Use AR3 SWCs in AR4

Version 1.02.00
2015-01-16
Application NoteAN-ISC-8-1170

Author(s)	Alexander Zeeb
Restrictions	Customer confidential - Vector decides
Abstract	This application note describes how to use SWC developed for AR3 in pure AR4 environment.

Table of Contents

1.0 Overview ... 2
2.0 WdgM Wrapper Service Component for AR3 Legacy SWC ... 2
2.1 DaVinci Developer ... 2
2.2 DaVinci Configurator Pro ... 3
2.3 Mode Port WdgM_GlobalMode / WdgM_IndividualMode ... 3
3.0 NVM Wrapper Service Component for AR3 Legacy SWC ... 3
3.1 DaVinci Developer ... 4
3.1.1 Port: NvMAdministration ... 4
3.1.2 Port: NvMService ... 4
4.0 Adaptation for ECUM and COMM ... 4
4.1 ECUM ... 4
4.2 COMM ... 4
5.0 Additional Resources ... 4
6.0 Contacts ... 4

1.0 Overview

This application note describes what has to be done to integrate AR3 SWC in an AR4 environment (BSW) with as little adaptations as possible. Adaptations are mainly necessary if the AR3 SWC has dependencies to the interfaces of the modules **NVM** and **WDGM**. These modules in contrast to **ECUM** and **COMM** do not provide an AR3 legacy interface and hence need a wrapper component which translates between AR3 and AR4 interfaces.

2.0 WdgM Wrapper Service Component for AR3 Legacy SWC

The wrapper has to be designed and implemented manually and is realized by a Service Component Type in DaVinci Developer. This wrapper for the AR4 WDGM Service Component has to translate between the AR4 Port Interfaces of WDGM and the AR3 Port Interfaces of the SWC.

Note:	The AR4 Port Interface WdgM_AliveSupervision has a different and reduced set of Operations compared to AR3. AR4 does not support the de-/activation of Supervised Entities any more. Instead of the de/activation of Supervised Entities separately like in AR3, WDGM modes have to be requested- and released by the wrapper to change the state of the Supervised Entities. The description of these modes however is not part of this document.

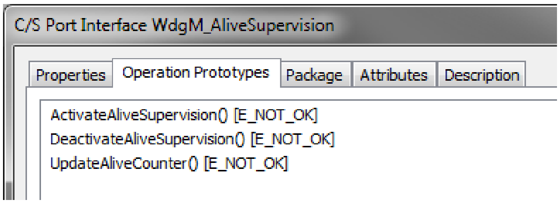
The wrapper component provides **twice** the set of Port Interfaces required by the AR3 legacy SWC:

- Compliant to AR3 (towards the SWC)
- Compliant to AR4 (towards the WDGM)

Both sets have to be connected appropriately in DaVinci Configurator Pro. All necessary steps are explained in the following chapters.

2.1 DaVinci Developer

- Create a **Service C/S Port Interface** WdgM_AliveSupervision with Operations according to AR3 like shown in the screenshot below.



ActivateAliveSupervision() [E_NOT_OK]
DeactivateAliveSupervision() [E_NOT_OK]
UpdateAliveCounter() [E_NOT_OK]

Figure 1 – AR3 C/S Port Interface WdgM_AliveSupervision

Create the **Service Component Type** of the wrapper component: **WdgM_AR3_Wrapper** and assign the following two Port Interfaces:

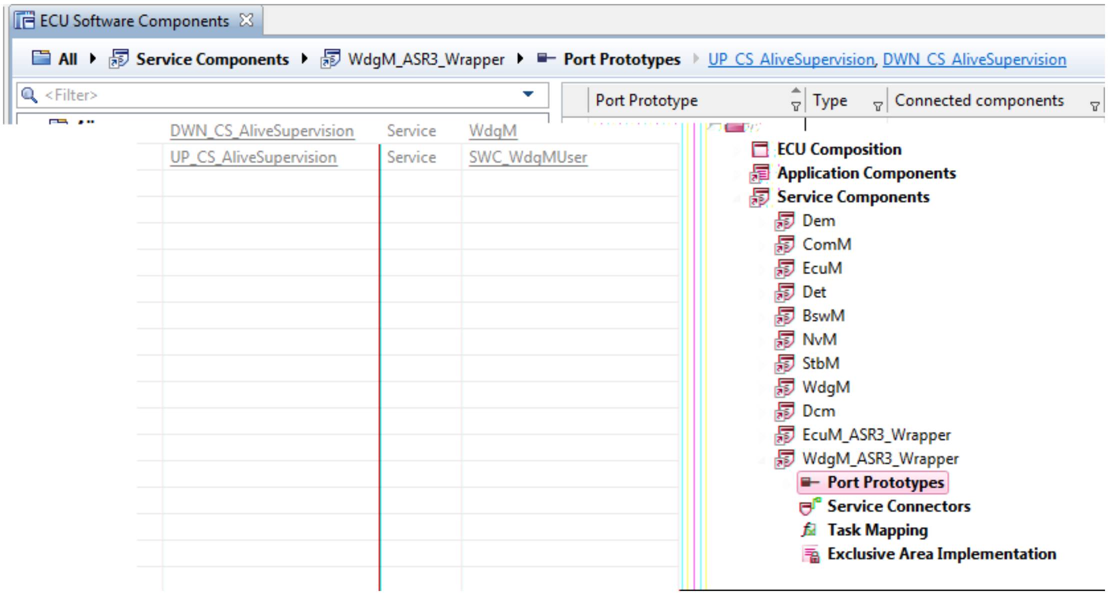
- WdgM_AliveSupervision (AR4) Client (DWN_CS_AliveSupervision)
- WdgM_AliveSupervision (AR3) Server (UP_CS_AliveSupervision)

Each Operation of the WdgM_AliveSupervision Server Port Prototype requires a Runnable, which is triggered on the corresponding Operation invocation. The Runnable of the Operation CheckpointReached can directly call the Operation UpdateAliveCounter.

Note: When using the **UpdateAliveCounter** interface, each call will trigger the report of a development error in case the development checks are activated within the WDGM. This is an AR feature used to notify the user that a deprecated functionality is used.

2.2 DaVinci Configurator Pro

The WDGM wrapper is connected between WDGM and application SWC. The SWC's remaining WDGM Interfaces are connected directly with the WDGM.



ECU Software Components

All ▸ Service Components ▸ WdgM_ASR3_Wrapper ▸ Port Prototypes ▸ UP_CS_AliveSupervision, DWN_CS_AliveSupervision

<Filter>

Port Prototype	Type	Connected components
DWN_CS_AliveSupervision	Service	WdgM
UP_CS_AliveSupervision	Service	SWC_WdgMUser

- ECU Composition
- Application Components
- Service Components
  - Dem
  - ComM
  - EcuM
  - Det
  - BswM
  - NvM
  - StbM
  - WdgM
  - Dcm
  - EcuM_ASR3_Wrapper
  - WdgM_ASR3_Wrapper
    - Port Prototypes
    - Service Connectors
    - Task Mapping
    - Exclusive Area Implementation

Figure 4 – Port Mapping in DaVinci Configurator Pro

2.3 Mode Port WdgM_GlobalMode / WdgM_IndividualMode

Both Mode Ports semantically describe the same modes in AR3 and AR4. The only difference is the name of those modes.

AR3	AR4	Value
ALIVE_OK	SUPERVISION_OK	0
ALIVE_FAILED	SUPERVISION_FAILED	1
ALIVE_EXPIRED	SUPERVISION_EXPIRED	2
ALIVE_STOPPED	SUPERVISION_STOPPED	3
ALIVE_DEACTIVATED	SUPERVISION_DEACTIVATED	4

Table 1 – Differences in Mode WdgM_GlobalMode and WdgM_IndividualMode

Since the modes are in the correct order, AR3- and AR4 Mode Ports are compatible from a functional point of view.

Note: The different naming can also be a reason for using the wrapper.

3.0 NVM Wrapper Service Component for AR3 Legacy SWC

The Port Interface of AR4 NVM is not compatible with Port Interface of AR3. Especially the return types have changed.

The wrapper has to be designed- and implemented manually and is realized by a Service Component Type in DaVinci Developer. This wrapper for the AR4 NVM Service Component translates between the AR4 Port Interfaces of NVM and the AR3 Port Interfaces of the SWC.

3.1 DaVinci Developer

3.1.1 Port: NvMAdministration

The only Operation of this Port Interface (`SetBlockProtection)` changed its return value from `POSSIBLE-ERRORS(<none>)` to `POSSIBLE-ERRORS(E_NOT_OK).`

To meet the AR4 return value, this definition can be adapted at the require-port instance of the AR3 SWC.

3.1.2 Port: NvMService

Return types (`POSSIBLE-ERRORS`) of following Operations changed:

- `GetErrorStatus`
- `GetDataIndex`
- `SetDataIndex`
- `SetRamBlockStatus`

Analog to NvMAdministration Port Interface, the SWC Port Interfaces have to be adapted to provide the new error codes.

Note: In contrast to AR3, NvM_SetDataIndex can return an error in AR4 which has to be handled by the application. This can only happen if ‘Development Error Detection’ is enabled.

4.0 Adaptation for ECUM and COMM

As mentioned in chapter 1.0, ECUM and COMM both provide legacy AR3 interfaces which which has to be activated. Further adaptations are not required..

4.1 ECUM

Configure fixed-behaviour of ECUM (refer to ECUM Technical Reference: **Support of EcuM fixed**).

4.2 COMM

Instantiate the parameter ‘**Operation GetInhibition Status Enabled’** and set this parameter to **FALSE** (/MICROSAR/ComM/ComMGeneral/ComMOperationGetInhibitionStatusEnabled).

5.0 Additional Resources

VECTOR TECHNICAL REFERENCES

TechnicalReference_Asr_EcuM.pdf	Technical Reference EcuM

6.0 Contacts

For a full list with all Vector locations and addresses worldwide, please visit http://vector.com/contact/.