Document Title	Specification of Watchdog Interface
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	041
Document Classification	Standard

Document Version	2.5.0
Document Status	Final
Part of Release	4.0
Revision	3

Document Change History

Date	Version	Changed by	Change Description
08.11.2011	2.5.0	AUTOSAR Administration	• Modification in DeviceIndex • New template with requirements traceability
25.11.2010	2.4.0	AUTOSAR Administration	Update of module version check, addition of invalid pointer as error code and checking for null pointer
30.11.2009	2.3.0	AUTOSAR Administration	• Modifications for windowed watchdog concept • Further maintenance for R4.0: see Chapter 11 • Legal disclaimer revised
23.06.2008	2.2.1	AUTOSAR Administration	Legal disclaimer revised
07.12.2007	2.2.0	AUTOSAR Administration	• The main bullets summarizing the changes are • Tables of chapter 8 has been replaced with • Contents generated from AUTOSAR BSW model • Document meta information extended • Small layout adaptations made
31.01.2007	2.1.0	AUTOSAR Administration	• In chapter 5.1.2 the file include structure has been changed to comply with the SPAL general include structure. • Legal disclaimer revised • Release Notes added • “Advice for users” revised • “Revision Information” added
20.03.2006	2.0.0	AUTOSAR Administration	Document structure adapted to common Release 2.0 SWS Template
23.06.2005	1.0.0	AUTOSAR Administration	Initial release

Disclaimer

Advice for users

AUTOSAR Specification Documents may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the Specification Documents for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such Specification Documents, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

Table of Contents

1 Introduction and functional overview

This specification describes the functionality, API and the configuration of the AUTOSAR Basic Software module Watchdog Interface.

In case of more than one watchdog device and watchdog driver (e.g. both an internal software watchdog and an external hardware watchdog) being used on an ECU, this module allows the watchdog manager (or any other client of the watchdog) to select the correct watchdog driver - and thus the watchdog device - while retaining the API and functionality of the underlying driver.

The Watchdog Interface is part of the Onboard Device Abstraction Layer (see [1]).

[WDGIF026] 「The Watchdog Interface provides uniform access to services of the underlying watchdog drivers like mode switching and setting trigger conditions.」(BSW12165, BSW12167, BSW14019)

2 Acronyms and abbreviations

Note: For this module there are no local acronyms and abbreviations. All used acronyms and abbreviations should be contained in the AUTOSAR glossary.

3 Related documentation

3.1 Input documents

[1] Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf

[2] General Requirements on Basic Software Modules
AUTOSAR_SRS_BSWGeneral.pdf

[3] General Requirements on SPAL
AUTOSAR_SRS_SPALGeneral.pdf

[4] Requirements on Memory Hardware Abstraction Layer
AUTOSAR_SRS_MemoryHWAbstractionLayer.pdf

[5] Specification of Watchdog Driver
AUTOSAR_SWS_WatchdogDriver.pdf

[6] Specification of Development Error Tracer
AUTOSAR_SWS_DevelopmentErrorTracer.pdf

[7] Basic Software Module Description Template
AUTOSAR_TPS_BSWModuleDescriptionTemplate.pdf

[8] AUTOSAR Requirements on Watchdog Driver
AUTOSAR_SRS_WatchdogDriver.pdf

3.2 Related standards and norms

None

4 Constraints and assumptions

4.1 Limitations

No limitations.

4.2 Applicability to car domains

No restrictions.

5 Dependencies to other modules

The Watchdog Interface is part of the ECU Abstraction Layer. It allows the upper layer, especially the watchdog manager, to uniformly access one or more watchdog drivers. The implementation of the Watchdog Interface therefore depends on the number of watchdog drivers below.

5.1 File structure

5.1.1 Code file structure

[WDGIF037] 「The code file structure shall not be completely defined within this specification.」()

[WDGIF051] 「The Watchdog Interface shall comprise, if required, an implementation source file `WdgIf.c` (e.g. for tables of function pointers).」()

5.1.2 Header file structure

[WDGIF001] 「The Watchdog Interface shall comprise a header file "`WdgIf.h`" declaring the API of the Watchdog Interface. If an API is implemented as a macro, it will be also contained here.」()

Note: This is the only header file to be imported by the "user" of the Watchdog Interface.

[WDGIF049] 「The Watchdog Interface shall comprise a header file "`WdgIf_Types.h`" providing type declarations for the watchdog interface and common type declarations to be imported by watchdog drivers.」()

[WDGIF050] 「The Watchdog Interface shall comprise a configuration header file "`WdgIf_Cfg.h`" providing its pre-compile configuration definitions.」(BSW00381)

[WDGIF002] 「The file include structure shall be as follows:

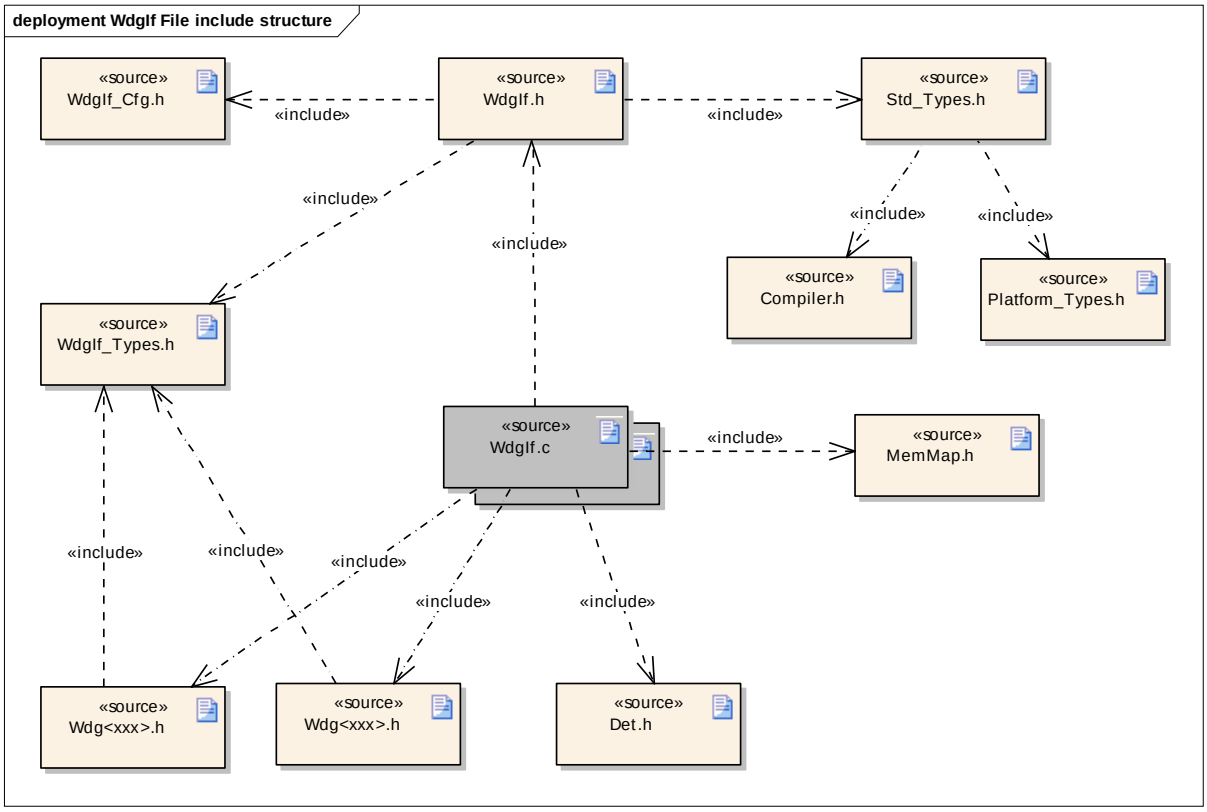

Figure 1: File include structure of the Watchdog Interface」(BSW00348, BSW00353, BSW00361)

Notes to the figure:

- WdgIf may be a pure macro implementation even in the case of configured development error tracing, which means WdgIf.c may not exist. In this case, Det.h and Wdg<xxx>.h must be included in WdgIf.h instead.
- Wdg<xxx>.h has to be included for the API declaration of the watchdog drivers which, in case of multiple existence, have driver specific “infixes” <xxx> according to BSW00374. The figure shows two driver instances as an example.

5.1.3 Version check

[WDGIF005] 「The Watchdog Interface module shall perform Inter Module Checks to avoid integration of incompatible files. The imported included files shall be checked by preprocessing directives. The following version numbers shall be verified:

- <MODULENAME>_AR_RELEASE_MAJOR_VERSION
- <MODULENAME>_AR_RELEASE_MINOR_VERSION

Where <MODULENAME> is the module short name of the other (external) modules which provide header files included by the Watchdog Interface module. If the values are not identical to the expected values, an error shall be reported.」(BSW167, BSW004, BSW14023)

6 Requirements traceability

Requirement	Satisfied by
-	WDGIF041
-	WDGIF054
-	WDGIF030
-	WDGIF048
-	WDGIF013
-	WDGIF044
-	WDGIF045
-	WDGIF043
-	WDGIF058
-	WDGIF052
-	WDGIF051
-	WDGIF001
-	WDGIF037
-	WDGIF055
-	WDGIF036
-	WDGIF049
-	WDGIF042
-	WDGIF053
-	WDGIF047
-	WDGIF057
-	WDGIF035
-	WDGIF010
BSW00300	WDGIF999
BSW00304	WDGIF999
BSW00306	WDGIF999
BSW00307	WDGIF999
BSW00308	WDGIF999
BSW00309	WDGIF999
BSW00312	WDGIF999
BSW00314	WDGIF999
BSW00323	WDGIF028
BSW00325	WDGIF999
BSW00326	WDGIF999
BSW00327	WDGIF006
BSW00328	WDGIF999

BSW00330	WDGIF999
BSW00331	WDGIF999
BSW00335	WDGIF999
BSW00336	WDGIF999
BSW00337	WDGIF009, WDGIF006
BSW00338	WDGIF007
BSW00339	WDGIF999
BSW00342	WDGIF999
BSW00343	WDGIF999
BSW00344	WDGIF999
BSW00347	WDGIF999
BSW00348	WDGIF002
BSW00350	WDGIF031, WDGIF032, WDGIF007
BSW00353	WDGIF002
BSW00355	WDGIF999
BSW00357	WDGIF046
BSW00358	WDGIF999
BSW00359	WDGIF999
BSW00360	WDGIF999
BSW00361	WDGIF002
BSW00370	WDGIF999
BSW00371	WDGIF999
BSW00373	WDGIF999
BSW00375	WDGIF999
BSW00376	WDGIF999
BSW00377	WDGIF999
BSW00378	WDGIF999
BSW00380	WDGIF999
BSW00381	WDGIF050
BSW00383	WDGIF999
BSW00385	WDGIF006
BSW00386	WDGIF007, WDGIF006
BSW00387	WDGIF999
BSW00398	WDGIF999
BSW00399	WDGIF999
BSW004	WDGIF005
BSW00400	WDGIF999
BSW00404	WDGIF999

BSW00405	WDGIF999
BSW00406	WDGIF999
BSW00409	WDGIF009
BSW00412	WDGIF999
BSW00413	WDGIF999
BSW00414	WDGIF999
BSW00415	WDGIF999
BSW00416	WDGIF999
BSW00417	WDGIF999
BSW00419	WDGIF999
BSW00421	WDGIF999
BSW00422	WDGIF999
BSW00423	WDGIF999
BSW00425	WDGIF999
BSW00426	WDGIF999
BSW00427	WDGIF999
BSW00428	WDGIF999
BSW00429	WDGIF999
BSW00432	WDGIF999
BSW00433	WDGIF999
BSW00437	WDGIF999
BSW00438	WDGIF999
BSW00439	WDGIF999
BSW00440	WDGIF999
BSW00441	WDGIF999
BSW00442	WDGIF999
BSW00445	WDGIF999
BSW004450032100341	WDGIF999
BSW0044500333	WDGIF999
BSW0044500334	WDGIF999
BSW0044500401	WDGIF999
BSW00445009	WDGIF999
BSW00445010	WDGIF999
BSW0044512015	WDGIF999
BSW0044512019	WDGIF999
BSW0044512056	WDGIF999
BSW0044512057	WDGIF999
BSW0044512063	WDGIF999

BSW0044512064	WDGIF999
BSW0044512067	WDGIF999
BSW0044512068	WDGIF999
BSW0044512069	WDGIF999
BSW0044512075	WDGIF999
BSW0044512077	WDGIF999
BSW0044512078	WDGIF999
BSW0044512092	WDGIF999
BSW0044512105	WDGIF999
BSW0044512106	WDGIF999
BSW0044512125	WDGIF999
BSW0044512129	WDGIF999
BSW0044512155	WDGIF999
BSW0044512163	WDGIF999
BSW0044512166	WDGIF999
BSW0044512168	WDGIF999
BSW0044512169	WDGIF999
BSW0044512263	WDGIF999
BSW0044512265	WDGIF999
BSW0044512267	WDGIF999
BSW0044512461	WDGIF999
BSW0044512462	WDGIF999
BSW0044512463	WDGIF999
BSW00445157	WDGIF999
BSW00445172	WDGIF999
BSW00446	WDGIF999
BSW00447	WDGIF999
BSW00449	WDGIF999
BSW00450	WDGIF999
BSW005	WDGIF999
BSW007	WDGIF999
BSW0424	WDGIF999
BSW101	WDGIF999
BSW12018	WDGIF016
BSW12165	WDGIF017, WDGIF026
BSW12167	WDGIF017, WDGIF026
BSW12448	WDGIF028
BSW14019	WDGIF017, WDGIF026

BSW14020	WDGIF018
BSW14021	WDGIF020, WDGIF019
BSW14022	WDGIF003
BSW14023	WDGIF028, WDGIF005
BSW14024	WDGIF003
BSW14025	WDGIF020, WDGIF019
BSW159	WDGIF999
BSW161	WDGIF999
BSW162	WDGIF999
BSW164	WDGIF999
BSW167	WDGIF005
BSW168	WDGIF999
BSW170	WDGIF999

Document: General Requirements on Basic Software Modules [2]

Requirement	***Satisfied by***
[[BSW00344] Reference to link-time configuration	Not applicable (this module only provides pre-compile time parameters)
[BSW00404] Reference to post build time configuration	Not applicable (this module only provides pre-compile time parameters)
[BSW00405] Reference to multiple configuration sets	Not applicable (this module does not provide an initialization routine)
[BSW00345] Pre-compile-time configuration	Chapter 10.2
[BSW159] Tool--based configuration	Not applicable (requirement on the implementation)
[BSW167] Static configuration checking	WDGIF005
[BSW171] Configurability of optional functionality	Chapter 10.2
[BSW170] Data for reconfiguration of AUTOSAR SW-components	Not applicable (this module does not depend on faults, signals, …)
[BSW00380] Separate C-File for configuration parameters	Not applicable (this module only provides pre-compile time parameters)
[BSW00419] Separate C-Files for pre-compile time configuration parameters	Not applicable (this module does only provide #define's as pre-compile time configuration parameters)
BSW00381] Separate configuration header file for pre-compile time parameters	WDGIF050
[BSW00412] Separate H-File for configuration parameters	Not applicable (this module only provides pre-compile time parameters)
[BSW00383] List dependencies of configuration files	Not applicable (this module does not use configuration files from other modules)
[BSW00384] List dependencies to other modules	Chapter 5
[BSW00387] Specify the configuration class of callback function	Not applicable (this module does not provide any callback

	functions)
[BSW00388] Introduce containers	Chapter 10.2
[BSW00389] Containers shall have names	Chapter 10.2
[BSW00390] Parameter content shall be unique within the module	Chapter 10.2
[BSW00391] Parameter shall have unique names	Chapter 10.2
[BSW00392] Parameters shall have a type	Chapter 10.2
[BSW00393] Parameters shall have a range	Chapter 10.2
[BSW00394] Specify the scope of the parameters	Chapter 10.2
[BSW00395] List the required parameters (per parameter)	Chapter 10.2
[BSW00396] Configuration classes	Chapter 10.2
[BSW00397] Pre-compile-time parameters	Chapter 10.2
[BSW00398] Link-time parameters	Not applicable (this module does not provide any link-time parameters)
[BSW00399] Loadable Post-build time parameters	Not applicable (this module does not provide any post build parameters)
[BSW00400] Selectable Post-build time parameters	Not applicable (this module does not provide any post build parameters)
[BSW00438] Post Build Configuration Data Structure	Not applicable (this module does not provide any post build parameters)
[BSW00402] Published information	Chapter 10.3
[BSW00375] Notification of wake-up reason	Not applicable (this module does not wake up the ECU / MCU)
[BSW101] Initialization interface	Not applicable (the module does not need to be initialized)
[BSW00416] Sequence of Initialization	Not applicable (requirement on system integration, not on a single module)
[BSW00406] Check module initialization	Not applicable (the module does not need to be initialized)
[BSW00437] NoInit--Area in RAM	Not applicable (the module does not need this feature)
[BSW168] Diagnostic Interface of SW components	Not applicable (the module does not support a special diagnostic interface)
[BSW00407] Function to read out published parameters	Chapter 8.3.3
[BSW00423] Usage of SW-C template to describe BSW modules with AUTOSAR Interfaces	Not applicable (this module does not provide an AUTOSAR interface)
[BSW00424] BSW main processing function task allocation	Not applicable (this module does not provide a main function)
[BSW00425] Trigger conditions for schedulable objects	Not applicable (this module does not provide any scheduled objects)
[BSW00426] Exclusive areas in BSW modules	Not applicable (this module does not have any exclusive areas)
[BSW00427] ISR description for BSW modules	Not applicable (this module does not implement any ISRs)
[BSW00428] Execution order dependencies of main processing functions	Not applicable (this module does not provide a main function)
[BSW00429] Restricted BSW OS functionality access	Not applicable (this module does not use any OS functions or objects)

[BSW00432] Modules should have separate main processing functions for read/receive and write/transmit data path	Not applicable (this module does not provide a main function, much less two)
[BSW00433] Calling of main processing functions	Not applicable (requirement on the BSW task scheduler)
[BSW00450] Main Function Processing for Un-Initialized Modules	Not applicable (this module does not provide a main function)
[BSW00442] Debugging Support in Modules	Not applicable (this module does not have internal states)
[BSW00336] Shutdown interface	Not applicable (the module does not need to be shut down)
[BSW00337] Classification of errors	WDGIF006, WDGIF009
[BSW00338] Detection and Reporting of development errors	WDGIF007
[BSW00369] Do not return development error codes via API	Chapter 8.3
[BSW00339] Reporting of production relevant error status	Not applicable (no production relevant errors)
[BSW00421] Reporting of production relevant error events	Not applicable (no production relevant errors)
[BSW00422] Pre-de-bouncing of production relevant error status	Not applicable (requirement for DEM, not a general requirement)
[BSW00417] Reporting of Error Events by Non-Basic Software	Not applicable (this is a BSW module)
[BSW00323] API parameter checking	WDGIF028
[BSW004] Version check	WDGIF005
[BSW00409] Header files for production code error IDs	WDGIF009
[BSW00385] List possible error notificatons	WDGIF006
[BSW00386] Configuration for detecting an error	WDGIF006, WDGIF007
[BSW161] Microcontroller abstraction	Not applicable (requirement on AUTOSAR architecture, not a single module)
[BSW162] ECU layout abstraction	Not applicable (requirement on AUTOSAR architecture, not a single module)
[BSW005] No hard coded horizontal interfaces within MCAL	Not applicable (requirement on AUTOSAR architecture, not a single module)
[BSW00415] User dependent include files	Not applicable (only one user for this module)
[BSW164] Implementation of interrupt service routines	Not applicable (this module does not implement any ISRs)
[BSW00325] Runtime of interrupt service routines	Not applicable (this module does not implement any ISRs)
[BSW00326] Transition from ISRs to OS tasks	Not applicable (this module does not implement any ISRs)
[BSW00342] Usage of source code and object code	Not applicable (requirement on AUTOSAR architecture, not a single module)
[BSW00343] Specification and configuration of time	Not applicable (no configurable timings)
[BSW160] Human-readable configuration data	Implicitly fulfilled through XML
[BSW007] HIS MISRA C	Not applicable (requirement on implementation, not on specification)
[BSW00300] Module naming convention	Not applicable (requirement on implementation, not on

	specification)
[BSW00413] Accessing instances of BSW modules	Not applicable (this is not a driver)
[BSW00347] Naming separation of different instances of BSW drivers	Not applicable (this is not a driver)
[BSW00441] Enumeration literals and #define naming convention	Not applicable (requirement on implementation)
[BSW00305] Data types naming convention	Chapter 8.2
[BSW00307] Global variables naming convention	Not applicable (requirement on the implementation, not on the specification)
[BSW00310] API naming convention	Chapters 8.3.1, 8.3.2, 8.3.3
[BSW00373] Main processing function naming convention	Not applicable (this module does not provide a main processing function)
[BSW00327] Error values naming convention	WDGIF006
[BSW00335] Status values naming convention	Not applicable (this module does not provide an internal status variable)
[BSW00350] Development error detection keyword	WDGIF007, WDGIF031, WDGIF032
[BSW00408] Configuration parameter naming convention	Chapter 10.2
[BSW00410] Compiler switches shall have defined values	Chapter 10.2
[BSW00411] Get version info keyword	Chapter 10.2
[BSW00346] Basic set of module files	Chapter 5.1
[BSW158] Separation of configuration from implementation	Chapter 5.1
[BSW00314] Separation of interrupt frames and service routines	Not applicable (this module does not implement any ISRs)
[BSW00370] Separation of callback interface from API	Not applicable (this module does not provide any callback routines)
[BSW00435] Module Header File Structure for the Basic Software Scheduler	Chapter 5.1.2
[BSW00436] Module Header File Structure for the Basic Software Memory Mapping	Chapter 5.1.2
[BSW00447] Standardizing Include file structure of BSW Modules Implementing Autosar Service	Not applicable (this module does not implement an Autosar Service)
[BSW00348] Standard type header	WDGIF002
[BSW00353] Platform specific type header	WDGIF002
[BSW00361] Compiler specific language extension header	WDGIF002
[BSW00301] Limit imported information	Chapter 5.1.2
[BSW00302] Limit exported information	Chapter 5.1.2
[BSW00328] Avoid duplication of code	Not applicable (requirement on the implementation, not on the specification)
[BSW00312] Shared code shall be reentrant	Not applicable (requirement on the implementation, not on the specification)
[BSW006] Platform independency	Fulfilled by the design of the WdgIf as an abstraction above the Wdg Driver(s)
[BSW00439] Declaration of interrupt handlers and ISRs	Not applicable (this module does not implement any ISRs)
[BSW00448] Module SWS shall not contain requirements from Other Modules	This is a process requirement; it should be fulfilled throughout the Spec.

[BSW00449] BSW Service APIs used by Autosar Application Software shall return a Std_ReturnType	Not applicable (this module does not implement an Autosar Service)
[BSW00357] Standard API return type	WDGIF046
[BSW00377] Module specific API return types	Not applicable (no module specific return types)
[BSW00304] AUTOSAR integer data types	Not applicable (requirement on implementation, not for specification)
[BSW00355] Do not redefine AUTOSAR integer data types	Not applicable (requirement on implementation, not for specification)
[BSW00378] AUTOSAR boolean type	Not applicable (requirement on implementation, not for specification)
[BSW00306] Avoid direct use of compiler and platform specific keywords	Not applicable (requirement on implementation, not for specification)
[BSW00308] Definition of global data	Not applicable (requirement on implementation, not for specification)
[BSW00309] Global data with read-only constraint	Not applicable (requirement on implementation, not for specification)
[BSW00371] Do not pass function pointers via API	Not applicable (no function pointers in this specification)
[BSW00358] Return type of init() functions	Not applicable (this module does not need to be initiaized)
[BSW00414] Parameter of init function	Not applicable (this module does not need to be initiaized)
[BSW00376] Return type and parameters of main processing functions	Not applicable (this module does not provide a main processing function)
[BSW00359] Return type of callback functions	Not applicable (this module does not provide any callback routines)
[BSW00360] Parameters of callback functions	Not applicable (this module does not provide any callback routines)
[BSW00440] Function prototype for callback functions of AUTOSAR Services	Not applicable (this module does not implement an Autosar Service)
[BSW00329] Avoidance of generic interfaces	Chapters 8.3.1, 8.3.2, 8.3.3 (explicit interfaces defined)
[BSW00330] Usage of macros / inline functions instead of functions	Not applicable (requirement on implementation, not for specification)
[BSW00331] Separation of error and status values	Not applicable (this module does not provide any internal status variable)
[BSW00443] Enabling / disabling defensive behavior of BSW	No concrete requirements for defensive behavior of Wdg were requested.
[BSW00444] Error reporting and logging for defensive behavior	No concrete requirements for defensive behavior of Wdg were requested.
[BSW00445] Protection against untimely call of BSW initialization	Not applicable (this module needs no inititalization)
[BSW00446] Protection against untimely call of BSW de-initialization	Not applicable (this module needs no de-inititalization)
[BSW009] Module User Documentation	Not applicable (requirement on documentation, not on

	specification)
[BSW00401] Documentation of multiple instances of configuration parameters	Not applicable (this module does not need to be initiaized)
[BSW172] Compatibility and documentation of scheduling strategy	Not applicable (no internal scheduling policy)
[BSW010] Memory resource documentation	Not applicable (requirement on documentation, not on specification)
[BSW00333] Documentation of callback function context	Not applicable (this module does not provide any callback routines)
[BSW00374] Module vendor identification	WDGIF034
[BSW00379] Module identification	WDGIF034
[BSW003] Version identification	WDGIF034
[BSW00318] Format of module version numbers	WDGIF034,
[BSW00321] Enumeration of module version numbers	Not applicable (requirement on implementation, not for specification)
[BSW00341] Microcontroller compatibility documentation	Not applicable (requirement on documentation, not on specification)
[BSW00334] Provision of XML file	Not applicable (requirement on documentation, not on specification)

Document: General Requirements on SPAL [3]
Note: This module does not belong to the MCAL layer, but to the Onboard Device Abstraction Layer. Nonetheless certain MCAL requirements might be applicable.

Requirement	***Satisfied by***
[BSW12263] Object code compatible configuration concept	Not applicable (the module is not configurable at runtime)
[BSW12056] Configuration of notification mechanisms	Not applicable (the module does not support any notification mechanism)
[BSW12267] Configuration of wake-up sources	Not applicable (the module does not wake up the ECU / MCU)
[BSW12057] Driver module initialization	Not applicable (the module does not support initialization)
[BSW12125] Initialization of hardware resources	Not applicable (the module does not support initialization)
[BSW12163] Driver module de-initialization	Not applicable (the module does not support initialization)
[BSW12461] Responsibility for register initialization	Not applicable (the module does not support initialization)
[BSW12462] Provide settings for register initialization	Not applicable (the module does not support initialization)
[BSW12463] Combine and forward settings for register initialization	Not applicable (requirement on configuration, not on specification)
[BSW12068] MCAL initialization sequence	Not applicable (not a requirement for a SW module but for system integration)
[BSW12069] Wake-up notification of ECU State Manager	Not applicable (the module does not wake up the ECU / MCU)
[BSW157] Notification mechanisms of drivers and handlers	Not applicable (the module does not support any notification mechanism)

[BSW12155] Prototypes of callback functions	Not applicable (the module does not provide any callback functions)
[BSW12169] Control of operation mode	Not applicable (the module does not support different operating modes)
[BSW12063] Raw value mode	Not applicable (the module does not provide any data to the user)
[BSW12075] Use of application buffers	Not applicable (the module does not operate on buffers)
[BSW12129] Resetting of interrupt flags	Not applicable (the module does not implement any interrupt service routines)
[BSW12064] Change of operation mode during running operation	Not applicable (the module does not support different operating modes)
[BSW12448] Behavior after development error detection	WDGIF028
[BSW12067] Setting of wake-up conditions	Not applicable (the module does not wake up the ECU / MCU)
[BSW12077] Non-blocking implementation	Not applicable (no long term loops)
[BSW12078] Runtime and memory efficiency	Not applicable (requirement for implementation, not for specification)
[BSW12092] Access to drivers	Not applicable (only interface to watchdog drivers)
[BSW12265] Configuration data shall be kept constant	Not applicable (no configuration data)
[BSW12264] Specification of configuration items	Chapter 10.2

Document: Requirements on Watchdog Driver [8]
This document states also requirements for the Watchdog Interface.

Requirement	***Satisfied by***
[BSW12015] Configuration of watchdog modes	Not applicable (this is a requirement for the Wdg Driver only)
[BSW12105] Watchdog initialization	Not applicable (the module does not support initialization)
[BSW12106] Prohibit disabling of watchdog	Not applicable (this is a requirement for the Wdg Driver only)
[BSW12018] Watchdog mode selection service	WDGIF016
[BSW12019] Watchdog trigger service	Not applicable (this is a requirement for the Wdg Driver only)
[BSW12165] Functional scope	WDGIF017, WDGIF026
[BSW12166] SPI channel configuration	Not applicable (this is a requirement for the Wdg Driver only)
[BSW12167] Common Watchdog API	WDGIF017, WDGIF026
[BSW12168] Microcontroller independency	Not applicable (requirement for implementation, not for specification)

Document: Requirements on Memory Hardware Abstraction Layer [4]
These requirements also hold for the Onboard Device Abstraction Layer, as far as applicable, and thus for the Watchdog Interface.

Requirement	***Satisfied by***
BSW14019 Provide uniform access to underlying memory abstraction modules	WDGIF017, WDGIF026
BSW14020 Selection of underlying memory abstraction modules	WDGIF018
BSW14021 Number of underlying memory abstraction modules	WDGIF019, WDGIF020
BSW14022 Preserving of functionality	WDGIF003, WDGIF004
BSW14023 Parameter checking	WDGIF005, WDGIF028
BSW14024 Preserving of timing behavior	WDGIF003,
BSW14025 Efficient implementation	WDGIF019, WDGIF020

7 Functional specification

7.1 General behavior

[WDGIF003] ⌈The Watchdog Interface shall not add functionality to the watchdog drivers. Also the Watchdog Interface does not abstract from watchdog properties like toggle or window mode, timeout periods etc. that is it does not hide any features of the underlying watchdog driver and watchdog hardware.⌋(BSW14022, BSW14024)

7.2 Error classification

[WDGIF006] ⌈The following errors and exceptions shall be detectable by the Watchdog Interface depending on its configuration (development / production).⌋(BSW00337, BSW00385, BSW00386, BSW00327)

Type or error	*Relevance*	*Related error code*	*Value [hex]*
API service called with wrong device index parameter	Development	`WDGIF_E_PARAM_DEVICE`	`0x01`
Invalid pointer in parameter list	Development	`WDGIF_E_INV_POINTER`	`0x02`

[WDGIF030] ⌈Development error values are of type uint8.⌋()

7.3 Error detection

[WDGIF007] ⌈The detection of development errors is configurable (*ON* / *OFF*) at pre-compile time. The switch *`WDGIF_DEV_ERROR_DETECT`* (see chapter 10) shall activate or deactivate the detection of all development errors.⌋(BSW00338, BSW00386, BSW00350)

[WDGIF031] ⌈If the *`WDGIF_DEV_ERROR_DETECT`* switch is enabled API parameter checking is enabled. The detailed description of the detected errors can be found in chapter 7.2.⌋(BSW00350)

7.4 Error notification

[WDGIF032] ⌈Detected development errors shall be reported to the *`Det_ReportError`* service of the Development Error Tracer (DET) if the pre-

processor switch *WDGIF_DEV_ERROR_DETECT* is set (see chapter 10). 」(BSW00350)

[WDGIF009] 「A detection of errors not listed in the table above [WDGIF006] shall not be implemented.」(BSW00337, BSW00409)

7.5 API parameter checking

[WDGIF028] 「If more than one watchdog driver is configured and the development error detection is enabled for this module, the parameter `DeviceIndex` shall be checked for being an existing device within the module's services. Detected errors shall be reported to the Development Error Tracer (DET) with the error code `WDGIF_E_PARAM_DEVICE` and the called service shall not be executed. If the called function has a return value this value shall be set `E_NOT_OK`.」(BSW00323, BSW12448, BSW14023)

7.6 Debugging

[WDGIF052] 「Each variable that shall be accessible by AUTOSAR Debugging shall be defined as global variable.」()

[WDGIF053] 「All type definitions of variables which shall be debugged shall be accessible by the header file WdgIf.h.」()

[WDGIF054] 「The declaration of variables in the header file shall be such, that it is possible to calculate the size of the variables by C-"`sizeof`".」()

[WDGIF055] 「Variables available for debugging shall be described in the respective Basic Software Module Description, see [7].」()

8 API specification

8.1 Imported types

In this chapter all types included from the following files are listed:

[WDGIF041] ⌈

Module	*Imported Type*
Std_Types	Std_ReturnType
	Std_VersionInfoType

⌋()

8.2 Type definitions

[WDGIF010] ⌈The Watchdog Interface's implementer shall place the type definitions of the Watchdog Interface in the file WdgIf_Types.h.⌋()

Note: The implementer of the Watchdog Interface shall not change or extend the type definitions of the Watchdog Interface for a specific watchdog device or platform.

8.2.1 WdgIf_ModeType

Name:	`WdgIf_ModeType`	
Type:	`Enumeration`	
Range:	`WDGIF_OFF_MODE`	In this mode, the watchdog driver is disabled (switched off).
	`WDGIF_SLOW_MODE`	In this mode, the watchdog driver is set up for a long timeout period (slow triggering).
	`WDGIF_FAST_MODE`	In this mode, the watchdog driver is set up for a short timeout period (fast triggering).
Description:	Mode type of the WdgIf module	

[WDGIF016] ⌈The `WdgIf_ModeType` values shall be passed as parameters to the watchdog drivers mode switching function (`Wdg_SetMode`). ⌋(BSW12018)

Note: The hardware specific settings behind these modes are given in the watchdog drivers configuration set.

8.3 Function definitions

[WDGIF017] ⌈The Watchdog Interface shall map the APIs specified in this chapter to the API of the underlying drivers. For functional behavior refer to the specification of the watchdog driver⌋(BSW12165, BSW12167, BSW14019)

[WDGIF018] ⌈The Watchdog Interface shall use the parameter `DeviceIndex` for selection of watchdog drivers. If only one watchdog driver is configured, the parameter `DeviceIndex` shall be ignored.⌋(BSW14020)

[WDGIF013] ⌈The data type for the watchdog device index shall be `uint8`.DeviceIndex shall provide a zero-based consecutive index.⌋()

[WDGIF019] ⌈If only one watchdog driver is configured, the Watchdog Interface shall cause no runtime overhead when mapping the Watchdog Interface API to the API of the corresponding Watchdog Driver.⌋(BSW14021, BSW14025)

Implementation hint: This could be done by using macros as for example
```
#define WdgIf_SetMode(DeviceIndex, WdgMode) \
      Wdg_SetMode(WdgMode)
```

[WDGIF020] ⌈If more than one watchdog driver is configured, the Watchdog Interface shall use efficient mechanisms to map the API calls to the appropriate watchdog driver. ⌋(BSW14021, BSW14025)

Implementation hint: One solution is to use tables of pointers to functions where the parameter `DeviceIndex` is used as array index, for example
```
#define WdgIf_SetMode(DeviceIndex, WdgMode) \
  SetModeFctPtr[DeviceIndex](WdgMode)
```
Note: The service IDs are related to the service IDs of the watchdog driver specification (see [5]). For that reason, they may not start with 0.

8.3.1 WdgIf_SetMode

[WDGIF042] ⌈

Service name:	WdgIf_SetMode	

Syntax:	`Std_ReturnType WdgIf_SetMode(`
	`  uint8 DeviceIndex,`
	`  WdgIf_ModeType WdgMode`
	`)`	

Service ID[hex]:	0x01	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	DeviceIndex	Identifies the Watchdog Driver instance.
	WdgMode	The watchdog driver mode (see Watchdog Driver).
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	--
Description:	Map the service WdgIf_SetMode to the service Wdg_SetMode of the corresponding Watchdog Driver.	

⌋()

[WDGIF043] ⌈The Watchdog Interface module shall map the service `WdgIf_SetMode` to the service `Wdg_SetMode` of the corresponding Watchdog Driver.⌋()

[WDGIF057] ⌈`WdgIf_SetMode` shall return the value which it gets from the service `Wdg_SetMode` of the corresponding Watchdog Driver.⌋()

Possible content of the return value is specified by the Watchdog Driver, see [5].

8.3.2 WdgIf_SetTriggerCondition

[WDGIF044] ⌈

Service name:	WdgIf_SetTriggerCondition	
Syntax:	`void WdgIf_SetTriggerCondition( uint8 DeviceIndex, uint16 Timeout )`	
Service ID[hex]:	0x02	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	DeviceIndex	Identifies the Watchdog Driver instance.
	Timeout	Timeout value (milliseconds) for setting the trigger counter.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Map the service WdgIf_SetTriggerCondition to the service Wdg_SetTriggerCondition of the corresponding Watchdog Driver.	

⌋()

[WDGIF045] ⌈The Watchdog Interface module shall map the service `WdgIf_SetTriggerCondition` to the service `Wdg_SetTriggerCondition` of the corresponding Watchdog Driver.⌋()

8.3.3 WdgIf_GetVersionInfo

[WDGIF046] ⌈

Service name:	WdgIf_GetVersionInfo	
Syntax:	`void WdgIf_GetVersionInfo( Std_VersionInfoType* VersionInfoPtr )`	
Service ID[hex]:	0x03	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	None	
Parameters (inout):	None	
Parameters (out):	VersionInfoPtr	Pointer to where to store the version information of this module.

Return value:	None
Description:	Returns the version information.

」(BSW00357)

[WDGIF058] 「If development error detection for the Watchdog Interface module is enabled, then the function WdgIf_GetVersionInfo shall check whether the parameter VersioninfoPtr is a NULL pointer (NULL_PTR). If VersioninfoPtr is a NULL pointer, then the function WdgIf_GetVersionInfo shall raise the development error WDGIF_E_INV_POINTER (i.e. invalid pointer) and return.
」()

[WDGIF035] 「The `WdgIf_GetVersionInfo` service shall return the version information of this module. The version information includes:

- Module Id
- Vendor Id
- Vendor specific version numbers (BSW00407).」()

[WDGIF036] 「The `WdgIf_GetVersionInfo` function shall be pre compile time configurable `On/Off` by the configuration parameter:
`WDGIF_VERSION_INFO_API`」()

Implementation hint:
If source code for caller and callee of this function is available this function should be realized as a macro. The macro should be defined in the modules header file.

8.4 Call-back notifications

This module does not provide any callback functions.

8.5 Scheduled functions

This module does not need any scheduled functions.

8.6 Expected interfaces

In this chapter all interfaces required from other modules are listed.

8.6.1 Mandatory interfaces

This chapter defines all interfaces which are required to fulfill the core functionality of the module.

[WDGIF047] ⌈

API function	*Description*
Wdg_SetMode	Switches the watchdog into the mode Mode.
Wdg_SetTriggerCondition	Sets the timeout value for the trigger counter.

⌋()

8.6.2 Optional interfaces

This chapter defines all interfaces which are required to fulfill an optional functionality of the module.

[WDGIF048] ⌈

API function	*Description*
Det_ReportError	Service to report development errors.

⌋()

8.6.3 Configurable interfaces

There are no configurable interfaces for this module.

9 Sequence diagrams

Refer to specification of watchdog driver [5].

10 Configuration specification

In general, this chapter defines configuration parameters and their clustering into containers. In order to support the specification Chapter 10.1 describes fundamentals. It also specifies a template (table) you shall use for the parameter specification. We intend to leave Chapter 10.1 in the specification to guarantee comprehension.

Chapter 10.2 specifies the structure (containers) and the parameters of the module WdgIf.

Chapter 10.3 specifies published information of the module WdgIf.

10.1 How to read this chapter

In addition to this section, it is highly recommended to read the documents:

- AUTOSAR Layered Software Architecture
- AUTOSAR ECU Configuration Specification
  This document describes the AUTOSAR configuration methodology and the AUTOSAR configuration metamodel in detail.

The following is only a short survey of the topic and it will not replace the ECU Configuration Specification document.

10.1.1 Configuration and configuration parameters

Configuration parameters define the variability of the generic part(s) of an implementation of a module. This means that only generic or configurable module implementation can be adapted to the environment (software/hardware) in use during system and/or ECU configuration.

The configuration of parameters can be achieved at different times during the software process: before compile time, before link time or after build time. In the following, the term “configuration class” (of a parameter) shall be used in order to refer to a specific configuration point in time.

10.1.2 Containers

Containers structure the set of configuration parameters. This means:

- *all* configuration parameters are kept in containers.
- (sub-) containers can reference (sub-) containers. It is possible to assign a multiplicity to these references. The multiplicity then defines the possible number of instances of the contained parameters.

10.1.3 Specification template for configuration parameters

The following tables consist of three sections:
- the general section
- the configuration parameter section
- the section of included/referenced containers

Pre-compile time - specifies whether the configuration parameter shall be of configuration class *Pre-compile time* or not

Label	***Description***
x	The configuration parameter shall be of configuration class *Pre-compile time*.
--	The configuration parameter shall never be of configuration class *Pre-compile time*.

Link time - specifies whether the configuration parameter shall be of configuration class *Link time* or not

Label	***Description***
x	The configuration parameter shall be of configuration class *Link time*.
--	The configuration parameter shall never be of configuration class *Link time*.

Post Build - specifies whether the configuration parameter shall be of configuration class *Post Build* or not

Label	***Description***
x	The configuration parameter shall be of configuration class *Post Build* and no specific implementation is required.
L	*Loadable* - the configuration parameter shall be of configuration class *Post Build* and only one configuration parameter set resides in the ECU.
M	*Multiple* - the configuration parameter shall be of configuration class *Post Build* and is selected out of a set of multiple parameters by passing a dedicated pointer to the init function of the module.
--	The configuration parameter shall never be of configuration class *Post Build*.

10.2 Containers and configuration parameters

The following chapters summarize all configuration parameters. The detailed meanings of the parameters are described in chapters 7 and 8.

10.2.1 Variants

[WDGIF056] 「This module shall support only the configuration variant `VARIANT-PRE-COMPILE`. Only parameters with "Pre-compile time" configuration are allowed in this variant.」()

10.2.2 WdgIf

SWS Item	**WDGIF033_Conf :**
Module Name	*WdgIf*
Module Description	Configuration of the WdgIf (Watchdog Interface) module.

Included Containers		
Container Name	***Multiplicity***	***Scope / Dependency***
WdgIfDevice	1..*	It contains the information for the selection of a particular Watchdog device in case multiple Watchdog drivers are connected.
WdgIfGeneral	1	This container collects all generic watchdog interface parameters.

10.2.3 WdgIfGeneral

SWS Item	**WdgIf001_Conf :**
Container Name	WdgIfGeneral{WdgIf_ModuleConfiguration}
Description	This container collects all generic watchdog interface parameters.
Configuration Parameters	

SWS Item	**WdgIf005_Conf :**		
Name	WdgIfDevErrorDetect {WDGIF_DEV_ERROR_DETECT}		
Description	Pre-processor switch for enabling the development error detection and reporting. true: Development error detection enabled false: Development error detection disabled		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: Module		

SWS Item	**WdgIf003_Conf :**
Name	WdgIfVersionInfoApi {WDGIF_VERSION_INFO_API}
Description	Pre-processor switch to enable / disable the service returning the version information. true: Version information service enabled false: Version information service disabled
Multiplicity	1

Type	EcucBooleanParamDef		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: Module		

No Included Containers

10.2.4 WdgIfDevice

SWS Item	**WdgIf002_Conf :**
Container Name	WdgIfDevice
Description	It contains the information for the selection of a particular Watchdog device in case multiple Watchdog drivers are connected.
Configuration Parameters	

SWS Item	**WdgIf006_Conf :**		
Name	WdgIfDeviceIndex		
Description	Represents the watchdog interface ID so that it can be referenced by the watchdog manager.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 255		
Default value	--		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency			

SWS Item	**WdgIf007_Conf :**		
Name	WdgIfDriverRef		
Description	Reference to the watchdog drivers that are controlled by the watchdog interface.		
Multiplicity	1		
Type	Reference to [WdgGeneral]		
ConfigurationClass	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency			

No Included Containers

10.3 Published parameters

[WDGIF034]: ⌈The standardized common published parameters as required by BSW00402 in the SRS General on Basic Software Modules [2] shall be published within the header file of this module and need to be provided in the BSW Module Description (see [7]).⌋(BSW00374, BSW00379, BSW003, BSW00318)

Additional module-specific published parameters are listed below if applicable.

11 Changes from Release 3.1 to Release 4.0

11.1 Deleted SWS Items

SWS Item	*Rationale*
WDGIF040	Removed (was redundant)
WDGIF014, WDGIF015	Removed
WDGIF029	Was superfluous
WdgIf004_Conf	Removed

11.2 Replaced SWS Items

SWS Item of Release 1	*replaced by SWS Item*	*Rationale*
WDGIF011	Replaced by a note	

11.3 Changed SWS Items

SWS Item	*Rationale*
WDGIF005	Improved ; changed again
WDGIF034	Updated
WDGIF049	Corrected wording
WDGIF044	Changed into WdgIf_SetTriggerCondition,
WDGIF045	Changed into WdgIf_SetTriggerCondition,
WDGIF042	Improved/added description fields
WDGIF016	Separated note about Wdg Driver from Specification item,
WDGIF001,WDGIF002, WDGIF049,WDGIF050	Reworked to comply with BSW00346,
WDGIF026	Rephrased
WdgIf006_Conf	Refined min and max values,
WDGIF003,WDGIF006, WDGIF010,WDGIF017, WDGIF018,WDGIF020, WDGIF026,WDGIF051	Corrected : Use uniform wording “Watchdog Interface” instead of “Watchdog Driver Interface”.
WDGIF035	Removed Instance ID

11.4 Added SWS Items

SWS Item	*Rationale*
WDGIF052	Debugging concept
WDGIF053	Debugging concept
WDGIF054	Debugging concept
WDGIF055	Debugging concept
WDGIF056	Added specification for compile time configuration variant,
WdgIf001_Conf	Item is not new, but ID was missing
WdgIf002_Conf	Item is not new, but ID was missing
WdgIf003_Conf	Item is not new, but ID was missing
WdgIf005_Conf	Item is not new, but ID was missing
WdgIf006_Conf	Item is not new, but ID was missing, add config. class

WdgIf007_Conf	Item is not new, but ID was missing, renamed symbol
WDGIF057	Added

12 Not applicable requirements

[WDGIF999]「These requirements are not applicable to this specification.」
(BSW00344, BSW00404, BSW00405, BSW159, BSW170, BSW00380, BSW00419, BSW00412, BSW00383, BSW00387, BSW00398, BSW00399, BSW00400, BSW00438, BSW00375, BSW101, BSW00416, BSW00406, BSW00437, BSW168, BSW00423, BSW0424, BSW00425, BSW00426, BSW00427, BSW00428, BSW00429, BSW00432, BSW00433, BSW00450, BSW00442, BSW00336, BSW00339, BSW00421, BSW00422, BSW00417, BSW161, BSW162, BSW005, BSW00415, BSW164, BSW00325, BSW00326, BSW00342, BSW00343, BSW007, BSW00300, BSW00413, BSW00347, BSW00441, BSW00307, BSW00373, BSW00335, BSW00314, BSW00370, BSW00447, BSW00328, BSW00312, BSW00439, BSW00449, BSW00377, BSW00304, BSW00355, BSW00378, BSW00306, BSW00308, BSW00309, BSW00371, BSW00358, BSW00414, BSW00376, BSW00359, BSW00360, BSW00440, BSW00330, BSW00331, BSW00445, BSW00446, BSW00445009, BSW0044500401, BSW00445172, BSW00445010, BSW0044500333, BSW004450032100341, BSW0044500334, BSW0044512263, BSW0044512056, BSW0044512267, BSW0044512057, BSW0044512125, BSW0044512163, BSW0044512461, BSW0044512462, BSW0044512463, BSW0044512068, BSW0044512069, BSW00445157, BSW0044512155, BSW0044512169, BSW0044512063, BSW0044512075, BSW0044512129, BSW0044512064, BSW0044512067, BSW0044512077, BSW0044512078, BSW0044512092, BSW0044512265, BSW0044512015, BSW0044512105, BSW0044512106, BSW0044512019, BSW0044512166, BSW0044512168)