channel-specific TX filter. In blocking mode, the filter shall only pass transmission of wakeup frames. In pass mode the filter shall pass every PDU transmitted by an upper layer.
Rationale:	—
AppliesTo:	CP
Use Case:	—
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

13.6 Specification Item RS_Nm_02531

Trace References:

RS_Main_00420

Content:

Type:	valid
Description:	CanIf shall provide the possibility to initiate clear and check wake-up flags in the transceiver
Rationale:	—
AppliesTo:	CP
Use Case:	—
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

13.7 Specification Item RS_Nm_02532

Trace References:

RS_Main_00420

Content:

Type:	valid
Description:	When full communication is requested, CanSm shall enable pass mode on the CanIf TX filter
Rationale:	—
AppliesTo:	CP
Use Case:	—
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

13.8 Specification Item RS_Nm_02533

Trace References:

RS_Main_00420

Content:

Type:	valid
Description:	CanSm shall provide the possibility to initiate clear and check wake-up flags in the transceiver
Rationale:	—
AppliesTo:	CP
Use Case:	—
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

13.9 Specification Item RS_Nm_02534

Trace References:

RS_Main_00420

Content:

Type:	valid
Description:	CanSm shall support a validPN shutdown sequence (CAN CC STOP -> CAN TRCV STANBY -> CAN CC SLEEP)
Rationale:	—
AppliesTo:	CP
Use Case:	—
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

13.10 Specification Item RS_Nm_02535

Trace References:

RS_Main_00420

Content:

Type:	valid
Description:	Extent the existing The NM coordination algorithm in shall support coordination of a way that it allows to coordinate a second level of bus hierarchy, when shutting down coordinated bussesbuses. There is no limitation of hierarchy levelswith this concept.
Rationale:	The network management stack allows to have a coordinated shutdown of more than one bus if an ECU exists which is connected to the busses buses which are to be coordinated. The functionality is included in the NmIf module. However, there are currently two limitations
AppliesTo:	CP
Use Case:	Nested Gateways
Supporting Material:	—

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

14 RS_ProjectObjectives

15 RS_SOMEIPProtocol

16 RS_SOMEIPServiceDiscoveryProtocol

17 RS_TimeSync

18 SRS_Diagnostics

18.1 Specification Item SRS_Diag_04230

Trace References:

RS_Main_00170

Content:

Type:	Valid
Description:	Diagnostics in AUTOSAR shall support the ISO 14229-1:2018 service 0x29 Authentication with sub-functions for "Authentication with PKI Certificate Exchange (APCE)" to grant access to diagnostic services. The service shall be implemented as internal service (in the BSW) without interaction with applications over middleware.
Rationale:	The authentication service provides a standardized way in authenticating a tester and ECU and grant access to diagnostic services depending on the certificate content.
AppliesTo:	CP, AP
Use Case:	A repair shop diagnostic tester authenticates with an ECU to gain access to diagnostic services that are explicitly allowed to be executed for a repair shop.
Supporting Material:	Concept 636 "Security Extensions" - C5, ISO14229-1:2018 Authentication Service 0x29

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

18.2 Specification Item SRS_Diag_04231

Trace References:

RS_Main_00260

Content:

Type:	Valid
Description:	Diagnostics in AUTOSAR shall support the client certificate formats CVC and X.509 in service 0x29 diagnostic requests. The supported certificate type shall be configurable.
Rationale:	Only the subset of common known and used client certificate types shall be accepted by diagnostics in AUTOSAR. This allows a standardized handling and evaluation of certificates and content.
AppliesTo:	CP, AP
Use Case:	The OEM PKI issues a certificate for a repair shop diagnostic tester. The diagnostic tester and authenticates itself with the ECU.
Supporting Material:	Concept 636 "Security Extensions"

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

18.3 Specification Item SRS_Diag_04232

Trace References:

RS_Main_00170

Content:

Type:	Valid
Description:	The client certificate extensions shall contain well-defined data with diagnostic access rights. The following access rights types shall be available:





Rationale:	Only well-defined client certificate shall be accepted by diagnostics in AUTOSAR that allows a standardized handling and evaluation of certificates and content.
AppliesTo:	CP, AP
Use Case:	The OEM PKI issues a certificate for a repair shop with role based or individual access rights for diagnostic services.
Supporting Material:	Concept 636 "Security Extensions"

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

18.4 Specification Item SRS_Diag_04233**Trace References:**[RS_Main_00170](#)**Content:**

Type:	Valid
Description:	Certificates provide role based and individual access rights definition. Diagnostics in AUTOSAR shall provide a diagnostic service access right by evaluation properties of services to be executed in the following order:
Rationale:	A definition is required how, the diagnostic service is identified to be executed. Especially the level of granularity is important to reduce the resource consumption to a minimum. The SID check is very coarse but efficient, for services with sub-function the sub-function can be taken into account. Further services with DIDs and RIDs are identified by this identifier only.
AppliesTo:	CP, AP
Use Case:	An authentication state allows to execute any ECU reset service, is restricted to extended session, allows 5 DIDs and one RID to be executed.
Supporting Material:	Concept 636 "Security Extensions"

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

18.5 Specification Item SRS_Diag_04234

Trace References:

RS_Main_00170

Content:

Type:	Valid
Description:	Diagnostics in AUTOSAR shall specify the white list binary layout. This layout shall be compatible for all ECUs independent from the endianness in place.
Rationale:	A definition is required how a white list shall look like so it can be downloaded into any ECU software independent from the used implementation.
AppliesTo:	CP, AP
Use Case:	A certain binary layout for a white list shall define a well-defined set of diagnostic services that are allowed to be executed.
Supporting Material:	Concept 636 "Security Extensions"

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

18.6 Specification Item SRS_Diag_04235

Trace References:

RS_Main_00170

Content:

Type:	Valid
Description:	Diagnostics in AUTOSAR shall evaluate the client certificates validity period and refuse expired or not yet valid certificates.
Rationale:	Control the certificate lifetime and limit the potential of outdated certificates.
AppliesTo:	CP, AP





Use Case:	The OEM PKI issues a certificate (e.g. for a repair shop) for a defined period.
Supporting Material:	Concept 636 "Security Extensions"

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

18.7 Specification Item SRS_Diag_04236**Trace References:**[RS_Main_00170](#)**Content:**

Type:	Valid
Description:	Diagnostics in AUTOSAR shall provide standardized means for target identification. A target can be identified by OEM defined criteria such as VIN, vehicle line or ECU type.
Rationale:	Control the certificates validity on defined targets only.
AppliesTo:	CP, AP
Use Case:	The OEM PKI issues a certificate for a vehicle with a certain VIN or an only for one type of ECU.
Supporting Material:	Concept 636 "Security Extensions"

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

18.8 Specification Item SRS_Diag_04237**Trace References:**[RS_Main_00170](#)**Content:**

Type:	Valid
Description:	Diagnostics in AUTOSAR shall use the diagnostic policy manager to evaluate client certificates from service 0x29 requests. The result of a certificate evaluation is the decision if the certificate is valid and which diagnostic services are allowed for execution. Based on the active certificate it grants access for received diagnostic requests.
Rationale:	AUTOSAR shall define the semantics for the certificate payload. This allows to have a standardized check for certificate validities and evaluation of contained access rights.
AppliesTo:	CP, AP
Use Case:	A repair shop diagnostic tester initiates an authentication by sending its certificate. A set of diagnostic services is available to the diagnostic tester after authentication.
Supporting Material:	Concept 636 "Security Extensions"

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

18.9 Specification Item SRS_Diag_04238

Trace References:

RS_Main_00170

Content:

Type:	Valid
Description:	The diagnostic policy manager shall report a security event every time a certificate is passed for evaluation. The event data shall contain at least the result of the certificate evaluation.
Rationale:	Forensic analysis and interested parties require information which kind of access was requested and granted to diagnostic testers.
AppliesTo:	CP, AP





Use Case:	A certificated with extended access rights is provided by the diagnostic tester. A security event provides information about that specific certificate was provided to the ECU.
Supporting Material:	Concept 636 "Security Extensions"

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

18.10 Specification Item SRS_Diag_04239**Trace References:**[RS_Main_00170](#)**Content:**

Type:	Valid
Description:	Diagnostics in AUTOSAR shall have a configuration to allow execution of dedicated diagnostic services in deauthenticated state.
Rationale:	At least the services to authenticate the tester, shall be available in all authentication states.
AppliesTo:	CP, AP
Use Case:	Sending a service 0x29 in deauthenticated state to reach an authentication state.
Supporting Material:	Concept 636 "Security Extensions"

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

18.11 Specification Item SRS_Diag_04240**Trace References:**[RS_Main_00170](#)

Content:

Type:	Valid
Description:	Diagnostics in AUTOSAR shall provide means to applications to change the authentication state of unauthenticated connections.
Rationale:	An individual authentication with each ECU in the vehicle might be take too much time some for some applications.
AppliesTo:	CP, AP
Use Case:	An application based centralized authentication broadcast is required to gain access to a set of diagnostic services.
Supporting Material:	Concept 636 "Security Extensions"

RfCs affecting this spec item between releases 1.4.0 and 1.5.0:

- Unknown reason for change.

19 SWS_HealthMonitoring