# User Manual

for S32K14X WDG Driver

Document Number: UM2WDGASR4.2 Rev0002 R1.0.1
Rev. 1.0

# Contents

**Section number** **Title** **Page**

## Chapter 1
## Revision History

## Chapter 2
## Introduction



## Chapter 3
## Driver

## Chapter 4
## Tresos Configuration Plug-in

## Chapter 5
## How to Configure

# Chapter 1
# Revision History

**Table 1-1. Revision History**

| Revision | Date | Author | Description |
|---|---|---|---|
| 1.0 | 13/07/2018 | NXP MCAL Team | Updated version for ASR 4.2.2S32K14X1.0.1 Release |

# Chapter 2
# Introduction

This User Manual describes NXP Semiconductors AUTOSAR Watchdog ( Wdg ) for S32K14X .

AUTOSAR Wdg driver configuration parameters and deviations from the specification are described in Wdg Driver chapter of this document. AUTOSAR Wdg driver requirements and APIs are described in the AUTOSAR Wdg driver software specification document.

## 2.1 Supported Derivatives

The software described in this document is intented to be used with the following microcontroller devices of NXP Semiconductors .

**Table 2-1. S32K14X Derivatives**

| | |
|---|---|
| NXP Semiconductors | s32k148_lqfp144, s32k148_lqfp176, s32k148_mapbga100, s32k146_lqfp144, s32k146_lqfp100, s32k146_lqfp64, s32k146_mapbga100, s32k144_lqfp100, s32k144_lqfp64, s32k144_mapbga100, s32k142_lqfp100, s32k142_lqfp64, s32k118_lqfp48, s32k118_lqfp64 |

All of the above microcontroller devices are collectively named as S32K14X .

## 2.2 Overview

**AUTOSAR (AUTomotive Open System ARchitecture)** is an industry partnership working to establish standards for software interfaces and software modules for automobile electronic control systems.

AUTOSAR

- paves the way for innovative electronic systems that further improve performance, safety and environmental friendliness.
- is a strong global partnership that creates one common standard: "Cooperate on standards, compete on implementation".
- is a key enabling technology to manage the growing electrics/electronics complexity. It aims to be prepared for the upcoming technologies and to improve cost-efficiency without making any compromise with respect to quality.
- facilitates the exchange and update of software and hardware over the service life of the vehicle.

## 2.3 About this Manual

This Technical Reference employs the following typographical conventions:

**Boldface** type: Bold is used for important terms, notes and warnings.

*Italic* font: Italic typeface is used for code snippets in the text. Note that C language modifiers such "const" or "volatile" are sometimes omitted to improve readability of the presented code.

Notes and warnings are shown as below:

**Note**

This is a note.

## 2.4 Acronyms and Definitions

**Table 2-2. Acronyms and Definitions**

| Abbreviation and Definitions | Description |
|---|---|
| BSW | Basic Software |
| DEM | Diagnostic Event Manager |
| DET | Development Error Tracer |
| ECU | Electronic Control Unit |
| WDG | Watchdog |
| MCU | MicroController Unit |
| MCL | MicroController Library |
| GPT | General Purpose Timers |
| ISR | interrupt Service Routine |
| OS | Operating System |

*Table continues on the next page...*

**Table 2-2. Acronyms and Definitions (continued)**

| Abbreviation and Definitions | Description |
|---|---|
| RAM | Random Access Memory |
| ROM | Read-only Memory |
| GUI | Graphical User Interface |
| EcuM | ECU state Manager |
| API | Application Programming Interface |
| PB Variant | Post Build Variant |
| PC Variant | Pre Compile Variant |

## 2.5 Reference List

**Table 2-3. Reference List**

| # | Title | Version |
|---|---|---|
| 1 | Specification of Wdg Driver | AUTOSAR Release 4.2.2 |
| 2 | S32K14X Reference Manual | Reference Manual, Rev. 4, 06/2017 |
| 3 | S32K142 Mask Set Errata for Mask 0N33V (0N33V) | 30/11/2017 |
| 4 | S32K144 Mask Set Errata for Mask 0N57U (0N57U) | 30/11/2017 |
| 5 | S32K146 Mask Set Errata for Mask 0N73V (0N73V) | 30/11/2017 |
| 6 | S32K148 Mask Set Errata for Mask 0N20V (0N20V) | 30/11/2017 |
| 7 | S32K118 Mask Set Errata for Mask 0N97V (0N97V) | 26/02/2018 |

# Chapter 3
# Driver

## 3.1 Requirements

Requirements for this driver are detailed in the AUTOSAR 4.2 Rev0002Wdg Driver Software Specification document (See Table Reference List ).

## 3.2 Driver Design Summary

The Watchdog Timer (WDOG) module is an independent timer that is available for system use. It provides a safety feature to ensure that software is executing as planned and that the CPU is not stuck in an infinite loop or executing unintended code. If the WDOG module is not serviced (refreshed) within a certain period, it resets the MCU.

Features of the WDOG module include:

• Configurable clock source inputs independent from the bus clock

• Programmable timeout period

• Optional fixed 256 clock prescaler when longer timeout periods are needed

• Robust write sequence for counter refresh

• Refresh sequence of writing 0xA602 and then 0xB480 within 16 bus clocks

• Window mode option for the refresh mechanism

• Optional timeout interrupt to allow post-processing diagnostics

• Configuration bits are write-once-after-reset to ensure watchdog configuration cannot be mistakenly altered.

• Robust write sequence for unlocking write-once configuration bits

## 3.3 Driver Limitations

None.

## 3.4 Driver Usage and Configuration Tips

1.Configure the WDG reference clock from MCU (see parameter WdgClkSrcRef ) according to reference point used by the WDG hardware on the platform. Example: If WDG is clocked by SIRC, then the MCU reference clock must be SIRC.

2.Configure WDG routine used for triggering as a GPT callback (Wdg_Cbk_GptNotificationX must be configured as a notification callback for the GPT channel intended for triggering)

3.If there are multiple WDG hardware instances on the platform, the API names will expand according to AUTOSAR requirement BSW00347. For example, if there are instances 0,1 and 2 available on the hardware,then the name of the init functions will be Wdg_43_Instance0_Init, Wdg_43_Instance1_Init and Wdg_43_Instance2_Init instead of Wdg_Init().

## 3.5 Requirements

The WDG driver uses the WDOG hardware IP.

## 3.6 Deviation from Requirements

The driver deviates from the AUTOSAR Wdg Driver software specification in some places.

There are also some additional requirements (on top of requirements detailed in AUTOSAR Wdg Driver software specification) which need to be satisfied for correct operation.

**Table 3-1. Deviations Status Column Description**

| Term | Definition |
|---|---|
| N/A | Not available |

*Table continues on the next page...*

**Table 3-1. Deviations Status Column Description (continued)**

| Term | Definition |
|---|---|
| N/T | Not testable |
| N/S | Out of scope |
| N/I | Not implemented |
| N/F | Not fully implemented |
| N/R | Unclear Requirement |
| N/V | Not Verifiable |

Below table identifies the AUTOSAR requirements that are not fully implemented, implemented differently, or out of scope for the driver.

**Table 3-2. Driver Deviations Table**

| Requirement | Status | Description | Notes |
|---|---|---|---|
| SWS_Wdg_00034 | N/A | General design rules: The start address of the watchdog trigger routine shall be statically configurable to a fixed memory location by the user. The user needs to take care that Configured memory location is valid for the platform on which driver is being implemented on.This configuration parameter shall only be given if supported/needed by the hardware. Rationale: This allows the watchdog device to identify the correct trigger input if supported by the hardware. | Not supported by hardware |
| SWS_Wdg_00055 | N/A | The Wdg module for an external watchdog driver shall have source code that is independent of the microcontroller platform. | External module is customer dependant and is not developed, and therefore not in scope. External assumption - "An external device, acting as supervisor of the operations, must provide a watchdog to cover common cause of failures of Leopard_1M for SIL3 applications". The implementation for WDG external must be supported by a dedicated Wdg module |
| SWS_Wdg_00076 | N/S | External watchdog driver To access the external watchdog hardware, the Wdg module shall use the functionality and API of the corresponding handler or driver, e.g. the SPI handler or DIO driver. | External module is customer dependant and is not yet developed. |
| SWS_Wdg_00077 | N/S | External watchdog driver - A Wdg module for an external watchdog shall satisfy the same functional requirements and offer the same functional scope as a Wdg module for an internal watchdog. Hence their respective APIs are semantically identical. | External module is customer dependant and is not yet developed. |
| SWS_Wdg_00078 | N/S | External watchdog driver - The Wdg module shall add all parameters required for accessing the external watchdog hardware, e.g. the used SPI channel or DIO port, to the module's published parameters and to the module's configuration parameters. | External module is customer dependant and is not yet developed. |

*Table continues on the next page...*

## Table 3-2. Driver Deviations Table (continued)

| Requirement | Status | Description | Notes |
|---|---|---|---|
| ECUC_Wdg_00112 | N/A | Container Name WdgExternalConfiguration Description Configuration items for an external watchdog hardware | External module is customer dependant and is not developed, and therefore not in scope. External assumption - "An external device, acting as supervisor of the operations, must provide a watchdog to cover common cause of failures of Leopard_1M for SIL3 applications". The implementation for WDG external must be supported by a dedicated Wdg module |
| ECUC_Wdg_00113 | N/A | Name WdgExternalContainerRef Description Reference to either a DioChannelGroup container in case the hardware watchdog is connected via DIO pins - an SpiSequenceConfiguration container in case the watchdog hardware is accessed via SPI Multiplicity 0..1 Type Choice reference to [ DioChannelGroup , SpiSequence ] Post-Build Variant Multiplicity true Post-Build Variant Value true Multiplicity Configuration Class Pre-compile time, Link time, Post-build time Value Configuration Class Pre-compile time, Link time, Post-build time Scope / Dependency scope: local dependency: See DIO resp. SPI SWS | External module is customer dependant and is not developed, and therefore not in scope. External assumption - "An external device, acting as supervisor of the operations, must provide a watchdog to cover common cause of failures of Leopard_1M for SIL3 applications". The implementation for WDG external must be supported by a dedicated Wdg module |
| ECUC_Wdg_00118 | N/A | Wdg_Configuration - Location (memory address) of the watchdog trigger routine. Dependancy: Only relevant if provided by hardware and needed by the system. | This paramater functionality is replaced by PR-MCAL-3268.wdg. |
| SWS_Wdg_00152 | N/I | The internal state of the module (which indicates whether it is not initialized, idle or busy) shall be available for debugging. | The internal variables should not be accesible outside the driver - independent of the purpose. |
| SWS_Wdg_00153 | N/I | The internal variable for the watchdog timeout counter shall be available for debugging. | The internal variables should not be accesible outside the driver - independent of the purpose. |
| SWS_Wdg_00154 | N/I | The internal variable for the watchdog mode shall be available for debugging. | Avoid global variables or else justify their usage. |
| SWS_Wdg_00162 | N/A | The routine servicing an external watchdog shall be implemented by usage of an own internal hardware timer to be independent from other peripherals or by using a GPT driver callback ? | External module is customer dependant and is not developed, and therefore not in scope. External assumption - "An external device, acting as supervisor of the operations, must provide a watchdog to cover common cause of failures of Leopard_1M for SIL3 applications". The implementation for WDG external must be supported by a dedicated Wdg module |
| SWS_Wdg_00172 | N/A | If more than one watchdog driver instance exits on an ECU (namely an external and an internal one) the API names and instance specific type names specified in this chapter shall be made unique by expansion according to SRS_BSW_00347. | External module is customer dependant and is not developed, and therefore not in scope. External assumption - "An external device, acting as supervisor of the operations, must provide a watchdog to cover common cause of failures of Leopard_1M for SIL3 applications". The implementation for WDG external must be supported by a dedicated Wdg module |

## 3.7 Runtime Errors

The driver generates the following DEM errors at runtime.

**Table 3-3. Runtime Errors**

| Function | Error Code | Condition triggering the error |
|---|---|---|
| Wdg_Init | WDG_E_DISABLE_REJECTED | Initialization or mode switch failed because it would disable the watchdog" has occurred |
| Wdg_SetMode | WDG_E_DISABLE_REJECTED | Initialization or mode switch failed because it would disable the watchdog" has occurred |
| Wdg_Init | WDG_E_MODE_FAILED | Setting a watchdog mode failed (during initialization or mode switch)" has occurred |
| Wdg_SetMode | WDG_E_MODE_FAILED | Setting a watchdog mode failed (during initialization or mode switch)" has occurred |

## 3.8 Software specification

The following sections contains driver software specifications.

### 3.8.1 Define Reference

Constants supported by the driver are as per AUTOSAR Wdg Driver software specification Version 4.2 Rev0002 .

#### 3.8.1.1 Define WDG_AR_RELEASE_MAJOR_VERSION

**Table 3-4. Define WDG_AR_RELEASE_MAJOR_VERSION Description**

| | |
|---|---|
| **Name** | WDG_AR_RELEASE_MAJOR_VERSION |
| **Initializer** | 4 |

### 3.8.1.2 Define WDG_AR_RELEASE_MINOR_VERSION

**Table 3-5. Define WDG_AR_RELEASE_MINOR_VERSION Description**

| | |
|---|---|
| **Name** | WDG_AR_RELEASE_MINOR_VERSION |
| **Initializer** | 2 |

### 3.8.1.3 Define WDG_AR_RELEASE_REVISION_VERSION

**Violates:** MISRA 2004 Rule 1.4, The compiler/linker shall be checked to ensure that 31 character significance and case sensitivity are supported for external identifiers.

**Table 3-6. Define WDG_AR_RELEASE_REVISION_VERSION Description**

| | |
|---|---|
| **Name** | WDG_AR_RELEASE_REVISION_VERSION |
| **Initializer** | 2 |

### 3.8.1.4 Define WDG_MODULE_ID

**Table 3-7. Define WDG_MODULE_ID Description**

| | |
|---|---|
| **Name** | WDG_MODULE_ID |
| **Initializer** | 102 |

### 3.8.1.5 Define WDG_SW_MAJOR_VERSION

**Table 3-8. Define WDG_SW_MAJOR_VERSION Description**

| | |
|---|---|
| **Name** | WDG_SW_MAJOR_VERSION |
| **Initializer** | 1 |

### 3.8.1.6 Define WDG_SW_MINOR_VERSION

**Table 3-9. Define WDG_SW_MINOR_VERSION Description**

| | |
|---|---|
| **Name** | WDG_SW_MINOR_VERSION |
| **Initializer** | 0 |

### 3.8.1.7 Define WDG_SW_PATCH_VERSION

**Table 3-10. Define WDG_SW_PATCH_VERSION Description**

| | |
|---|---|
| **Name** | WDG_SW_PATCH_VERSION |
| **Initializer** | 1 |

### 3.8.1.8 Define WDG_VENDOR_ID

**Table 3-11. Define WDG_VENDOR_ID Description**

| | |
|---|---|
| **Name** | WDG_VENDOR_ID |
| **Initializer** | 43 |

### 3.8.1.9 Define WDG_DEV_ERROR_DETECT

Compile switch to enable/disable development error detection for this module.

**Table 3-12. Define WDG_DEV_ERROR_DETECT Description**

| | |
|---|---|
| **Name** | WDG_DEV_ERROR_DETECT |
| **Initializer** | STD_ON |

### 3.8.1.10 Define WDG_DISABLE_ALLOWED

Compile switch to allow/forbid disabling the watchdog driver during runtime.

**Table 3-13. Define WDG_DISABLE_ALLOWED Description**

| | |
|---|---|
| **Name** | WDG_DISABLE_ALLOWED |

*Table continues on the next page...*

**Table 3-13. Define WDG_DISABLE_ALLOWED Description (continued)**

| | |
|---|---|
| **Initializer** | STD_OFF |

### 3.8.1.11 Define WDG_PRECOMPILE_SUPPORT

**Table 3-14. Define WDG_PRECOMPILE_SUPPORT Description**

| | |
|---|---|
| **Name** | WDG_PRECOMPILE_SUPPORT |
| **Initializer** | STD_OFF |

### 3.8.1.12 Define WDG_LINKTIME_SUPPORT

**Table 3-15. Define WDG_LINKTIME_SUPPORT Description**

| | |
|---|---|
| **Name** | WDG_LINKTIME_SUPPORT |
| **Initializer** | STD_OFF |

### 3.8.1.13 Define WDG_POSTBUILD_SUPPORT

**Table 3-16. Define WDG_POSTBUILD_SUPPORT Description**

| | |
|---|---|
| **Name** | WDG_POSTBUILD_SUPPORT |
| **Initializer** | STD_OFF |

### 3.8.1.14 Define WDG_ROM

This variable will indicate RAM/ROM execution.

**Table 3-17. Define WDG_ROM Description**

| | |
|---|---|
| **Name** | WDG_ROM |
| **Initializer** | |

### 3.8.1.15 Define WDG_VERSION_INFO_API

Compile switch to enable/disable the version information.

**Table 3-18. Define WDG_VERSION_INFO_API Description**

| | |
|---|---|
| **Name** | WDG_VERSION_INFO_API |
| **Initializer** | STD_ON |

### 3.8.1.16 Define WDG_DISABLE_DEM_REPORT_ERROR_STATUS

Compile switch enable / disable Diagnostic Event Manager for this module.

**Table 3-19. Define WDG_DISABLE_DEM_REPORT_ERROR_STATUS Description**

| | |
|---|---|
| **Name** | WDG_DISABLE_DEM_REPORT_ERROR_STATUS |
| **Initializer** | STD_OFF |

## 3.8.2 Enum Reference

Enumeration of all constants supported by the driver are as per AUTOSAR Wdg Driver software specification Version 4.2 Rev0002 .

### 3.8.2.1 Enumeration Wdg_ErrorIdType

Indicates the aditional det errors used by the watchdog driver.

**Table 3-20. Enumeration Wdg_ErrorIdType Values**

| Name | Initializer | Description |
|---|---|---|
| WDG_E_DRIVER_STATE | 0x10 | Type of error: API service used in wrong context (e.g. module not initialized). |
| WDG_E_PARAM_MODE | 0x11 | Type of error:API service called with wrong / inconsistent parameter(s). |
| WDG_E_PARAM_CONFIG | 0x12 | Type of error:API service called with wrong / inconsistent parameter(s). |
| WDG_E_PARAM_TIMEOUT | 0x13 | Type of error: The passed timeout value is higher than the maximum timeout value. |
| WDG_E_PARAM_POINTER | 0x14 | Type of error: API is called with wrong pointer value (e.g. NULL pointer). |
| WDG_E_INIT_FAILED | 0x15 | Type of error: Invalid configuration set selection. |

### 3.8.2.2 Enumeration Wdg_ServiceIdType

This enumerated type will contain the service ids for the watchodg functions.

**Pre:** To define WDG_GETVERSION_ID, WDG_VERSION_INFO_API has to be equal to STD_ON.

**Table 3-21. Enumeration Wdg_ServiceIdType Values**

| Name | Initializer | Description |
|---|---|---|
| WDG_GETVERSION_ID | 0x04 | The service id for the Wdg_GetVersion function. |
| WDG_INIT_ID | 0x00 | The service id for the Wdg_Init function. |
| WDG_SETMODE_ID | 0x01 | The service id for the Wdg_SetMode function. |
| WDG_SETTRIGGERCONDITION_ID | 0x02 | The service id for the Wdg_SetTriggerCondition function. |
| WDG_TRIGGER_ID | 0x03 | The service id for the Wdg_Trigger function. |

## 3.8.3 Function Reference

Functions of all functions supported by the driver are as per AUTOSAR Wdg Driver software specification Version 4.2 Rev0002 .

### 3.8.3.1 Function Wdg_GetVersionInfo

Returns the version information of the module.

**Details:**

The Wdg_ChannelGetVersionInfo function shall return the version information of this module. The version information includes:

- Module Id,
- Vendor Id,
- Vendor specific version numbers.

**Pre:** This function is only required if the WDG_VERSION_INFO_API has to be equal STD_ON.

**Implements:** Wdg_ChannelGetVersionInfo_Activity

**Prototype:** `void Wdg_GetVersionInfo(pVersioninfo);`

**Table 3-22. Wdg_GetVersionInfo Arguments**

| Type | Name | Direction | Description |
|---|---|---|---|
| | pVersioninfo | **input, output** | Pointer to where to store the version information of this module. |

### 3.8.3.2 Function Wdg_Init

Include Memory mapping specification.

**Details:**

The Wdg_Init function shall initialize the Wdg module and the watchdog hardware, i.e. it shall set the default watchdog mode and timeout period as provided in the configuration set.

**Violates:** MISRA 2004 Required Rule 19.15 precautions to prevent the contents of a header file being included twice This function initializes the WDG module.

**Implements:** Wdg_Init_Activity

**Prototype:** `void Wdg_Init(const Wdg_ConfigType *pConfigPtr);`

**Table 3-23. Wdg_Init Arguments**

| Type | Name | Direction | Description |
|---|---|---|---|
| | ConfigPtr | **input** | Pointer to configuration set. |

### 3.8.3.3 Function Wdg_SetMode

Switches the watchdog into the mode Mode.

**Details:**

By choosing one of a limited number of statically configured settings (e.g. toggle or window watchdog, different timeout periods) the Wdg module and the watchdog hardware can be switched between the following three different watchdog modes using the Wdg_SetModefunction:

- WDGIF_OFF_MODE,
- WDGIF_SLOW_MODE,
- WDGIF_FAST_MODE.

**Return:** Std_ReturnType.

**Implements:** Wdg_SetMode_Activity

**Prototype:** `void Wdg_SetMode(const WdgIf_ModeType Mode);`

**Table 3-24. Wdg_SetMode Arguments**

| Type | Name | Direction | Description |
|---|---|---|---|
| const WdgIf_ModeType | Mode | **input** | One of the following statically configured modes: WDGIF_OFF_MODE, WDGIF_SLOW_MODE, WDGIF_FAST_MODE. |

**Table 3-25. Wdg_SetMode Return Values**

| Name | Description |
|---|---|
| E_OK | Mode switch executed completely and successfully. |
| E_NOT_OK | The mode switch encountered errors. |

### 3.8.3.4 Function Wdg_SetTriggerCondition

Reset the watchdog timeout counter according to the timeout value passed.

**Implements:** Wdg_SetTriggerCondition_Activity

**Prototype:** `void Wdg_SetTriggerCondition(const uint16 u16Timeout);`

**Table 3-26. Wdg_SetTriggerCondition Arguments**

| Type | Name | Direction | Description |
|---|---|---|---|
| | Timeout | **input** | Value (milliseconds) for setting the trigger counter. |

### 3.8.3.5 Function Wdg_Cbk_GptNotification0

This function performs the triggering of the watchdog.In order to have a functional WDG module it is mandatory to configure this API as a GPT callback notification.

There are two configurations needed:

1. The definition of a GPT channel(please see configuration parameter WdgExternalTriggerCounterRef)

2. In GPT configuration set Wdg_Cbk_GptNotification0 as the GptNotification for the respective channel

**Violates:** MISRA 2004 Required Rule 19.1 , only preprocessor statements and comments before 'include'

**Violates:** MISRA 2004 Required Rule 19.15 precautions to prevent the contents of a header file being included twice

**Violates:** MISRA 2004 Required Rule 8.10, All declarations and definitions of objects or functions at file scope shall have internal linkage unless external linkage is required.

**Prototype:** `void Wdg_Cbk_GptNotification0(void);`

#### 3.8.3.6 Function Wdg_Wdog0_Isr

This function process the interrupt Wdog0.

**Details:**

This function process the Wdog0 interrupt

**Prototype:** `void Wdg_Wdog0_Isr(void);`

### 3.8.4 Structs Reference

Data structures supported by the driver are as per AUTOSAR Wdg Driver software specification Version 4.2 Rev0002 .

#### 3.8.4.1 Structure Wdg_ConfigType

Defines the configuration structure.

**Declaration:**

```
typedef struct
                        {
```

```
    const WdgIf_ModeType Wdg_DefaultMode;
            const Wdg_IPW_InstanceType Wdg_Instance;
            const Gpt_ChannelType Wdg_TimerChannel;
            const uint32 Wdg_u32TriggerSourceClock;
            const Wdg_IPW_ConfigType* Wdg_ModeSettings[3];
            Wdg_CallbackPtrType Wdg_CallbackPtr;
} Wdg_ConfigType;
```

**Table 3-27. Structure Wdg_ConfigType member description**

| Member | Description |
|---|---|
| Wdg_DefaultMode | The number of configured channels. |
| Wdg_Instance | The instance id. |
| Wdg_TimerChannel | Gpt Channel configured. |
| Wdg_u32TriggerSourceClock | The frequency of the configured timer channel. |
| Wdg_ModeSettings | Pointer to Watchdog Specific implementation details. |
| Wdg_CallbackPtr | Pointer to callback notification. |

### 3.8.5 Types Reference

Types supported by the driver are as per AUTOSAR Wdg Driver software specification Version 4.2 Rev0002 .

#### 3.8.5.1 Typedef Wdg_CallbackPtrType

**Type:** void(*

## 3.9 Symbolic Names Disclaimer

All containers having the symbolic name tag set as true in the Autosar schema will generate defines like:

#define <Container_Short_Name> <Container_ID>

For this reason it is forbidden to duplicate the name of such containers across the MCAL configuration, or to use names that may trigger other compile issues (e.g. match existing #ifdefs arguments).

# Chapter 4
# Tresos Configuration Plug-in

This chapter describes the Tresos configuration plug-in for the Wdg Driver. The most of the parameters are described below.

## 4.1 Configuration elements of Wdg

**Included forms :**

- IMPLEMENTATION_CONFIG_VARIANT
- WdgDemEventParameterRefs
- WdgGeneral
- WdgPublishedInformation
- CommonPublishedInformation
- WdgClockReferencePoint
- WdgSettingsConfig

**Table 4-1. Revision table**

| Revision | Date |
|---|---|
| 4.1.0 | 2010-12-03 |

## 4.2 Form IMPLEMENTATION_CONFIG_VARIANT

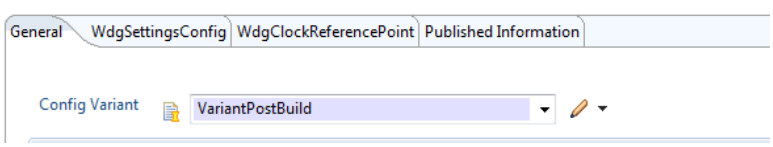


**Figure 4-1. Tresos Plugin snapshot for IMPLEMENTATION_CONFIG_VARIANT form.**

**Table 4-2. Attribute IMPLEMENTATION_CONFIG_VARIANT detailed description**

| Property | Value |
|---|---|
| Label | Config Variant |
| Type | ENUMERATION |
| Default | VariantLinkTime |
| Range | VariantLinkTime<br>VariantPostBuild<br>VariantPreCompile |

# 4.3 Form WdgDemEventParameterRefs

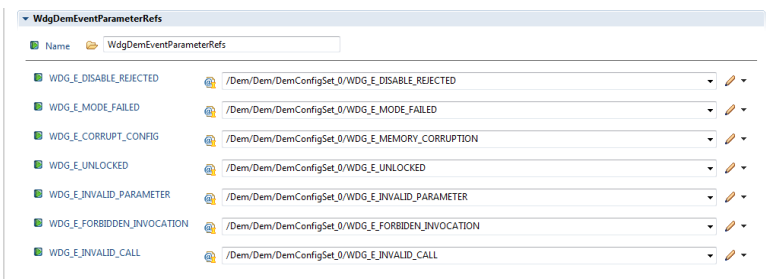


**Figure 4-2. Tresos Plugin snapshot for WdgDemEventParameterRefs form.**

## 4.3.1 WDG_E_DISABLE_REJECTED (WdgDemEventParameterRefs)

**Table 4-3. Attribute WDG_E_DISABLE_REJECTED (WdgDemEventParameterRefs) detailed description**

| Property | Value |
|---|---|
| Type | SYMBOLIC-NAME-REFERENCE |
| Origin | AUTOSAR_ECUC |
| Enable | true |

### 4.3.2 WDG_E_MODE_FAILED (WdgDemEventParameterRefs)

**Table 4-4. Attribute WDG_E_MODE_FAILED (WdgDemEventParameterRefs) detailed description**

| Property | Value |
|---|---|
| Type | SYMBOLIC-NAME-REFERENCE |
| Origin | AUTOSAR_ECUC |
| Enable | true |

### 4.3.3 WDG_E_CORRUPT_CONFIG (WdgDemEventParameterRefs)

**Table 4-5. Attribute WDG_E_CORRUPT_CONFIG (WdgDemEventParameterRefs) detailed description**

| Property | Value |
|---|---|
| Type | SYMBOLIC-NAME-REFERENCE |
| Origin | Custom |
| Enable | true |

### 4.3.4 WDG_E_UNLOCKED (WdgDemEventParameterRefs)

**Table 4-6. Attribute WDG_E_UNLOCKED (WdgDemEventParameterRefs) detailed description**

| Property | Value |
|---|---|
| Type | SYMBOLIC-NAME-REFERENCE |
| Origin | Custom |
| Enable | true |

### 4.3.5 WDG_E_INVALID_PARAMETER (WdgDemEventParameterRefs)

**Table 4-7. Attribute WDG_E_INVALID_PARAMETER (WdgDemEventParameterRefs) detailed description**

| Property | Value |
|---|---|
| Type | SYMBOLIC-NAME-REFERENCE |
| Origin | Custom |
| Enable | true |

### 4.3.6 WDG_E_FORBIDDEN_INVOCATION (WdgDemEventParameterRefs)

**Table 4-8. Attribute WDG_E_FORBIDDEN_INVOCATION (WdgDemEventParameterRefs) detailed description**

| Property | Value |
|---|---|
| Type | SYMBOLIC-NAME-REFERENCE |
| Origin | Custom |
| Enable | true |

### 4.3.7 WDG_E_INVALID_CALL (WdgDemEventParameterRefs)

**Table 4-9. Attribute WDG_E_INVALID_CALL (WdgDemEventParameterRefs) detailed description**

| Property | Value |
|---|---|
| Type | SYMBOLIC-NAME-REFERENCE |
| Origin | Custom |
| Enable | true |

## 4.4 Form WdgGeneral

### WdgGeneral

All general parameters of the watchdog driver are collected here.

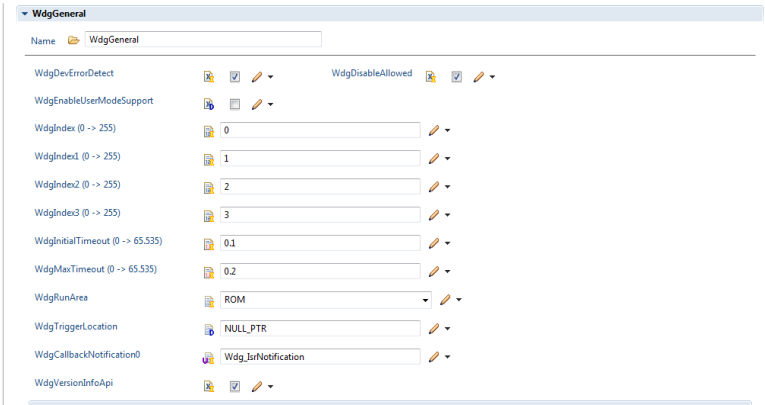


**Figure 4-3. Tresos Plugin snapshot for WdgGeneral form.**

## 4.4.1 WdgDisableDemReportErrorStatus (WdgGeneral)

### Wdg Disable Production Error Reporting

Enable/Disable Dem error reporting. **True**: Dem error reporting enabled **False**: Dem error reporting disabled

**Table 4-10. Attribute Wdg Disable Production Error Reporting (WdgGeneral) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Disable Production Error Reporting |
| Type | BOOLEAN |
| Origin | AUTOSAR_ECUC |
| Symbolic Name | false |
| Default | false |

## 4.4.2 WdgDevErrorDetect (WdgGeneral)

### Wdg Development Error Detect

Compile switch to enable / disable development error detection for this module. **True**: Development error detection enabled **False**: Development error detection disabled

**Table 4-11. Attribute WdgDevErrorDetect (WdgGeneral) detailed description**

| Property | Value |
|---|---|
| Label | Development Error Detection |
| Type | BOOLEAN |

*Table continues on the next page...*

**Table 4-11. Attribute WdgDevErrorDetect (WdgGeneral) detailed description (continued)**

| Property | Value |
|---|---|
| Origin | AUTOSAR_ECUC |
| Symbolic Name | false |
| Default | true |

## 4.4.3 WdgDisableAllowed (WdgGeneral)

### Wdg Disable Allowed

Compile switch to allow / forbid disabling the watchdog driver during runtime. **True**: Disabling the watchdog driver at runtime is allowed **False**:Disabling the watchdog driver at runtime is not allowed

**Table 4-12. Attribute WdgDisableAllowed (WdgGeneral) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Disable Allowed |
| Type | BOOLEAN |
| Origin | AUTOSAR_ECUC |
| Symbolic Name | false |
| Default | false |

## 4.4.4 WdgIndex (WdgGeneral)

### Wdg Instance 0 Index

Represents the watchdog driver's ID so that it can be referenced by the watchdog interface.

**Table 4-13. Attribute WdgIndex (WdgGeneral) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Instance 0 Index |
| Type | INTEGER |
| Origin | AUTOSAR_ECUC |
| Symbolic Name | true |
| Default | 0 |
| Invalid | Range<br><=255<br>>=0 |

## 4.4.5 WdgInitialTimeout (WdgGeneral)

### Wdg Initial Timeout

The initial timeout (sec) for the trigger condition to be initialized during Init function. It shall be not larger than WdgMaxTimeout.

**Table 4-14. Attribute WdgInitialTimeout (WdgGeneral) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Initial Timeout [s] |
| Type | FLOAT |
| Origin | AUTOSAR_ECUC |
| Symbolic Name | false |
| Default | 0 |
| Invalid | Range<br><=65.535<br>>=0 |

## 4.4.6 WdgMaxTimeout (WdgGeneral)

### Wdg Max Timeout

The maximum timeout (miliseconds) to which the watchdog trigger condition can be initialized.

**Table 4-15. Attribute WdgMaxTimeout (WdgGeneral) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Max Timeout [s] |
| Type | FLOAT |
| Origin | AUTOSAR_ECUC |
| Symbolic Name | false |
| Default | 0 |
| Invalid | Range<br><=65.535<br>>=0 |

## 4.4.7 WdgRunArea (WdgGeneral)

### Wdg Run Area

Represents the watchdog driver execution area is either from ROM(Flash) or RAM as required with the particular microcontroller.

**Table 4-16. Attribute WdgRunArea (WdgGeneral) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Run Area |
| Type | ENUMERATION |
| Origin | AUTOSAR_ECUC |
| Symbolic Name | false |
| Default | ROM |
| Range | RAM<br>ROM |

## 4.4.8 WdgTriggerLocation (WdgGeneral)

### Wdg Trigger Location

Location (memory address) of the watchdog trigger routine.

**Note**

Not supported by the current hardware.

**Table 4-17. Attribute WdgTriggerLocation (WdgGeneral) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Trigger Location |
| Type | FUNCTION-NAME |
| Origin | AUTOSAR_ECUC |
| Symbolic Name | false |
| Default | NULL_PTR |
| Enable | false |

## 4.4.9 WdgCallbackNotification0 (WdgGeneral)

### WdgCallbackNotification0

Callback notification for the ISR Wdg_Wdog0_Isr function

**Table 4-18. Attribute WdgCallbackNotification0 (WdgGeneral) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Instance 0 CallBack Notification |
| Type | FUNCTION-NAME |
| Origin | Custom |
| Symbolic Name | false |
| Default | NULL_PTR |

## 4.4.10 WdgVersionInfoApi (WdgGeneral)

### Wdg VersionInfo Api

Compile switch to enable / disable the version information API. **True**: API enabled **False**:API disabled

**Table 4-19. Attribute WdgVersionInfoApi (WdgGeneral) detailed description**

| Property | Value |
|---|---|
| Label | Provide Version Info API |
| Type | BOOLEAN |
| Origin | AUTOSAR_ECUC |
| Symbolic Name | false |
| Default | true |

# 4.5 Form WdgPublishedInformation

### WdgPublishedInformation

Container holding all Wdg specific published information parameters


**Figure 4-4. Tresos Plugin snapshot for WdgPublishedInformation form.**

## 4.5.1 WdgTriggerMode (WdgPublishedInformation)

### Wdg Trigger Mode

Watchdog trigger mode (toggle/window/both).

**Table 4-20. Attribute WdgTriggerMode (WdgPublishedInformation) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Trigger Mode |
| Type | ENUMERATION_LABEL |
| Origin | AUTOSAR_ECUC |
| Symbolic Name | false |
| Default | WDG_BOTH |
| Range | WDG_BOTH<br>WDG_TOGGLE<br>WDG_WINDOW |

## 4.6 Form CommonPublishedInformation

Common container, aggregated by all modules. It contains published information about vendor and versions.

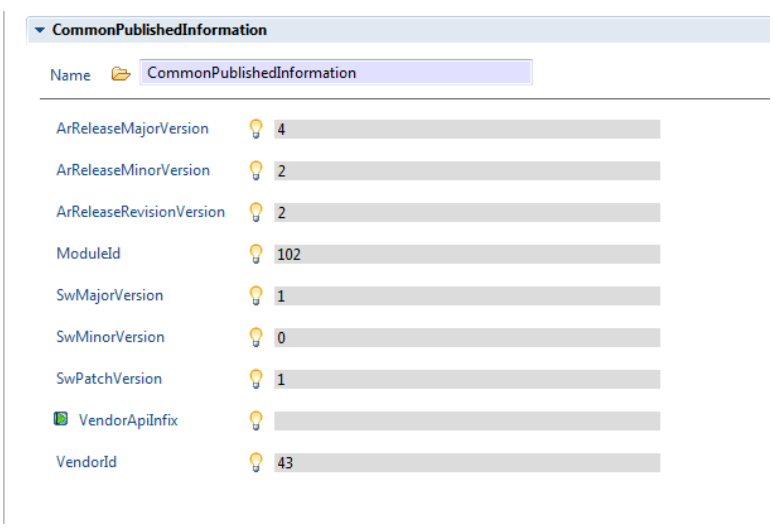


**Figure 4-5. Tresos Plugin snapshot for CommonPublishedInformation form.**

### 4.6.1 ArReleaseMajorVersion (CommonPublishedInformation)

Major version number of AUTOSAR specification on which the appropriate implementation is based on.

**Table 4-21. Attribute ArReleaseMajorVersion (CommonPublishedInformation) detailed description**

| Property | Value |
|---|---|
| Label | AUTOSAR Major Version |
| Type | INTEGER_LABEL |
| Origin | Custom |
| Symbolic Name | false |
| Default | 4 |
| Invalid | Range<br>>=4<br><=4 |

## 4.6.2 ArReleaseMinorVersion (CommonPublishedInformation)

Minor version number of AUTOSAR specification on which the appropriate implementation is based on.

**Table 4-22. Attribute ArReleaseMinorVersion (CommonPublishedInformation) detailed description**

| Property | Value |
|---|---|
| Label | AUTOSAR Minor Version |
| Type | INTEGER_LABEL |
| Origin | Custom |
| Symbolic Name | false |
| Default | 2 |
| Invalid | Range<br>>=2<br><=2 |

## 4.6.3 ArReleaseRevisionVersion (CommonPublishedInformation)

Revision version number of AUTOSAR specification on which the appropriate implementation is based on.

**Table 4-23. Attribute ArReleaseRevisionVersion (CommonPublishedInformation) detailed description**

| Property | Value |
|---|---|
| Label | AUTOSAR Release Revision Version |
| Type | INTEGER_LABEL |

*Table continues on the next page...*

**Table 4-23. Attribute ArReleaseRevisionVersion (CommonPublishedInformation) detailed description (continued)**

| Property | Value |
|---|---|
| Origin | Custom |
| Symbolic Name | false |
| Default | 2 |
| Invalid | Range<br>>=2<br><=2 |

## 4.6.4 ModuleId (CommonPublishedInformation)

Module ID of this module from Module List.

**Table 4-24. Attribute ModuleId (CommonPublishedInformation) detailed description**

| Property | Value |
|---|---|
| Label | Module Id |
| Type | INTEGER_LABEL |
| Origin | Custom |
| Symbolic Name | false |
| Default | 102 |
| Invalid | Range<br>>=102<br><=102 |

## 4.6.5 SwMajorVersion (CommonPublishedInformation)

Major version number of the vendor specific implementation of the module. The numbering is vendor specific.

**Table 4-25. Attribute SwMajorVersion (CommonPublishedInformation) detailed description**

| Property | Value |
|---|---|
| Label | Software Major Version |
| Type | INTEGER_LABEL |
| Origin | Custom |
| Symbolic Name | false |
| Default | 1 |
| Invalid | Range<br>>=1<br><=1 |

### 4.6.6 SwMinorVersion (CommonPublishedInformation)

Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.

**Table 4-26. Attribute SwMinorVersion (CommonPublishedInformation) detailed description**

| Property | Value |
|---|---|
| Label | Software Minor Version |
| Type | INTEGER_LABEL |
| Origin | Custom |
| Symbolic Name | false |
| Default | 0 |
| Invalid | Range<br>>=0<br><=0 |

### 4.6.7 SwPatchVersion (CommonPublishedInformation)

Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.

**Table 4-27. Attribute SwPatchVersion (CommonPublishedInformation) detailed description**

| Property | Value |
|---|---|
| Label | Software Patch Version |
| Type | INTEGER_LABEL |
| Origin | Custom |
| Symbolic Name | false |
| Default | 1 |
| Invalid | Range<br>>=1<br><=1 |

### 4.6.8 VendorApiInfix (CommonPublishedInformation)

In driver modules which can be instantiated several times on a single ECU, BSW00347 requires that the name of APIs is extended by the VendorId and a vendor specific name. This parameter is used to specify the vendor specific name. In total, the implementation

specific name is generated as follows: <ModuleName>_>VendorId>_<VendorApiInfix><Api name from SWS>. E.g. assuming that the VendorId of the implementor is 123 and the implementer chose a VendorApiInfix of "v11r456" a api name Can_Write defined in the SWS will translate to Can_123_v11r456Write. This parameter is mandatory for all modules with upper multiplicity > 1. It shall not be used for modules with upper multiplicity =1.

**Table 4-28. Attribute VendorApiInfix (CommonPublishedInformation) detailed description**

| Property | Value |
|---|---|
| Label | Vendor Api Infix |
| Type | STRING_LABEL |
| Origin | Custom |
| Symbolic Name | false |
| Default | |
| Enable | false |

### 4.6.9 VendorId (CommonPublishedInformation)

Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list.

**Table 4-29. Attribute VendorId (CommonPublishedInformation) detailed description**

| Property | Value |
|---|---|
| Label | Vendor Id |
| Type | INTEGER_LABEL |
| Origin | Custom |
| Symbolic Name | false |
| Default | 43 |
| Invalid | Range<br>>=43<br><=43 |

## 4.7 Form WdgClockReferencePoint

### WdgClockReferencePoint

This container contains a parameter, which represents a reference to a container of the type McuClockReferencePoint (defined in module MCU).

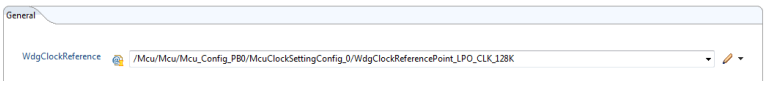


**Figure 4-6. Tresos Plugin snapshot for WdgClockReferencePoint form.**

## 4.7.1 WdgClockReference (WdgClockReferencePoint)

### WdgClockReference

Reference to a container of the type McuClockReferencePoint, to select an input clock.

**Table 4-30. Attribute WdgClockReference (WdgClockReferencePoint) detailed description**

| Property | Value |
|---|---|
| Label | WdgClockReference |
| Type | REFERENCE |
| Origin | Custom |
| Symbolic Name | false |

# 4.8 Form WdgSettingsConfig

### WdgSettingsConfig

Configuration items for the different watchdog settings, including those for external watchdog hardware.

### Included forms :

- Form WdgExternalConfiguration
- Form WdgSettingsFast
- Form WdgSettingsOff
- Form WdgSettingsSlow

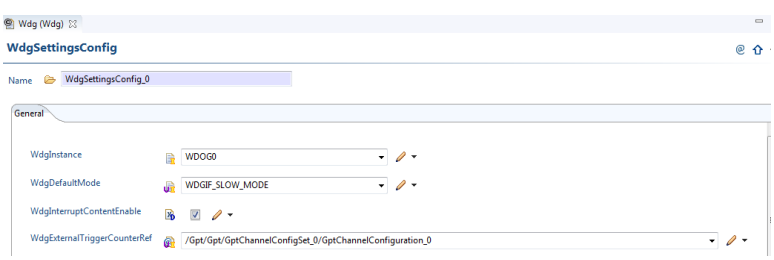


**Figure 4-7. Tresos Plugin snapshot for WdgSettingsConfig form.**

## 4.8.1 WdgInstance (WdgSettingsConfig)

### Wdg Hardware Instance

Select specific hardware instance for watchdog driver initialization.

**Table 4-31. Attribute WdgInstance (WdgSettingsConfig) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Hardware Instance |
| Type | ENUMERATION |
| Origin | Custom |
| Symbolic Name | false |

## 4.8.2 WdgDefaultMode (WdgSettingsConfig)

### Wdg Default Mode

Default mode for watchdog driver initialization.

**Table 4-32. Attribute WdgDefaultMode (WdgSettingsConfig) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Default Mode |
| Type | ENUMERATION |
| Origin | AUTOSAR_ECUC |
| Symbolic Name | false |
| Default | WDGIF_SLOW_MODE |
| Range | WDGIF_FAST_MODE<br>WDGIF_OFF_MODE<br>WDGIF_SLOW_MODE |

## 4.8.3 WdgExternalTriggerCounterRef (WdgSettingsConfig)

### Wdg External Trigger Counter

Reference to the GptChannel configuration which set for the watchdog servicing routine implementation.

**Table 4-33. Attribute WdgExternalTriggerCounterRef (WdgSettingsConfig) detailed description**

| Property | Value |
|---|---|
| Label | Wdg External Trigger Counter |
| Type | CHOICE-REFERENCE |
| Origin | Custom |

## 4.8.4 WdgInterruptContentEnable (WdgSettingsConfig)

### Wdg Interrupt Enable

This parametter is used to generate interrupt content for each SWT. True = Interrupt content is generated. False = Interrupt content is not generated.

**Table 4-34. Attribute WdgExternalTriggerCounterRef (WdgSettingsConfig) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Interrupt Enable |
| Type | BOOLEAN |
| Origin | Custom |
| Default | true |

## 4.8.5 Form WdgExternalConfiguration

### WdgExternalConfiguration

Configuration items for an external watchdog hardware

**Is included by form :** Form WdgSettingsConfig


**Figure 4-8. Tresos Plugin snapshot for WdgExternalConfiguration form.**

### 4.8.5.1 WdgExternalContainerRef (WdgExternalConfiguration)

### WdgExternalContainerRef

Reference to either **- a DioChannelGroup** container in case the hardware watchdog is connected via DIO pins **- a SpiSequenceConfiguration** container in case the watchdog hardware is accessed via SPI

**Note**

This parameter is not used by current implementation

**Table 4-35. Attribute WdgExternalContainerRef (WdgExternalConfiguration) detailed description**

| Property | Value |
|---|---|
| Type | CHOICE-REFERENCE |
| Origin | AUTOSAR_ECUC |
| Enable | false |

## 4.8.6 Form WdgSettingsFast

### WdgSettingsFast

Hardware dependent settings for the watchdog driver's fast mode.

**Is included by form :** Form WdgSettingsConfig

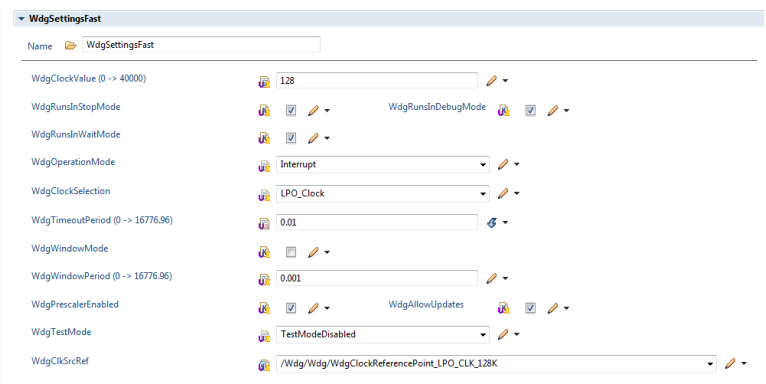


**Figure 4-9. Tresos Plugin snapshot for WdgSettingsFast form.**

## 4.8.6.1 WdgClockValue

### Wdg Clock Value

This is the Implementation Specific parameter. Indicates Wdg Clock Value in KHz

**Table 4-36. Attribute WdgClockValue detailed description**

| Property | Value |
|---|---|
| Label | Wdg Clock Value[KHz] |
| Type | INTEGER |
| Origin | Custom |
| Symbolic Name | false |
| Default | 40000 |
| Invalid | Range<br><=40000<br>>0 |

### 4.8.6.2 WdgClkSrcRef

**WdgClkSrcRef**

Reference to the WdgClockReferencePoint from which the clock is derived

**Table 4-37. WdgClkSrcRef detailed description**

| Property | Value |
|---|---|
| Label | WdgClkSrcRef |
| Type | REFERENCE |
| Origin | Custom |
| Symbolic Name | false |

### 4.8.6.3 WdgRunsInStopMode (WdgSettingsFast)

**Wdg Runs in Stop Mode**

This is the Implementation Specific parameter.

- Enabled: WDOG continues to count even while the processor core is in stop mode.
- Disabled: WDOG stops counting if the processor core is in stop mode.

**Note**

The 'WdgRunsInStopMode' parameter specifies if the watchdog timer should run or not while the clock to the core is halted. This is true only for the STOP0 mode of the controller. It will always run while the controller is in the HALT0 mode.

**Table 4-38. Attribute WdgRunsInStopMode (WdgSettingsFast) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Runs In Stop mode |
| Type | BOOLEAN |
| Origin | Custom |
| Symbolic Name | false |
| Default | true |

## 4.8.6.4 WdgRunsInDebugMode (WdgSettingsFast)

### Wdg Runs In Debug Mode

This is the Implementation Specific parameter.

- Enabled: WDOG continues to count even while the device enters the debug mode.
- Disabled: WDOG stops counting if the processor core when the device enters the debug mode

**Table 4-39. Attribute WdgRunsInDebugMode (WdgSettingsFast) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Runs In Debug Mode |
| Type | BOOLEAN |
| Origin | Custom |
| Symbolic Name | false |
| Default | false |

## 4.8.6.5 WdgRunsInWaitmode (WdgSettingsFast)

### Wdg Runs In Wait Mode

This is the Implementation Specific parameter.

- Enabled: WDOG continues to count even while the device enters the wait mode.
- Disabled: WDOG stops counting if the processor core when the device enters the wait mode

**Table 4-40. Attribute WdgRunsInDebugMode (WdgSettingsFast) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Runs In Wait Mode |
| Type | BOOLEAN |

*Table continues on the next page...*

**Table 4-40. Attribute WdgRunsInDebugMode (WdgSettingsFast) detailed description (continued)**

| Property | Value |
|---|---|
| Origin | Custom |
| Symbolic Name | false |
| Default | false |

## 4.8.6.6 WdgOperationMode (WdgSettingsFast)

### Wdg Operation Mode

This is the Implementation Specific parameter.

- ResetOnTimeOut: Generate a reset on a time-out.
- Interrupt: Generate an interrupt on an initial time-out, reset on a second consecutive time-out.

**Table 4-41. Attribute WdgOperationMode (WdgSettingsFast) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Operation Mode |
| Type | ENUMERATION |
| Origin | Custom |
| Symbolic Name | false |
| Default | ResetOnTimeOut |
| Range | ResetOnTimeOut<br>Interrupt |

## 4.8.6.7 WdgClockSelection (WdgSettingsFast)

### Wdg Clock Selection

WDOG clock selection.

**Table 4-42. Attribute WdgClockSelection (WdgSettingsFast) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Clock Selection |
| Type | ENUMERATION |
| Origin | Custom |
| Symbolic Name | false |

*Table continues on the next page...*

**Table 4-42. Attribute WdgClockSelection (WdgSettingsFast) detailed description (continued)**

| Property | Value |
|---|---|
| Default | Bus_Clock |
| Range | Bus_Clock<br>LPO_Clock<br>SOSC_Clock<br>SIRC_Clock |

## 4.8.6.8 WdgTimeoutPeriod (WdgSettingsFast)

### Wdg Timeout Period

This is the Implementation Specific parameter. Software Watchdog Time-Out Period in seconds. Selects the time-out period for the WDOG.

**Table 4-43. Attribute WdgTimeoutPeriod (WdgSettingsFast) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Timeout Period [s] |
| Type | FLOAT |
| Origin | Custom |
| Symbolic Name | false |
| Default | 0 |
| Invalid | Range<br><=4294967295<br>>=0 |

## 4.8.6.9 WdgWindowMode (WdgSettingsFast)

### Wdg WindowMode.

- Disabled: Regular mode, service sequence can be done at any time.
- Enabled : Windowed mode, the service sequence is only valid when the up counter is greater than value in the WDOG_WIN register.

**Table 4-44. Attribute WdgWindowMode (WdgSettingsFast) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Window Mode |

*Table continues on the next page...*

**Table 4-44. Attribute WdgWindowMode (WdgSettingsFast) detailed description (continued)**

| Property | Value |
|---|---|
| Type | BOOLEAN |
| Origin | Custom |
| Symbolic Name | false |
| Default | false |

## 4.8.6.10 WdgWindowPeriod (WdgSettingsFast)

### Wdg Window Period

This is the Implementation Specific parameter. Window start value. When window mode is enabled, the service sequence can only be written when the internal down counter is less than this value.

**Table 4-45. Attribute WdgWindowPeriod (WdgSettingsFast) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Window Period[s] |
| Type | FLOAT |
| Origin | Custom |
| Symbolic Name | false |
| Default | 0 |
| Invalid | Range<br><=268.4354559375<br>>=0 |

## 4.8.6.11 WdgPrescalerEnabled (WdgSettingsFast)

### Wdg Prescaler Enabled

Disabled: Prescaler Disabled. Enabled : Prescaler Enabled.

**Table 4-46. Attribute WdgPrescalerEnabled (WdgSettingsFast) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Prescaler Enabled |
| Type | BOOLEAN |
| Origin | Custom |
| Symbolic Name | false |
| Default | 1 |

### 4.8.6.12 WdgAllowUpdates (WdgSettingsFast)

**Wdg Allow Updates**

Disabled: Updates not allowed. Enabled : Updates allowed.

Note:Due to the 128 bus clocks requirement for reconfiguring the watchdog, some delays must be inserted before executing STOP or WAIT instructions after reconfiguring the watchdog. This ensures that the watchdog's new configuration takes effect before the MCU enters low power mode. Otherwise, the MCU may not be waken up from low power mode.

**Table 4-47. Attribute WdgAllowUpdates (WdgSettingsFast) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Allow Updates |
| Type | BOOLEAN |
| Origin | Custom |
| Symbolic Name | false |
| Default | 1 |

### 4.8.6.13 WdgTestMode (WdgSettingsFast)

**Wdg Test Mode**

TestModeDisabled: Watchdog test mode disabled. UserModeEnabled : Watchdog user mode enabled. TestModeLowByte: Watchdog test mode enabled, only the low byte is used. TestModeHighByte : Watchdog test mode enabled, only the high byte is used.

**Table 4-48. Attribute WdgTestMode (WdgSettingsFast) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Clock Selection |
| Type | ENUMERATION |
| Origin | Custom |
| Symbolic Name | false |
| Default | TestModeDisabled |
| Range | TestModeDisabled<br>UserModeEnabled<br>TestModeLowByte<br>TestModeHighByte |

## 4.8.7 Form WdgSettingsOff

### WdgSettingsOff

Hardware dependent settings for the watchdog driver's off mode.

**Is included by form :** Form WdgSettingsConfig


**Figure 4-10. Tresos Plugin snapshot for WdgSettingsOff form.**

### 4.8.7.1 Wdg Allow Updates (WdgSettingsOff)

#### WdgAllowUpdates

Disabled: Updates not allowed.

Enabled : Updates allowed.

**Table 4-49. Attribute WdgSoftLockConfiguration (WdgSettingsOff) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Allow Updates |
| Type | BOOLEAN |
| Origin | Custom |
| Symbolic Name | false |
| Default | false |
| Enable | true |

## 4.8.8 Form WdgSettingsSlow

### WdgSettingsSlow

Hardware dependent settings for the watchdog driver's fast mode.

**Is included by form :** Form WdgSettingsConfig

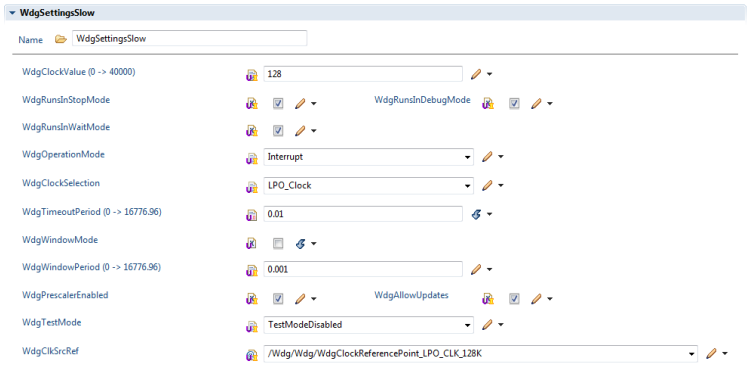


**Figure 4-11. Tresos Plugin snapshot for WdgSettingsSlow form.**

### 4.8.8.1 WdgClockValue

#### Wdg Clock Value

This is the Implementation Specific parameter. Indicates Wdg Clock Value in KHz

**Table 4-50. Attribute WdgClockValue detailed description**

| Property | Value |
|---|---|
| Label | Wdg Clock Value[KHz] |
| Type | INTEGER |
| Origin | Custom |
| Symbolic Name | false |
| Default | 40000 |
| Invalid | Range<br><=40000<br>>0 |

### 4.8.8.2 WdgClkSrcRef

#### WdgClkSrcRef

Reference to the WdgClockReferencePoint from which the clock is derived

**Table 4-51. WdgClkSrcRef detailed description**

| Property | Value |
|---|---|
| Label | WdgClkSrcRef |
| Type | REFERENCE |
| Origin | Custom |
| Symbolic Name | false |

### 4.8.8.3 WdgRunsInStopMode (WdgSettingsSlow)

**Wdg Runs in Stop Mode**

This is the Implementation Specific parameter.

- Enabled: WDOG continues to count even while the processor core is in stop mode.
- Disabled: WDOG stops counting if the processor core is in stop mode.

**Note**

The 'WdgRunsInStopMode' parameter specifies if the watchdog timer should run or not while the clock to the core is halted. This is true only for the STOP0 mode of the controller. It will always run while the controller is in the HALT0 mode.

**Table 4-52. Attribute WdgRunsInStopMode (WdgSettingsSlow) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Runs In Stop mode |
| Type | BOOLEAN |
| Origin | Custom |
| Symbolic Name | false |
| Default | true |

### 4.8.8.4 WdgRunsInDebugMode (WdgSettingsSlow)

**Wdg Runs In Debug Mode**

This is the Implementation Specific parameter.

- Enabled: WDOG continues to count even while the device enters the debug mode.
- Disabled: WDOG stops counting if the processor core when the device enters the debug mode

**Table 4-53. Attribute WdgRunsInDebugMode (WdgSettingsSlow) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Runs In Debug Mode |
| Type | BOOLEAN |
| Origin | Custom |
| Symbolic Name | false |
| Default | false |

### 4.8.8.5 WdgRunsInWaitmode (WdgSettingsSlow)

#### Wdg Runs In Wait Mode

This is the Implementation Specific parameter.

- Enabled: WDOG continues to count even while the device enters the wait mode.
- Disabled: WDOG stops counting if the processor core when the device enters the wait mode

**Table 4-54. Attribute WdgRunsInDebugMode (WdgSettingsSlow) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Runs In Wait Mode |
| Type | BOOLEAN |
| Origin | Custom |
| Symbolic Name | false |
| Default | false |

### 4.8.8.6 WdgOperationMode (WdgSettingsSlow)

#### Wdg Operation Mode

This is the Implementation Specific parameter.

- ResetOnTimeOut: Generate a reset on a time-out.
- Interrupt: Generate an interrupt on an initial time-out, reset on a second consecutive time-out.

**Table 4-55. Attribute WdgOperationMode (WdgSettingsSlow) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Operation Mode |
| Type | ENUMERATION |
| Origin | Custom |
| Symbolic Name | false |
| Default | ResetOnTimeOut |
| Range | ResetOnTimeOut<br>Interrupt |

### 4.8.8.7 WdgClockSelection (WdgSettingsSlow)

**Wdg Clock Selection**

WDOG clock selection.

**Table 4-56. Attribute WdgClockSelection (WdgSettingsSlow) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Clock Selection |
| Type | ENUMERATION |
| Origin | Custom |
| Symbolic Name | false |
| Default | Bus_Clock |
| Range | Bus_Clock<br>LPO_Clock<br>SOSC_Clock<br>SIRC_Clock |

### 4.8.8.8 WdgTimeoutPeriod (WdgSettingsSlow)

**Wdg Timeout Period**

This is the Implementation Specific parameter. Software Watchdog Time-Out Period in seconds. Selects the time-out period for the WDOG.

**Table 4-57. Attribute WdgTimeoutPeriod (WdgSettingsSlow) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Timeout Period [s] |
| Type | FLOAT |
| Origin | Custom |
| Symbolic Name | false |
| Default | 0 |
| Invalid | Range<br><=4294967295<br>>=0 |

### 4.8.8.9 WdgWindowMode (WdgSettingsSlow)

**Wdg WindowMode.**

- Disabled: Regular mode, service sequence can be done at any time.
- Enabled : Windowed mode, the service sequence is only valid when the up counter is greater than value in the WDOG_WIN register.

**Table 4-58. Attribute WdgWindowMode (WdgSettingsSlow) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Window Mode |
| Type | BOOLEAN |
| Origin | Custom |
| Symbolic Name | false |
| Default | false |

## 4.8.8.10 WdgWindowPeriod (WdgSettingsSlow)

### Wdg Window Period

This is the Implementation Specific parameter. Window start value. When window mode is enabled, the service sequence can only be written when the internal down counter is less than this value.

**Table 4-59. Attribute WdgWindowPeriod (WdgSettingsSlow) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Window Period[s] |
| Type | FLOAT |
| Origin | Custom |
| Symbolic Name | false |
| Default | 0 |
| Invalid | Range<br><=268.4354559375<br>>=0 |

## 4.8.8.11 WdgPrescalerEnabled (WdgSettingsSlow)

### Wdg Prescaler Enabled

Disabled: Prescaler Disabled. Enabled : Prescaler Enabled.

**Table 4-60. Attribute WdgPrescalerEnabled (WdgSettingsSlow) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Prescaler Enabled |
| Type | BOOLEAN |
| Origin | Custom |
| Symbolic Name | false |
| Default | 1 |

### 4.8.8.12 WdgAllowUpdates (WdgSettingsSlow)

**Wdg Allow Updates**

Disabled: Updates not allowed. Enabled : Updates allowed.

Note:Due to the 128 bus clocks requirement for reconfiguring the watchdog, some delays must be inserted before executing STOP or WAIT instructions after reconfiguring the watchdog. This ensures that the watchdog's new configuration takes effect before the MCU enters low power mode. Otherwise, the MCU may not be waken up from low power mode.

**Table 4-61. Attribute WdgAllowUpdates (WdgSettingsSlow) detailed description**

| Property | Value |
|---|---|
| Label | Wdg Allow Updates |
| Type | BOOLEAN |
| Origin | Custom |
| Symbolic Name | false |
| Default | 1 |

### 4.8.8.13 WdgTestMode (WdgSettingsSlow)

**Wdg Test Mode**

TestModeDisabled: Watchdog test mode disabled. UserModeEnabled : Watchdog user mode enabled. TestModeLowByte: Watchdog test mode enabled, only the low byte is used. TestModeHighByte : Watchdog test mode enabled, only the high byte is used.

## Table 4-62. Attribute WdgTestMode (WdgSettingsSlow) detailed description

| Property | Value |
|---|---|
| Label | Wdg Clock Selection |
| Type | ENUMERATION |
| Origin | Custom |
| Symbolic Name | false |
| Default | TestModeDisabled |
| Range | TestModeDisabled<br>UserModeEnabled<br>TestModeLowByte<br>TestModeHighByte |

# Chapter 5
# How to Configure

This chapter describes the configure Tresos plug-in for the Wdg Driver.

## 5.1 Watchdog Settings Config

This chapter describle how to configure a channel in EB Tresos plug-in WdgSettingsConfig_0

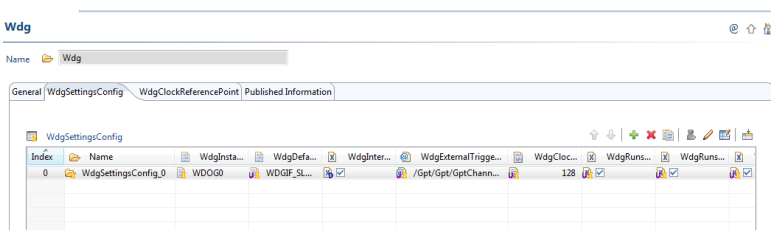


**Figure 5-1. Select Watchdog Config Set.**

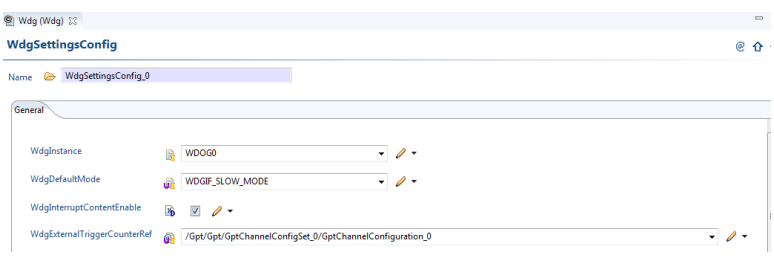


**Figure 5-2. Config Watchdog Settings General.**


**Figure 5-3. Watchdog External Config.**

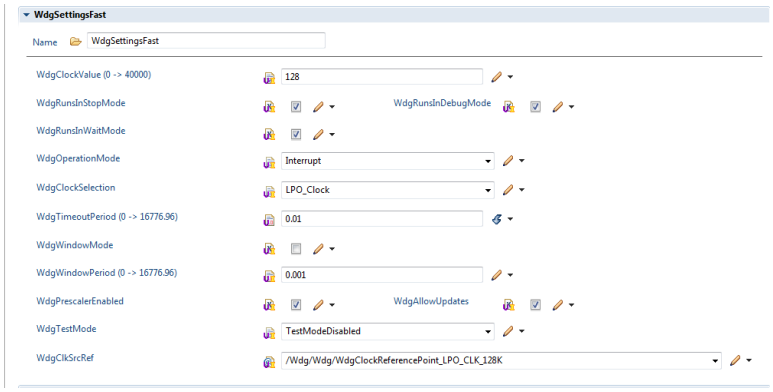


**Figure 5-4. Watchdog Settings.**

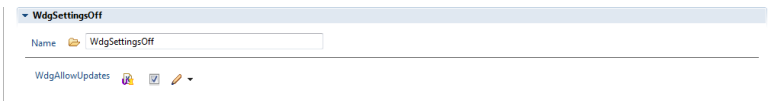


**Figure 5-5. Watchdog Settings Off.**

***How to Reach Us:***

**Home Page:**
nxp.com

**Web Support:**
nxp.com/support



Document Number UM2WDGASR4.2 Rev0002 R1.0.1
Revision 1.0

AUTOSAR