Your systems.
Working as one.

RTI Connext

Integration Toolkit for AUTOSAR Classic

Getting Started Guide

Version 0.1

rti.com
blogs.rti.com
rtisoftware
connextpodcast
rti_software
rti_software

Real-Time Innovations International, Inc.
232 E. Java Drive, Sunnyvale, California 94089
Office +1 (408) 990-7400 • Fax +1 (408) 990-7402

Introduction

What is the Integration Toolkit for AUTOSAR Classic?

RTI Connext Integration Toolkit for AUTOSAR Classic is an experimental, complementary component of RTI Connext Micro and RTI Connext Cert. This Integration Toolkit allows you to convert data type definitions across standard formats (OMG IDL, OMG DDS-XML and AUTOSAR ARXML). It also generates supporting C code for data conversion and marshaling between the RTE and DDS communication frameworks.

This document is aimed at AUTOSAR ECU designers, implementers, and integrators already familiar with the vocabulary and concepts of the AUTOSAR Classic Platform, as well as basic DDS middleware concepts. Basic DDS concepts are covered in the RTI Connext Getting Started Guide.

This Integration Toolkit is meant to extend (not replace or change) the AUTOSAR Classic design methodology, starting with common ECU design practices (item 1 below) then complementing them (items 2-5 below):

1. **ECU design:** ECU Software Components (SW-C's) are modeled with already existing AUTOSAR Classic design methodologies and techniques, along with their communications ports and connectors. Data flows to and from DDS topics are to be represented by Send/Receive Port Prototypes in one or more purpose-specific or "gateway" Complex Device Driver (CDD) Software Components
2. **ARXML export:** these "gateway" Software Component definitions are exported to ARXML-formatted file(s)
3. **Generation:** *rtiarcgen*, this Integration Toolkit's command-line code generation tool, processes these files to produce:
    a. DDS-compatible representations of the data types used by the "gateway" SW-C interfaces in DDS-IDL or DDS-XML format, which can later be fed to rtiddsgen for generation of C-language type support code;
    b. Bidirectional C-language conversion routines between RTE and DDS frameworks;
    c. Bidirectional C-language marshaling routines between RTE and DDS frameworks;
    d. Interfaces for integration between RTE and DDS frameworks.

    NOTE: *rtiarcgen* is independent from *rtiddsgen*, RTI's code generation tool for *Connext Micro and Connext Professional*. *rtiarcgen* is used to derive DDS-related artifacts from AUTOSAR ARXML model exports, while *rtiddsgen* is used to produce type support code for DDS-IDL and DDS-XML type definitions. See the Usage section below for more details.
4. **Integration:** in this step, the interfaces described in item (d) of the previous step are implemented, providing:

Real-Time Innovations International, Inc.
232 E. Java Drive, Sunnyvale, California 94089
Office +1 (408) 990-7400 • Fax +1 (408) 990-7402

a. Project and ECU-specific DDS middleware initialization;
b. A project-specific DDS Entity model, defining Domain Participants, Topics, Publishers, Subscribers, DataWriters, and DataReaders that will be instantiated at run-time;
c. Project-specific mappings between RTE Port Prototype Interface Elements and DDS Endpoint Entities (DataReaders, DataWriters) defined in (b) above;

5. **Building, testing, iterating:** as the ECU design evolves in successive repetitions of step (1), the rest of the steps shall be incrementally executed again whenever DDS communication needs to change.

As shown in Figure 1, the result of this process is an AUTOSAR Classic ECU where Application and Complex Device Driver Software Components interact with each other through the RTE in normal AUTOSAR Classic fashion. Some of these Complex Device Driver Software Components may actually encapsulate data marshaling and interactions with Connext Micro or Cert, which in turn leverages the Communication Services to exercise DDS communication with other system components.

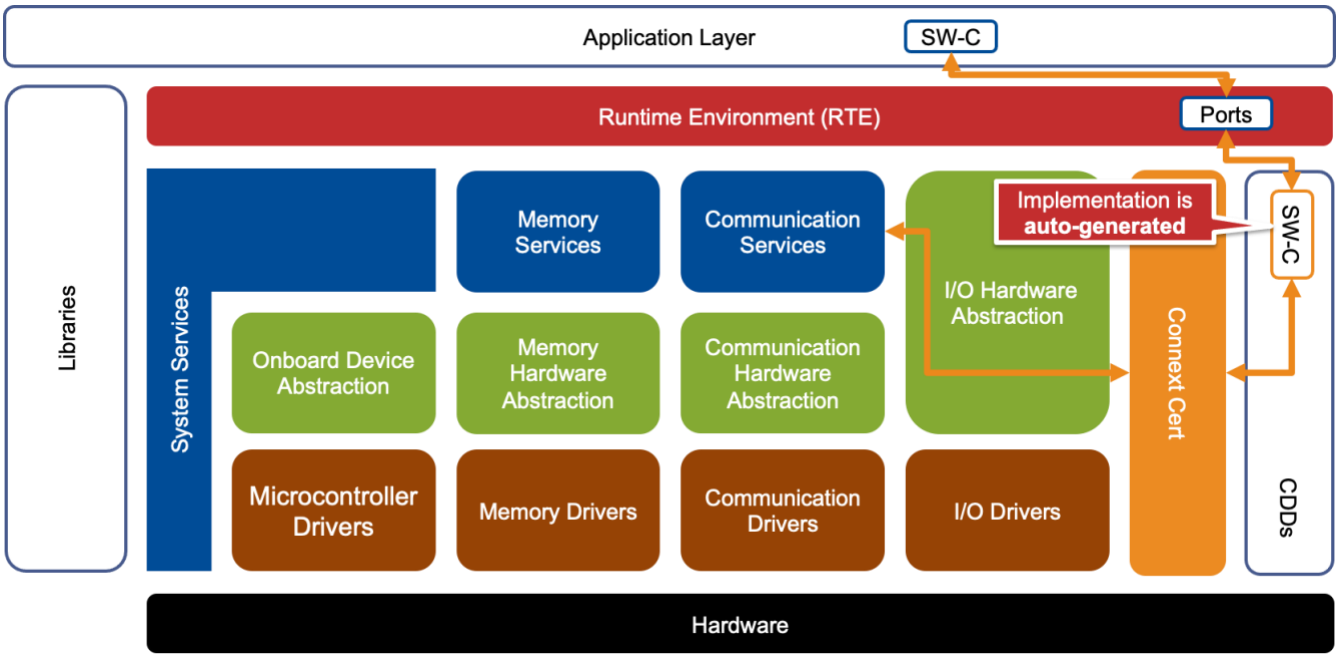

Figure 1: Example AUTOSAR Classic ECU architecture featuring DDS integration

Downloading and Installing the Integration Toolkit for AUTOSAR Classic

RTI Connext Integration Toolkit for AUTOSAR Classic is part of Connext Drive. For more information, see https://www.rti.com/drive.

Real-Time Innovations International, Inc.
232 E. Java Drive, Sunnyvale, California 94089
Office +1 (408) 990-7400 • Fax +1 (408) 990-7402

The AUTOSAR Classic Platform and Connext

The AUTOSAR Classic Platform provides processes, metamodels, and tooling for the joint specification, design, implementation, verification, validation, and deployment of Electronic Control Unit (ECU) software in the tiered automotive industry.

A key element of ECU modeling in AUTOSAR Classic is the Software Component (SW-C), whose intra- and inter-ECU interfaces with other software components are modeled in great detail before the actual functional behavior is implemented.

More information on the AUTOSAR Classic Platform can be found at https://www.autosar.org/standards/classic-platform/.

Connext Micro and Connext Cert are RTI's DDS middleware implementations that target constrained, real-time, and safety platforms, including AUTOSAR Classic OS-based platforms with UDP over IP support generally provided by the Socket Adapter (SoAd) and/or TCP/IP (TcpIp) Basic Software Modules. More information can be found at https://www.rti.com/products/connext-dds-micro and https://www.rti.com/products/connext-dds-cert.

Real-Time Innovations International, Inc.
232 E. Java Drive, Sunnyvale, California 94089
Office +1 (408) 990-7400 • Fax +1 (408) 990-7402

Features

Data description format conversion

ARXML to DDS-IDL

- Translation of AUTOSAR packages (AR-PACKAGES / AR-PACKAGE)
- Translation of base (SwBaseType) types categorized as FIXED_LENGTH with either:
  - A SHORT-NAME value in the following list:
    - boolean
    - [s|u]int[8|16|32|64]
    - float[32|64]
    - char
  - A BASE-TYPE-ENCODING value in the following list:
    - NONE (with BASE-TYPE-SIZE 8, 16, 32 or 64)
    - 2C (with BASE-TYPE-SIZE 8, 16, 32 or 64)
    - BOOLEAN
    - ISO-8859-1
- Translation of implementation (ImplementationDataType) types categorized as either:
  - VALUE (referencing a SwBaseType)
  - TYPE_REFERENCE
  - STRUCTURE (including recursive nesting of any supported type category)
  - ARRAY (including recursive nesting of any supported type category)
    - Note: only fixed-size array semantics are supported
- Translation of computation methods (CompuMethod) types categorized as TEXTTABLE
  - Containing a COMPU-SCALE defining COMPU-CONST and LOWER-LIMIT for each element
- Translation of STRUCTURE-based discriminated unions, as defined by AUTOSAR_SWS_SOMEIPTransformer (R21-11), SWS_SomeIpXf_00249
  - Containing an unsigned integer (FIXED_LENGTH/NONE) “memberSelector” field
  - Containing a UNION “payload” field

ARXML to DDS-XML

- Translation of AUTOSAR packages

Real-Time Innovations International, Inc.
232 E. Java Drive, Sunnyvale, California 94089
Office +1 (408) 990-7400 • Fax +1 (408) 990-7402

- Translation of base (SwBaseType) types as defined for DDS-IDL
- Translation of implementation (ImplementationDataType) types as defined for DDS-IDL
- Translation of computation methods (CompuMethod) types as defined for DDS-IDL
- Translation of STRUCTURE-based discriminated union types as defined for DDS-IDL

Data conversion code generation

ARXML to C

- Generation of bi-directional conversion routines between
  - DDS C runtime types, derived from the ARXML to DDS IDL/XML translations described above
  - RTE C runtime types, derived from the ARXML input of the translations described above

Runnable code generation

ARXML to C

- Generation of AUTOSAR Classic runnable implementations, for COMPLEX-DEVICE-DRIVER-SW-COMPONENT-TYPEs hosting RUNNABLE-ENTITY declarations with either
  - DATA-SEND-POINTs referencing P-PORT-PROTOTYPEs, implementing
    - DDS data sample reception from DataReader
    - DDS to RTE type conversion
    - RTE data sample write to port
  - DATA-RECEIVE-POINT-BY-ARGUMENTS referencing R-PORT-PROTOTYPE-REFs
    - RTE data sample read from port
    - RTE to DDS type conversion
    - DDS data sample write to DataWriter
- To broker data between the RTE and DDS frameworks, generated code for each Port Prototype Interface Element may:
  - Forward data samples by reference (zero cost) between RTE and DDS, as long as this optimization is enabled by the user, and the memory layout of the data types match for both frameworks
  - If this optimization is not enabled, or in-memory layout of DDS and RTE C-language types are not compatible, conversions defined in Data conversion code generation are used in generated code

DDS entity configuration and mapping generation

Real-Time Innovations International, Inc.
232 E. Java Drive, Sunnyvale, California 94089
Office +1 (408) 990-7400 • Fax +1 (408) 990-7402

- Generation of DDS C code for:
  - ECU-global initialization and finalization of the DDS Middleware
  - SW-C-specific initialization and finalization of DDS Entities
  - Mapping between DDS Entities and RTE Component Port Elements

Real-Time Innovations International, Inc.
232 E. Java Drive, Sunnyvale, California 94089
Office +1 (408) 990-7400 • Fax +1 (408) 990-7402

Usage

Overview

As seen in Figure 2, the Integration Toolkit for AUTOSAR Classic is comprised of a single application (*rti**arc**gen*) that ingests an AUTOSAR model export in ARXML format containing one or more Software Components (SW-Cs) of ComplexDeviceDriver (CDD) type, producing all the necessary artifacts to exercise the CDD SW-C interfaces over DDS:

- DDS-IDL and DDS-XML type definitions
- C-language bidirectional conversion routines between RTE and DDS types
- C-language bidirectional marshaling routines between RTE and DDS endpoint entities
- C-language DDS middleware/entity configuration, and RTE/DDS mapping routines

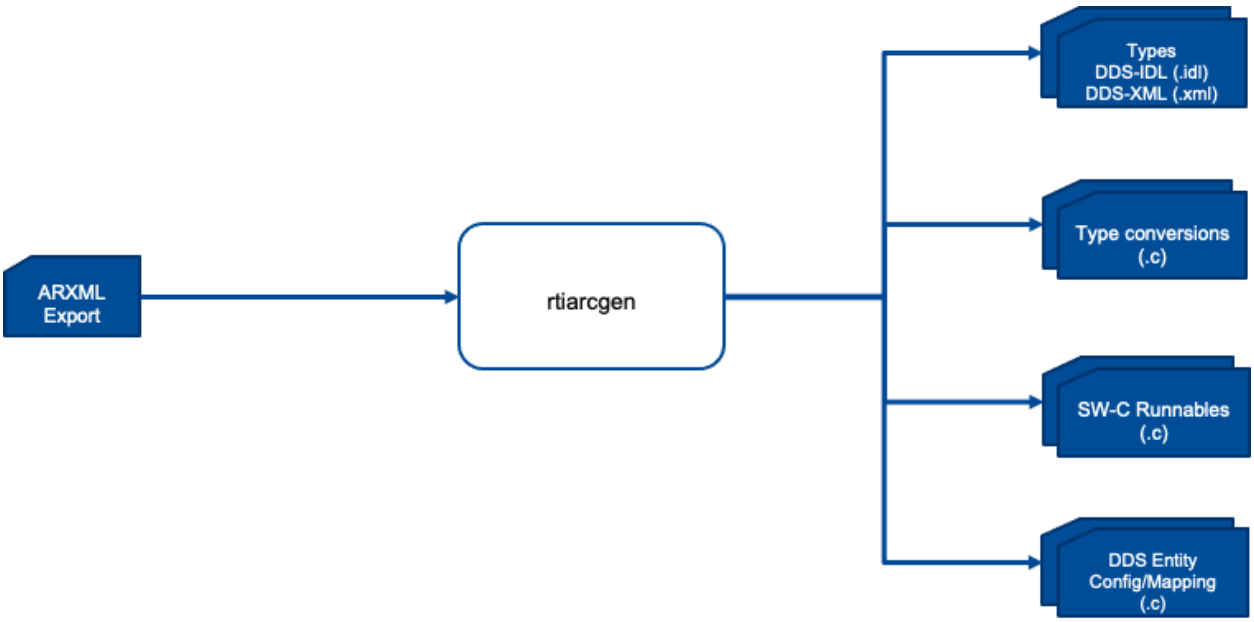

Figure 2: Integration Toolkit for AUTOSAR Classic workflow

DDS-IDL and DDS-XML outputs are meant to be processed by an DDS-IDL/DDS-XML type support code generator, like *rtiddsgen*, in order to produce C-language (.c and .h extensions) type support files that integrate with *Connext Micro* and the rest of generated C-language artifacts.

rtiarcgen is provided as a command-line executable that, once properly installed, shall be run as follows:

Real-Time Innovations International, Inc.
232 E. Java Drive, Sunnyvale, California 94089
Office +1 (408) 990-7400 • Fax +1 (408) 990-7402

```
$ rtiarcgen [options] <path-to-arxml-input-file>
```

rtiarcgen is located in the installation directory of RTI Connext Drive, with a platform-dependent launch script located in the "bin" subdirectory. You can launch the application by using its absolute path or, as a convenience, put the absolute path to the "bin" subdirectory in your PATH environment variable.

See the Features section for a detailed description of which AUTOSAR model elements can be parsed and transformed by the Integration Toolkit. The Command-line Parameters section describes the options you can use to control the Integration Toolkit's output.

Real-Time Innovations International, Inc.
232 E. Java Drive, Sunnyvale, California 94089
Office +1 (408) 990-7400 • Fax +1 (408) 990-7402

Command-line parameters for rtiarcgen

Global options

--help

Prints product version information, and a quick reference of all command-line parameters.

--idl-path=<path>

Configures the output path for DDS-IDL files (.idl extension) generated from the AUTOSAR model.

Default value: "."

--xml-path=<path>

Configures the output path for DDS-XML files (.xml extension) generated from the AUTOSAR model.

Default value: "."

--schema-path=<path>

Configures the DDS-XML schema path to include in DDS-XML files (.xml extension) generated from the AUTOSAR model, if any.

Default value: "" (no schema-related attributes in the top-level "types" element)

--c-header-path=<path>

Configures the output path for C language header files (.h extension) generated from the AUTOSAR model.

Default value: "."

--c-source-path=<path>

Configures the output path for C language source files (.c extension) generated from the AUTOSAR model.

Default value: "."

--single-file

Configures whether the different code generators should produce single files "amalgamating" elements of each AUTOSAR element kind (i.e., one file per data type, one file for data type conversion,

Real-Time Innovations International, Inc.
232 E. Java Drive, Sunnyvale, California 94089
Office +1 (408) 990-7400 • Fax +1 (408) 990-7402

etc), instead of one separate file per model element.

Please note that this option also references how runnable code files (.c extensions) reference type and conversion routine files (i.e., a single "#include" statement for all referenced elements, instead of one "#include" statement for each).

For example, suppose your AUTOSAR "model.arxml" ARXML file contains two supported Implementation data types, "Foo" and "Bar":

- With *--single-file*, the DDS-IDL code generator will produce a single "model_types.idl" file
- Without *--single-file*, the DDS-IDL code generator will produce separate "Foo.idl" and "Bar.idl" files

Default value: disabled

--verbose

Configures whether the different processors and code generators produce extra messages reflecting its inner workings, like what exact model elements are being found and processed at each stage.

Default value: disabled

--optimize

Configures whether the different C code generators will consider C-language memory layouts of both DDS and RTE types. The goal is to eliminate, both the conversion function calls and these conversion functions themselves.

If used, the conversion function calls suitable for optimization are replaced by simple C pointer casts, which of course improves performance and reduces the final binary footprint.

Default value: disabled

ARXML input options

The following options configure how ARXML input files are processed, and what kinds of artifacts will be generated from the input files.

--component-filter=[regex]

Configures an ECMA-Script regular expression that the ARXML parser will use to identify Complex Device Driver Software Components (CDD SW-C) relevant for code generation. Please make sure to

Real-Time Innovations International, Inc.
232 E. Java Drive, Sunnyvale, California 94089
Office +1 (408) 990-7400 • Fax +1 (408) 990-7402

include your regular expression within double quotes in order to avoid interference by the operating system's shell interpreter when processing special characters such as parentheses, asterisks, etc.

The complete AUTOSAR model path (including enclosing package names) of each CDD SW-C is used when matching against this regular expression.

For example, suppose your AUTOSAR "model.arxml" ARXML file contains two CDD SW-Cs, "/Foo/Bar/Alpha" and "/Foo/Bar/Beta":

- With *--component-filter=".*Alpha.*"*, only "/Foo/Bar/Alpha" is processed
- With *--component-filter="\/Foo\/Bar\/.+"*, both "/Foo/Bar/Alpha" and "/Foo/Bar/Beta" are processed, but not SW-Cs in other packages
- Without *--component-filter="..."*, all SW-Cs in all packages are processed (see default value below)

Default value: ".*" (all components included)

--idl-types

Configures the Integration Toolkit to produce DDS-IDL data type files (.idl extension).

Default value: disabled

Related: --component-filter, --idl-path

--xml-types

Configures the Integration Toolkit to produce DDS-XML data type files (.xml extension).

Default value: disabled

Related: --component-filter, --xml-path

--conversions

Configures the Integration Toolkit to produce DDS to/from RTE data type conversion files (.c and .h extensions).

Default value: disabled

Related: --component-filter, --c-source-path, --c-header-path, --optimize

--runnables

Configures the Integration Toolkit to produce runnable implementation files for selected CDD SW-C

Real-Time Innovations International, Inc.
232 E. Java Drive, Sunnyvale, California 94089
Office +1 (408) 990-7400 • Fax +1 (408) 990-7402

model elements.

Default value: disabled

Related: --component-filter, --c-source-path, --c-header-path, --optimize

--entities

Configures the Integration Toolkit to produce DDS entity support files (dds_entities.h, dds_entities.c).

Default value: disabled

Related: --component-filter, --c-source-path, --c-header-path

--entities-template

Configures the Integration Toolkit to produce a template configuration file (dds_entities_config.c) for the DDS middleware and the DDS entities it will instantiate. The file is only a template, which the AUTOSAR ECU integrator shall adapt to suit the project needs..

Default value: disabled

Related: --component-filter, --c-source-path

--runnables-suffix=[suffix]

For example, suppose your AUTOSAR "model.arxml" ARXML file contains a CDD SW-Cs "/Foo/Bar/Alpha" with runnables "F1" and "F2":

- With *--component-filter="_dds"*, the Integration Toolkit will generate "Alpha_dds.c", which implements "F1_dds" and "F2_dds"
- Without *--runnables-suffix="..."*, the Integration Toolkit will generate "Alpha.c", which implements "F1" and "F2" (see default value below)

Default value: ""

Related: --c-header-path, --c-source-path

Real-Time Innovations International, Inc.
232 E. Java Drive, Sunnyvale, California 94089
Office +1 (408) 990-7400 • Fax +1 (408) 990-7402

Release notes

Supported platforms

RTI Connext Integration Toolkit for AUTOSAR Classic is supported on the following platforms with x64 CPUs:

- Ubuntu Linux 18.04 LTS (with “g++-9” backport)
- Ubuntu Linux 20.04 LTS
- RedHat Enterprise Linux 8 (with “devtoolset-9” toolset)
- Windows 8.1
- Windows 10
- Windows 11
- macOS version 11.6 (“Big Sur”) and higher

Third-party dependencies

Expat

- Related to: XML parsing
- Software is included in the *rtiarcgen* executable
- Release 2.4.7
- https://github.com/libexpat/libexpat
- License

Real-Time Innovations International, Inc.
232 E. Java Drive, Sunnyvale, California 94089
Office +1 (408) 990-7400 • Fax +1 (408) 990-7402

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Real-Time Innovations International, Inc.
232 E. Java Drive, Sunnyvale, California 94089
Office +1 (408) 990-7400 • Fax +1 (408) 990-7402