

Document Title	SWS_CompilerAbstraction: Complete Change Documentation 4.3.0 - 4.3.1
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	695
Document Status	Final
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.3.1

Table of Contents

1	SWS_CompilerAbstraction	3
1.1	Specification Item SWS_COMPILER_00066	3
1.2	Specification Item SWS_COMPILER_00067	4
1.3	Specification Item SWS_COMPILER_00068	6

1 SWS_CompilerAbstraction

1.1 Specification Item SWS_COMPILER_00066

Trace References:

none

Content:

The parameter 'MemMapCompilerAddressingMode' shall contain the implementation behind a module-specific memory class symbol.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76235: Clarification regarding Module-Specific Memory Classes and Global Memory Classes.

Problem description:

The information provided for the following requirements under Module-Specific Memory classes and Global Memory Classes sections are not clear.

1. SWS_COMPILER_00066
2. SWS_COMPILER_00067
3. SWS_COMPILER_00068

Please provide more information like use cases of the above mentioned requirements, complete details(multiplicity, scope, Dependency, Value Configuration Class, Post-Build Variant

Value etc) of the all containers and parameters mentioned below.

- 1.MemMapAddressingModeSet
- 2.MemMapCompilerAddressingMode
- 3.MemMapGenericCompilerClass
- 4.MemMapGenericCompilerAddressingMode

Agreed solution:

Change chapter 10.2.1 Module-Specific Memory Classes from

[13] Defines module-specific memory classes in the container MemMapAddressingModeSet. This container has been extended by the parameter MemMapCompilerAddressingMode.

[SWS_COMPILER_00066] The parameter MemMapCompilerAddressingMode shall contain the implementation behind a module-specific memory class symbol. (SRS_BSW_00429])

to

It is also possible to configure module-specific memory classes. This is done by using the container 'MemMapAddressingModeSet' and the contained parameter 'MemMapCompilerMemClassSymbolImpl'. For detailed information about these configuration parameters refer to [13].

Change chapter 10.2.2 Global Memory classes from

Furthermore, there are global memory classes that are valid for all modules. These can be configured in the container MemMapGenericCompilerClass. [SWS_COMPILER_00067] Global memory classes (e.g. REGSPACE) shall be configured in the container MemMapGenericCompilerClass. (SRS_BSW_00429, SRS_BSW_00389)

[SWS_COMPILER_00068] The parameter MemMapGenericCompilerAddressingMode shall contain the implementation behind a global memory class symbol. (SRS_BSW_00429))

to

Furthermore it is possible to configure global memory classes that are valid for all modules. This is done by using the container 'MemMapGenericCompilerMemClass' and the contained parameter 'MemMapGenericCompilerMemClassSymbolImpl'. For detailed information about these configuration parameters refer to [13].

–Last change on issue 76235 comment 11–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.2 Specification Item SWS_COMPILER_00067

Trace References:

SRS_BSW_00429, SRS_BSW_00389

Content:

Global memory classes (e.g. REGSPACE) shall be configured in the container 'MemMapGenericCompilerClass'.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76235: Clarification regarding Module-Specific Memory Classes and Global Memory Classes.

Problem description:

The information provided for the following requirements under Module-Specific Memory classes and Global Memory Classes sections are not clear.

1. SWS_COMPILER_00066
2. SWS_COMPILER_00067
3. SWS_COMPILER_00068

Please provide more information like use cases of the above mentioned requirements, complete details(multiplicity, scope, Dependency, Value Configuration Class, Post-Build Variant

Value etc) of the all containers and parameters mentioned below.

1. MemMapAddressingModeSet
2. MemMapCompilerAddressingMode
3. MemMapGenericCompilerClass
4. MemMapGenericCompilerAddressingMode

Agreed solution:

Change chapter 10.2.1 Module-Specific Memory Classes from

[13] Defines module-specific memory classes in the container MemMapAddressingModeSet. This container has been extended by the parameter MemMapCompilerAddressingMode.

[SWS_COMPILER_00066] The parameter MemMapCompilerAddressingMode shall contain the implementation behind a module-specific memory class symbol. (SRS_BSW_00429])

to

It is also possible to configure module-specific memory classes. This is done by using the container 'MemMapAddressingModeSet' and the contained parameter 'MemMapCompilerMemClassSymbolImpl'. For detailed information about these configuration parameters refer to [13].

Change chapter 10.2.2 Global Memory classes from

Furthermore, there are global memory classes that are valid for all modules. These can be configured in the container MemMapGenericCompilerClass.

[SWS_COMPILER_00067] Global memory classes (e.g. REGSPACE) shall be configured in the container MemMapGenericCompilerClass. (SRS_BSW_00429, SRS_BSW_00389)

[SWS_COMPILER_00068] The parameter MemMapGenericCompilerAddressingMode shall contain the implementation behind a global memory class symbol. (SRS_BSW_00429])

to

Furthermore it is possible to configure global memory classes that are valid for all modules. This is done by using the container 'MemMapGenericCompilerMemClass' and the contained parameter 'MemMapGenericCompilerMemClassSymbolImpl'. For detailed information about these configuration parameters refer to [13].
 –Last change on issue 76235 comment 11–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.3 Specification Item SWS_COMPILER_00068

Trace References:

none

Content:

The parameter 'MemMapGenericCompilerAddressingMode' shall contain the implementation behind a global memory class symbol.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76235: Clarification regarding Module-Specific Memory Classes and Global Memory Classes.

Problem description:

The information provided for the following requirements under Module-Specific Memory classes and Global Memory Classes sections are not clear.

1. SWS_COMPILER_00066
2. SWS_COMPILER_00067
3. SWS_COMPILER_00068

Please provide more information like use cases of the above mentioned require-

ments, complete details(multiplicity, scope, Dependency, Value Configuration Class, Post-Build Variant

Value etc) of the all containers and parameters mentioned below.

- 1.MemMapAddressingModeSet
- 2.MemMapCompilerAddressingMode
- 3.MemMapGenericCompilerClass
- 4.MemMapGenericCompilerAddressingMode

Agreed solution:

Change chapter 10.2.1 Module-Specific Memory Classes from

[13] Defines module-specific memory classes in the container MemMapAddressingModeSet. This container has been extended by the parameter MemMapCompilerAddressingMode.

[SWS_COMPILER_00066] The parameter MemMapCompilerAddressingMode shall contain the implementation behind a module-specific memory class symbol. (SRS_BSW_00429])

to

It is also possible to configure module-specific memory classes. This is done by using the container 'MemMapAddressingModeSet' and the contained parameter 'MemMapCompilerMemClassSymbolImpl'. For detailed information about these configuration parameters refer to [13].

Change chapter 10.2.2 Global Memory classes from

Furthermore, there are global memory classes that are valid for all modules. These

can be configured in the container MemMapGenericCompilerClass.

[SWS_COMPILER_00067] Global memory classes (e.g. REGSPACE) shall be configured in the container MemMapGenericCompilerClass. (SRS_BSW_00429, SRS_BSW_00389)

[SWS_COMPILER_00068] The parameter MemMapGenericCompilerAddressingMode shall contain the implementation behind a global memory class symbol. (SRS_BSW_00429])

to

Furthermore it is possible to configure global memory classes that are valid for all modules. This is done by using the container 'MemMapGenericCompilerMemClass'

and the contained parameter 'MemMapGenericCompilerMemClassSymbolImpl'.
For detailed information about these configuration parameters refer to [13].

–Last change on issue 76235 comment 11–

BW-C-Level:

Application	Specification	Bus
1	1	1