MTS Software Container command-line interface

Contents

Chapter 1

MTS Software Container Tool

1.1 Introduction

This documentation is about using the MTS Software Container Tool (command-line interface) for the purpose of creating Software Containers.

1.2 General information

The Software Container defines a mechanism by which all version and configuration components of a Continental ECU can be combined into a single file-system entity. A Software Container file is created by the MTS Software Container Tool in the data acquisition preparation phase (build process tool chain).

For a quick introduction to the Software Container please refer to the Software Container format specification `swc_format_specification.docx`.

1.3 First steps

1.3.1 Creating your first Software Container

It is recommended to start with a Motorola S-Record file and one or more SDL files. These files can be used to instruct the tool to generate XML template files for all the configuration components that will be embedded in the Software Container.

Run the tool as described in the Command line example: Performing first time preparation of Software Container files section.

Edit the generated XML files to suit your configuration. The generated XML schema files can be used to validate the XML files.

After editing component configurations (XML), run the tool as described in the Command line example: create a Software Container section to create your first Software Container.

1.3.2 Browse a Software Container

Software Containers are ZIP files so by appending .ZIP as the file name extension one can use any of a variety of ZIP tools to look inside them.

For browsing and advanced navigation support a modified version of the Open XML Package Editor for Visual Studio extension is provided in the *SoftwareContainerTool/Visual Studio 2010/OpenXMLEditor/OpenXMLEditor.vsix* location. After installing the extension, one can open Software Container files with ".swc" extension directly in

Visual Studio 2010.

Chapter 2

Command line

The MTS Software Container command line interface supports both short (consisting of a single character and prefixed by a single dash) and long (prefixed by double dashes) commandline options. The long and short options are interchangeable in all contexts.

2.1 Command line options

Option name	Long option	Description
-h	–help	Display help text and exit program
	–help-command	Display help for a single command and exit program
-p	–prepare	Prepares xml file templates for a software container
-b	–build	Creates a software container
-e	–extract	Extracts a software container or CDL file
-s	–sign	Signs the software container
-v	–validate	Validates the software container

2.2 Return code

The application will return a non-zero return code if an error was encountered while executing the requested command. A value of 0 is returned if no errors were encountered.

2.3 Command line example: Performing first time preparation of Software Container files

The *prepare* command is recommended when using the Software Container for the first time. In this phase the tool will create sample XML files and store them with the associated XML schemas.

This command requires a Motorola S-Record file and one or more SDL files.

Argument name	Long option	Required	Description
-m	–mot	Y	Motorola S-Record file
-l	–sdl	Y	SDL file or SDL files separated by spaces
-o	–output	N	Output directory where the files should be generated

The following files will be generated:

Option name	Description
cycle_definition.xml	List of cycle IDs and associated names. This file is pre-populated based on the input SDL file(s) content
cycle_definition.xsd	XML Schema for the cycle_definition.xml file
channel_definition.xml	List of data channels to be used
channel_definition.xsd	XML Schema for the channel_definition.xml file
hw_data_cycle.xml	Mapping of MTA interface IDs to cycle IDs. This information is optional
hw-data-cycle.xsd	XML Schema for the hw_data_cycle.xml file
channel_mapping.xml	Mapping of virtual address (ranges) and cycle IDs to data channel index. This information is optional
channel_mapping.xsd	XML Schema for the channel_mapping.xml file
manifest.xml	Software Container manifest based on the Motorola S-Record file content. This file is provided for debugging purposes only
manifest.xsd	XML Schema for the manifest.xml file

The generated files should be edited to suit the correct configuration for a specific project.

Important: after editing the content of the XML files do not run the prepare command on the same directory as it will overwrite the edited files.

2.4 Command line example: create a Software Container

The following option arguments are available for the *build* command:

Argument name	Long option	Required	Description
-m	–mot	Y	Motorola S-Record file
-l	–sdl	Y	SDL file or SDL files separated by spaces
-c	–cycles	Y	Cycle defintion XML file
-d	–channels	N	Data channel defintion XML file
-w	–hw-data	N	Hardware data defintion XML file
-e	–mapping	N	Channel mapping defintion XML file
-a	–a2l	N	A2L file or A2L files separated by spaces
-o	–output	Y	Output directory and filename of the generated Software Container
-k	–keep-sizes	N	If specified will instruct the tool to use type definition sizes specified
			in the SDL files instead of recalculating them

The following command creates a new unsigned Software Container

```
mtsswccli --build -o .\srr520\srr520.swc --mot .\srr520\srr520.mot --sdl .\srr520\srr520.sdl --a2l .
      \srr520\srr520.a2l --cycles .\srr520\cycle_definition.xml --channels .\srr520\channel_definition.xml --hw-data .
      \srr520\hw_data_cycle.xml --mapping .\srr520\channel_mapping.xml
```

2.5 Command line example: sign a Software Container

The following command digitally signs a Software Container

```
mtsswccli.exe --sign "sample.swc"
```

Note: This command should not be used until a public key infrastructure (PKI) will be rolled out.

2.6 Command line example: validate a Software Container

The following command validates a Software Container

```
mtsswccli.exe --validate "sample.swc"
```

This command checks that XML files part of the Software Container are properly validated against their XSD schema and that a digital signature is present.