



Elektrobit

# EB tresos<sup>®</sup> AutoCore Generic 8 ArtopCmd documentation

Module release 0.0.108





Elektrobit Automotive GmbH  
Am Wolfsmantel 46  
91058 Erlangen, Germany  
Phone: +49 9131 7701 0  
Fax: +49 9131 7701 6333  
Email: [info.automotive@elektrobit.com](mailto:info.automotive@elektrobit.com)

## Technical support

<https://www.elektrobit.com/support>

## Legal disclaimer

Confidential information.

ALL RIGHTS RESERVED. No part of this publication may be copied in any form, by photocopy, microfilm, retrieval system, or by any other means now known or hereafter invented without the prior written permission of Elektrobit Automotive GmbH.

All brand names, trademarks, and registered trademarks are property of their rightful owners and are used only for description.

Copyright 2022, Elektrobit Automotive GmbH.

# Table of Contents

1. Overview .....	4
2. ArtopCmd module release notes .....	5
2.1. Change log .....	5
2.2. New features .....	5
2.3. Elektrobit-specific enhancements .....	6
2.4. Deviations .....	6
2.5. Limitations .....	6
2.6. Open-source software .....	6
3. ArtopCmd user's guide .....	7
3.1. Overview .....	7
3.2. How to use the ArtopCmd .....	7
3.3. ArtopCmd Argument details .....	7
3.3.1. Configuration via xml .....	7
3.3.2. Configuration via commandline argument .....	8
3.3.3. Configuring patches .....	8
3.3.4. Tool Argument Details .....	9
4. ArtopCmd module references .....	11
4.1. Integration notes .....	11
4.1.1. ....	11
4.1.1.1. ArtopCmd dependencies .....	11



# 1. Overview

Welcome to the ArtopCmd product release notes and documentation.

This document provides:

- ▶ [Chapter 2, “ArtopCmd module release notes”](#): details of changes and new features in the current release
- ▶ [Chapter 3, “ArtopCmd user's guide”](#): concept information and configuration instructions
- ▶ [Chapter 4, “ArtopCmd module references”](#): configuration parameters and the application programming interface

## 2. ArtopCmd module release notes

- ▶ Module version: 0.0.108.B567464
- ▶ Supplier: Elektrobit Automotive GmbH

### 2.1. Change log

This chapter lists the changes between different versions.

#### Module version 0.0.20

2020-06-01

#### Module version 0.0.10

2020-04-01

- ▶ Inclusion of documentation.

#### Module version 0.0.4

2020-02-28

#### Module version 0.0.3

2020-02-04

### 2.2. New features

- ▶ No new features have been added since the last release.

## 2.3. Elektrobit-specific enhancements

This module is not part of the AUTOSAR specification.

## 2.4. Deviations

This module is not part of the AUTOSAR specification.

## 2.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- For this module no limitations are known.

## 2.6. Open-source software

Open-source software information is not available for this module.

## 3. ArtopCmd user's guide

### 3.1. Overview

The `ArtopCmd` provides the infrastructure for other tools to do commandline based operations on arxmls.

### 3.2. How to use the `ArtopCmd`

The tool operates in pure commandline mode. It is not called directly, but instead it is called via Tool extensions. For details see the particular Tool extension own documentations.

### 3.3. `ArtopCmd` Argument details

The tool takes complex arguments to control the behavior as well as to define what arxml to process. They are NOT casesensitive.

These arguments can be given as regular commandline arguments, or as xml files or a combination as both. Recommended way is to use xml files, as the configurations can get very complex and are hard to read on commandline as well as there is the risk to exceed the maximum windows command line length.

Functional wise, both options are equal.

Arguments can be primitive(`String,Integer,Long,Float,Boolean,Path,...`), but also structures and lists.

All given paths will be used relative to the current working directory of the tool call.

The available arguments depend on the installed and used extensions. For the arguments provided by the extensions, see the extension own documentations. For general arguments see "Tool Argument Details"

#### 3.3.1. Configuration via xml

The xml file is given to the Tool by the argument `-ArgumentFile`. The xml root is always `Config`, followed by the argument tree as documented in later chapters

If multiple xml files are given, the configuration will be merged. If a list is in more than one file, the result will be sum of both. If a primitive value is given multiple times, only last occurrence will be used. Lists will be appended to the existing one.

Example of how configuring two arxmls would look like:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Config Version="1.0.0">
  <Arxmls>
    <Arxml>
      <Source>in\Arxml1.arxml</Source>
      <Destination>out\Arxml1.arxml</Destination>
    </Arxml>
    <Arxml>
      <Source>in\Arxml2.arxml</Source>
      <Destination>out\Arxml2.arxml</Destination>
    </Arxml>
  </Arxmls>
</Config>
```

To simplify the structure, primitiv struct childs can also be defines as attribues:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Config Version="1.0.0">
  <Arxmls>
    <Arxml Source="in\Arxml1.arxml" Destination="out\Arxml1.arxml" />
    <Arxml Source="in\Arxml2.arxml" Destination="out\Arxml2.arxml" />
  </Arxmls>
</Config>
```

### 3.3.2. Configuration via commandline argument

direct commandline arguments work similar to xml. they always start with - and do not have the "Config" at the beginning.

primitiv values are assigned by "=". e.g. "-LogFile=out\log.txt"

childs of more complex structures are seperates by "/". e.g. "-Arxmls/Arxml"

multiple childs of a list or structure are grouped by "[]" and seperated by ";" e.g. -Arxmls/Arxml[Source=in\Arxml1.arxml;Destination=out\Arxml1.arxml]

### 3.3.3. Configuring patches

Patches to be executed are added as Arguments. For details see the Argument documentation of "Patches"



### 3.3.4. Tool Argument Details

The EcuExtractUpdate takes all arguments from the ArtopCmd plus the specific ones listed here. For the general once, please refer to the ArtopCmd documentation

► **Debug** (Boolean)

Enables debug mode to print additional information to the log/console.

► **LogFile** (Path)

Defines the file to write the log to. If the given file extension is xml, the log will be created in xml format. Otherwise it will be plain text.

► **Tracelevel** (Integer)

If tracelevel is given and larger 0, additional messages will be written to the log and all log messages will be timestamped

► **PrintTime** (Boolean)

Enables the print of timestamps and some durations

Default : false

► **FileNameOnly** (Boolean)

If enabled only the Filenames will be printed to the log. Otherwise the full path will be written to the Log.

Default : false

► **Tool** (String)

Tool to execute. Tools come as separate extensions. For available tools, check the individual extension documentations.

► **ArVersion** (String)

Autosar Version to use for the output arxmls. If not given, highest version of all input arxmls will be used.

► **DisableVersionInfo** (Boolean)

Suppress the printing of the Tool version.

Default : false

► **DisableInstallationInfo** (Boolean)

Suppress the printing of the list of installed extensions.

Default : false

► **SortOutput** (Boolean)

If enabled, all arxml nodes will be sorted by SHORT-NAME.

Default : false

► **AddAsrPath** (Boolean)

If enabled, a xml comment with the autosar short name path will be added at every element with a short name.

Default : false

► **Support64Bit** (Boolean)

If disabled, byte arrays will be used instead of 64 bit types. Enabled by default.

► **ArgumentFile** (Path)

Argument file to read additional arguments from. One argument file can include other argument files via the argument as well.

► **Arxmls** (List)

Arxml files to use as input.

► **Arxml** (Record)

► **Source** (Path)

Path to the arxml to read from.

► **Destination** (Path)

Path to write the modified arxml to. If not specified, the export will be skipped.

► **Variables** (Record)

List of variables that can then be used in other arguments. e.g. for a variable named 'MyVariable', all occurrence in all arguments of \${MyVariable} will be replaced with then value of this variable.

## 4. ArtopCmd module references

ArtopCmd configuration parameter reference is not available.

ArtopCmd API reference is not available.

### 4.1. Integration notes

#### 4.1.1.1. ArtopCmd dependencies

The usage of the module depends on the following other modules

- ▶ ArtopLib