Document Title	SWS_DefaultErrorTracer: 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_DefaultErrorTracer

1.1 Specification Item SWS_Det_00009

Trace References:

SRS_BSW_00310, SRS_Diag_04086, SRS_Diag_04085

Content:

Service name:	Det_ReportErrorDet_ReportError	
Syntax:	Std_ReturnType Det_ReportError(uint16 ModuleId, uint8 InstanceId, uint8 ApiId, uint8 ErrorId)	
Service ID[hex]:	0x01	
Sync/Async:	Depending on implemented functionality: 1. Breakpoint set: no return 2. Internal error counting/logging in RAM: synchronous 3. External error logging via communication interface: asynchronous Not Applicable: The function never returns	
Reentrancy:	Reentrant	
Parameters (in):	ModuleIdDet_ReportError.ModuleId	Module ID of calling module.
	InstanceIdDet_ReportError.InstanceId	The identifier of the index based instance of a module, starting from 0, If the module is a single instance module it shall pass 0 as the InstanceId.
	ApiIdDet_ReportError.ApiId	ID of API service in which error is detected (defined in SWS of calling module)
	ErrorIdDet_ReportError.ErrorId	ID of detected development error (defined in SWS of calling module).
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	returns always E_OK (is required for services) never returns a value, but has a return type for compatibility with services and hooks
Description:	Service to report development errors.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76404: [Det] Clarifications on runtime errors

  Problem description:

  There are several uncertainties/problems in the SWS DET:
  1. According to SWS_Det_00180, the callouts should have the same signatures as the corresponding DET functions, but they are void(void) (SWS_Det_00181,

SWS_Det_00184, SWS_Det_00187).
2. Section 8.2.3.1 does not describe how the instance ID is passed to DET.
3. Configuration of header files for all three error type callouts are missing.
4. Why does the development error callout reside in DetNotification, while the other two callouts reside in DetGeneral?
5. The limitation in section 4.1 regarding "supervisor mode" does not really make sense. It is assumed that the DET is ignorant regarding the call context, and the software receiving DET callbacks (like DLT or the implementers of the callouts) need to take care of resolving the calling context, if necessary (e.g. in multi-core environments).
6. SWS_Det_00302 defines several runtime errors. But apart from DET_E_CANNOT_REPORT, it is unclear in which situation these errors could be reported by DET: For errors reported by BSW, the DET has no means to validate anything that could lead to such an error. And for SWCs, the modeling already takes care that DET_E_WRONG_MODULE and DET_E_WRONG_INSTANCE cannot occur, while the other two errors can also not be checked by DET without further configuration.
7. Det_ReportTransientFault (SWS_Det_01003) shall return the return value of a configured callout. But what shall happen if more than one callout exisits, and the return different values?
8. SWS_Det_00052: The only API that can result in DET_E_PARAM_POINTER is Det_GetVersionInfo (as the error description mentions correctly). Please reformulate this requirement and move it to section 8.1.3.6 "Det_GetVersionInfo".
–Last change on issue 76404 comment 13–

Agreed solution:

1.
~change SWS_Det_00181/184/187 such that signatures match the APIs
~Figures 3,5, and 7 to be corrected (return missing)
5. remove from 4.1. the sentence: "It is assumed that the whole Basic Software runs in supervisor mode or the switch to supervisor mode is done by a system call within the error reporting function of the DET module."
6. remove SWS_Det_00302 and SWS_Det_00303 and all included errors
7. change SWS_Det_01003 (Return Value-Part only): "Std_ReturnType" If no callout exists it shall return E_OK, otherwise it shall return the value of the configured callout. In case several callouts are configured the logical or (sum) of the callout return values shall be returned. Rationale: since E_OK=0, E_OK will be only returned if all are E_OK, and for multiple error codes there is a good chance to detect several of them.
8. change SWS_Det_00052 from "in case a null pointer error occurs." to "in case a null pointer error occurs in Det_GetVersionInfo." Do not move the requirement, since otherwise the section 7.7 would be empty, but add the following sentence to 8.1.3.6: "In case a null pointer is passed, DET_E_PARAM_POINTER is returned,

see SWS_Det_00052."
–Last change on issue 76404 comment 30–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.2 Specification Item SWS_Det_00052

Trace References:

SRS_BSW_00480

Content:

The DET shall notify the error DET_E_PARAM_POINTER to all functions configured in callouts in case a null pointer error occurs in Det_GetVersionInfo.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76404: [Det] Clarifications on runtime errors

  Problem description:

  There are several uncertainties/problems in the SWS DET:
  1. According to SWS_Det_00180, the callouts should have the same signatures as the corresponding DET functions, but they are void(void) (SWS_Det_00181, SWS_Det_00184, SWS_Det_00187).
  2. Section 8.2.3.1 does not describe how the instance ID is passed to DET.
  3. Configuration of header files for all three error type callouts are missing.
  4. Why does the development error callout reside in DetNotification, while the other two callouts reside in DetGeneral?
  5. The limitation in section 4.1 regarding "supervisor mode" does not really make sense. It is assumed that the DET is ignorant regarding the call context, and the software receiving DET callbacks (like DLT or the implementers of the callouts) need to take care of resolving the calling context, if necessary (e.g. in multi-core environments).
  6. SWS_Det_00302 defines several runtime errors. But apart from DET_E_CANNOT_REPORT, it is unclear in which situation these errors could be reported by DET: For errors reported by BSW, the DET has no means to validate anything that could lead to such an error. And for SWCs, the modeling already takes care that DET_E_WRONG_MODULE and DET_E_WRONG_INSTANCE cannot occur, while the other two errors can also not be checked by DET without further configuration.

7. Det_ReportTransientFault (SWS_Det_01003) shall return the return value of a configured callout. But what shall happen if more than one callout exisits, and the return different values?
8. SWS_Det_00052: The only API that can result in DET_E_PARAM_POINTER is Det_GetVersionInfo (as the error description mentions correctly). Please reformulate this requirement and move it to section 8.1.3.6 "Det_GetVersionInfo".
–Last change on issue 76404 comment 13–

Agreed solution:

1.
~change SWS_Det_00181/184/187 such that signatures match the APIs
~Figures 3,5, and 7 to be corrected (return missing)
5. remove from 4.1. the sentence: "It is assumed that the whole Basic Software runs in supervisor mode or the switch to supervisor mode is done by a system call within the error reporting function of the DET module."
6. remove SWS_Det_00302 and SWS_Det_00303 and all included errors
7. change SWS_Det_01003 (Return Value-Part only): "Std_ReturnType" If no callout exists it shall return E_OK, otherwise it shall return the value of the configured callout. In case several callouts are configured the logical or (sum) of the callout return values shall be returned. Rationale: since E_OK=0, E_OK will be only returned if all are E_OK, and for multiple error codes there is a good chance to detect several of them.
8. change SWS_Det_00052 from "in case a null pointer error occurs." to "in case a null pointer error occurs in Det_GetVersionInfo." Do not move the requirement, since otherwise the section 7.7 would be empty, but add the following sentence to 8.1.3.6: "In case a null pointer is passed, DET_E_PARAM_POINTER is returned, see SWS_Det_00052."
–Last change on issue 76404 comment 30–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.3 Specification Item SWS_Det_00181

Trace References:

SRS_BSW_00463

Content:

Service name:	<User_Error_Hooks>User_ErrorHooks

Syntax:	Std_ReturnType <User_Error_Hooks>(uint16 ModuleId, uint8 InstanceId, uint8 ApiId, uint8 ErrorId)	
Service ID[hex]:	0x10	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	ModuleIdUser_ErrorHooks.ModuleId	Module ID of calling module.
	InstanceIdUser_ErrorHooks.InstanceId	The identifier of the index based instance of a module, starting from 0, If the module is a single instance module it shall pass 0 as the InstanceId.
	ApiIdUser_ErrorHooks.ApiId	ID of API service in which error is detected (defined in SWS of calling module)
	ErrorIdUser_ErrorHooks.ErrorId	ID of detected development error (defined in SWS of calling module).
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	returns always E_OK (is required for services)
Description:	–	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76404: [Det] Clarifications on runtime errors

  Problem description:

  There are several uncertainties/problems in the SWS DET:
  1. According to SWS_Det_00180, the callouts should have the same signatures as the corresponding DET functions, but they are void(void) (SWS_Det_00181, SWS_Det_00184, SWS_Det_00187).
  2. Section 8.2.3.1 does not describe how the instance ID is passed to DET.
  3. Configuration of header files for all three error type callouts are missing.
  4. Why does the development error callout reside in DetNotification, while the other two callouts reside in DetGeneral?
  5. The limitation in section 4.1 regarding "supervisor mode" does not really make sense. It is assumed that the DET is ignorant regarding the call context, and the software receiving DET callbacks (like DLT or the implementers of the callouts) need

to take care of resolving the calling context, if necessary (e.g. in multi-core environments).
6. SWS_Det_00302 defines several runtime errors. But apart from DET_E_CANNOT_REPORT, it is unclear in which situation these errors could be reported by DET: For errors reported by BSW, the DET has no means to validate anything that could lead to such an error. And for SWCs, the modeling already takes care that DET_E_WRONG_MODULE and DET_E_WRONG_INSTANCE cannot occur, while the other two errors can also not be checked by DET without further configuration.
7. Det_ReportTransientFault (SWS_Det_01003) shall return the return value of a configured callout. But what shall happen if more than one callout exisits, and the return different values?
8. SWS_Det_00052: The only API that can result in DET_E_PARAM_POINTER is Det_GetVersionInfo (as the error description mentions correctly). Please reformulate this requirement and move it to section 8.1.3.6 "Det_GetVersionInfo".
–Last change on issue 76404 comment 13–

Agreed solution:

1.
~change SWS_Det_00181/184/187 such that signatures match the APIs
~Figures 3,5, and 7 to be corrected (return missing)
5. remove from 4.1. the sentence: "It is assumed that the whole Basic Software runs in supervisor mode or the switch to supervisor mode is done by a system call within the error reporting function of the DET module."
6. remove SWS_Det_00302 and SWS_Det_00303 and all included errors
7. change SWS_Det_01003 (Return Value-Part only): "Std_ReturnType" If no callout exists it shall return E_OK, otherwise it shall return the value of the configured callout. In case several callouts are configured the logical or (sum) of the callout return values shall be returned. Rationale: since E_OK=0, E_OK will be only returned if all are E_OK, and for multiple error codes there is a good chance to detect several of them.
8. change SWS_Det_00052 from "in case a null pointer error occurs." to "in case a null pointer error occurs in Det_GetVersionInfo." Do not move the requirement, since otherwise the section 7.7 would be empty, but add the following sentence to 8.1.3.6: "In case a null pointer is passed, DET_E_PARAM_POINTER is returned, see SWS_Det_00052."
–Last change on issue 76404 comment 30–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.4 Specification Item SWS_Det_00184

Trace References:

SRS_BSW_00463

Content:

Service name:	<DetReportRuntimeErrorCallout>DetReportRuntimeErrorCallout	
Syntax:	Std_ReturnType <DetReportRuntimeErrorCallout>(uint16 ModuleId, uint8 InstanceId, uint8 ApiId, uint8 ErrorId)	
Service ID[hex]:	0x11	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	ModuleIdDetReportRuntimeErrorCallout.ModuleId	Module ID of calling module.
	InstanceIdDetReportRuntimeErrorCallout.InstanceId	The identifier of the index based instance of a module, starting from 0, If the module is a single instance module it shall pass 0 as the InstanceId.
	ApiIdDetReportRuntimeErrorCallout.ApiId	ID of API service in which error is detected (defined in SWS of calling module)
	ErrorIdDetReportRuntimeErrorCallout.ErrorId	ID of detected runtime error (defined in SWS of calling module).
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	returns always E_OK (is required for services)
Description:	–	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76404: [Det] Clarifications on runtime errors

  Problem description:

  There are several uncertainties/problems in the SWS DET:
  1. According to SWS_Det_00180, the callouts should have the same signatures as the corresponding DET functions, but they are void(void) (SWS_Det_00181,

SWS_Det_00184, SWS_Det_00187).
2. Section 8.2.3.1 does not describe how the instance ID is passed to DET.
3. Configuration of header files for all three error type callouts are missing.
4. Why does the development error callout reside in DetNotification, while the other two callouts reside in DetGeneral?
5. The limitation in section 4.1 regarding "supervisor mode" does not really make sense. It is assumed that the DET is ignorant regarding the call context, and the software receiving DET callbacks (like DLT or the implementers of the callouts) need to take care of resolving the calling context, if necessary (e.g. in multi-core environments).
6. SWS_Det_00302 defines several runtime errors. But apart from DET_E_CANNOT_REPORT, it is unclear in which situation these errors could be reported by DET: For errors reported by BSW, the DET has no means to validate anything that could lead to such an error. And for SWCs, the modeling already takes care that DET_E_WRONG_MODULE and DET_E_WRONG_INSTANCE cannot occur, while the other two errors can also not be checked by DET without further configuration.
7. Det_ReportTransientFault (SWS_Det_01003) shall return the return value of a configured callout. But what shall happen if more than one callout exisits, and the return different values?
8. SWS_Det_00052: The only API that can result in DET_E_PARAM_POINTER is Det_GetVersionInfo (as the error description mentions correctly). Please reformulate this requirement and move it to section 8.1.3.6 "Det_GetVersionInfo".
–Last change on issue 76404 comment 13–

Agreed solution:

1.
~change SWS_Det_00181/184/187 such that signatures match the APIs
~Figures 3,5, and 7 to be corrected (return missing)
5. remove from 4.1. the sentence: "It is assumed that the whole Basic Software runs in supervisor mode or the switch to supervisor mode is done by a system call within the error reporting function of the DET module."
6. remove SWS_Det_00302 and SWS_Det_00303 and all included errors
7. change SWS_Det_01003 (Return Value-Part only): "Std_ReturnType" If no callout exists it shall return E_OK, otherwise it shall return the value of the configured callout. In case several callouts are configured the logical or (sum) of the callout return values shall be returned. Rationale: since E_OK=0, E_OK will be only returned if all are E_OK, and for multiple error codes there is a good chance to detect several of them.
8. change SWS_Det_00052 from "in case a null pointer error occurs." to "in case a null pointer error occurs in Det_GetVersionInfo." Do not move the requirement, since otherwise the section 7.7 would be empty, but add the following sentence to 8.1.3.6: "In case a null pointer is passed, DET_E_PARAM_POINTER is returned,

see SWS_Det_00052."
–Last change on issue 76404 comment 30–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.5 Specification Item SWS_Det_00187

Trace References:

SRS_BSW_00463

Content:

Service name:	<DetReportTransientFaultCallout>DetReportTransientFaultCallout	
Syntax:	Std_ReturnType <DetReportTransientFaultCallout>(uint16 ModuleId, uint8 InstanceId, uint8 ApiId, uint8 FaultId)	
Service ID[hex]:	0x12	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	ModuleIdDetReportTransientFaultCallout.ModuleId	Module ID of calling module.
	InstanceIdDetReportTransientFaultCallout.InstanceId	The identifier of the index based instance of a module, starting from 0, If the module is a single instance module it shall pass 0 as the InstanceId.
	ApiIdDetReportTransientFaultCallout.ApiId	ID of API service in which transient fault is detected (defined in SWS of calling module)
	FaultIdDetReportTransientFaultCallout.FaultId	ID of detected transient fault (defined in SWS of calling module).
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	Value is propagated to caller of Det_ReportTransientFault.
Description:	–	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76404: [Det] Clarifications on runtime errors

  Problem description:

  There are several uncertainties/problems in the SWS DET:
  1. According to SWS_Det_00180, the callouts should have the same signatures as the corresponding DET functions, but they are void(void) (SWS_Det_00181, SWS_Det_00184, SWS_Det_00187).
  2. Section 8.2.3.1 does not describe how the instance ID is passed to DET.
  3. Configuration of header files for all three error type callouts are missing.
  4. Why does the development error callout reside in DetNotification, while the other two callouts reside in DetGeneral?
  5. The limitation in section 4.1 regarding "supervisor mode" does not really make sense. It is assumed that the DET is ignorant regarding the call context, and the software receiving DET callbacks (like DLT or the implementers of the callouts) need to take care of resolving the calling context, if necessary (e.g. in multi-core environments).
  6. SWS_Det_00302 defines several runtime errors. But apart from DET_E_CANNOT_REPORT, it is unclear in which situation these errors could be reported by DET: For errors reported by BSW, the DET has no means to validate anything that could lead to such an error. And for SWCs, the modeling already takes care that DET_E_WRONG_MODULE and DET_E_WRONG_INSTANCE cannot occur, while the other two errors can also not be checked by DET without further configuration.
  7. Det_ReportTransientFault (SWS_Det_01003) shall return the return value of a configured callout. But what shall happen if more than one callout exisits, and the return different values?
  8. SWS_Det_00052: The only API that can result in DET_E_PARAM_POINTER is Det_GetVersionInfo (as the error description mentions correctly). Please reformulate this requirement and move it to section 8.1.3.6 "Det_GetVersionInfo".
  –Last change on issue 76404 comment 13–

  Agreed solution:

  1.
  ~change SWS_Det_00181/184/187 such that signatures match the APIs
  ~Figures 3,5, and 7 to be corrected (return missing)
  5. remove from 4.1. the sentence: "It is assumed that the whole Basic Software runs in supervisor mode or the switch to supervisor mode is done by a system call within the error reporting function of the DET module."
  6. remove SWS_Det_00302 and SWS_Det_00303 and all included errors
  7. change SWS_Det_01003 (Return Value-Part only): "Std_ReturnType" If no callout exists it shall return E_OK, otherwise it shall return the value of the configured

callout. In case several callouts are configured the logical or (sum) of the callout return values shall be returned. Rationale: since E_OK=0, E_OK will be only returned if all are E_OK, and for multiple error codes there is a good chance to detect several of them.
8. change SWS_Det_00052 from "in case a null pointer error occurs." to "in case a null pointer error occurs in Det_GetVersionInfo." Do not move the requirement, since otherwise the section 7.7 would be empty, but add the following sentence to 8.1.3.6: "In case a null pointer is passed, DET_E_PARAM_POINTER is returned, see SWS_Det_00052."
–Last change on issue 76404 comment 30–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.6 Specification Item SWS_Det_01003

Trace References:

SRS_BSW_00310, SRS_Diag_04086, SRS_Diag_04144

Content:

Service name:	Det_ReportTransientFaultDet_ReportTransientFault	
Syntax:	Std_ReturnType Det_ReportTransientFault(uint16 ModuleId, uint8 InstanceId, uint8 ApiId, uint8 FaultId)	
Service ID[hex]:	0x05	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	ModuleIdDet_ReportTransient Fault.ModuleId	Module ID of calling module.
	InstanceIdDet_ReportTransient Fault.InstanceId	The identifier of the index based instance of a module, starting from 0, If the module is a single instance module it shall pass 0 as the InstanceId.
	ApiIdDet_ReportTransientFault.ApiId	ID of API service in which transient fault is detected (defined in SWS of calling module)
	FaultIdDet_ReportTransientFault.FaultId	ID of detected transient fault (defined in SWS of calling module).
Parameters (inout):	None	
Parameters (out):	None	

Return value:	Std_ReturnType	Propagates return value of assigned calloutif exists, otherwise If no callout exists it shall return E_OK, otherwise it shall return the value of the configured callout. In case several callouts are configured the logical or (sum) of the callout return values shall be returned. Rationale: since E_OK=0, E_OK will be only returned if all are E_OK, and for multiple error codes there is a good chance to detect several of them.
Description:	Service to report transient faults. If a callout has been configured than this callout shall be called and the returned value of the callout shall be returned. Otherwise it returns immediately with E_OK.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76404: [Det] Clarifications on runtime errors

  Problem description:

  There are several uncertainties/problems in the SWS DET:
  1. According to SWS_Det_00180, the callouts should have the same signatures as the corresponding DET functions, but they are void(void) (SWS_Det_00181, SWS_Det_00184, SWS_Det_00187).
  2. Section 8.2.3.1 does not describe how the instance ID is passed to DET.
  3. Configuration of header files for all three error type callouts are missing.
  4. Why does the development error callout reside in DetNotification, while the other two callouts reside in DetGeneral?
  5. The limitation in section 4.1 regarding "supervisor mode" does not really make sense. It is assumed that the DET is ignorant regarding the call context, and the software receiving DET callbacks (like DLT or the implementers of the callouts) need to take care of resolving the calling context, if necessary (e.g. in multi-core environments).
  6. SWS_Det_00302 defines several runtime errors. But apart from DET_E_CANNOT_REPORT, it is unclear in which situation these errors could be reported by DET: For errors reported by BSW, the DET has no means to validate anything that could lead to such an error. And for SWCs, the modeling already takes care that DET_E_WRONG_MODULE and DET_E_WRONG_INSTANCE cannot occur, while the other two errors can also not be checked by DET without further configuration.
  7. Det_ReportTransientFault (SWS_Det_01003) shall return the return value of a configured callout. But what shall happen if more than one callout exisits, and the return different values?
  8. SWS_Det_00052: The only API that can result in DET_E_PARAM_POINTER is Det_GetVersionInfo (as the error description mentions correctly). Please reformu-

late this requirement and move it to section 8.1.3.6 "Det_GetVersionInfo".
–Last change on issue 76404 comment 13–

Agreed solution:

1.
~change SWS_Det_00181/184/187 such that signatures match the APIs
~Figures 3,5, and 7 to be corrected (return missing)
5. remove from 4.1. the sentence: "It is assumed that the whole Basic Software runs in supervisor mode or the switch to supervisor mode is done by a system call within the error reporting function of the DET module."
6. remove SWS_Det_00302 and SWS_Det_00303 and all included errors
7. change SWS_Det_01003 (Return Value-Part only): "Std_ReturnType" If no callout exists it shall return E_OK, otherwise it shall return the value of the configured callout. In case several callouts are configured the logical or (sum) of the callout return values shall be returned. Rationale: since E_OK=0, E_OK will be only returned if all are E_OK, and for multiple error codes there is a good chance to detect several of them.
8. change SWS_Det_00052 from "in case a null pointer error occurs." to "in case a null pointer error occurs in Det_GetVersionInfo." Do not move the requirement, since otherwise the section 7.7 would be empty, but add the following sentence to 8.1.3.6: "In case a null pointer is passed, DET_E_PARAM_POINTER is returned, see SWS_Det_00052."
–Last change on issue 76404 comment 30–

BW-C-Level:

Application	Specification	Bus
1	4	1