AMiT, spol. s r.o. does not provide any warranty concerning the contents of this publication and reserves the right to change the documentation without any obligation to inform about it.

This document can be copied and redistributed under following conditions:

1. The whole text (all pages) must be copied without any changes.
2. All redistributed copies must retain the AMiT, spol. s r.o. copyright notice and any other notices contained in the documentation.
3. This document must not be distributed for purpose making of profit.

The names of products and companies used herein can be trademarks or registered trademarks of their respective owners.

AMiT is a registered trademark.
Contents

List of figures ................................................................. 5
History of revisions .......................................................... 6
Related documentation ...................................................... 6

Definition of abbreviations .................................................. 7

1. Introduction ........................................................................ 8
2. Technical parameters ....................................................... 9
3. Principles of WTB networks behaviour ................................. 10
4. Mapping PD between WTB and MVB busses .......................... 12
   4.1. Export marshalling .................................................... 12
   4.2. Import marshalling .................................................... 13
   4.3. Mapping example ..................................................... 14
       4.3.1 Rear vehicle closed doors check at two–vehicle rolling stock ........ 14
           Export marshalling: ................................................ 14
           Import marshalling: .............................................. 15
       4.3.2 All doors closed check at two–vehicle rolling stock ..................... 15
           Import marshalling: .............................................. 15
       4.3.3 All doors closed check on three–vehicle rolling stock .................. 15
       4.3.4 Sending order to close the doors of the rolling stock .................... 15
           Export marshalling: ................................................ 15
           Import marshalling: .............................................. 16
5. Mapping PD between WTB and Ethernet busses .................... 17
6. Transmission of E-Telegrams over the Ethernet ........................ 18
7. Configuration application control description ........................ 19
   7.1. Elementary structuring ............................................... 19
   7.2. Menu ........................................................................ 19
   7.3. Tabs ........................................................................ 20
   7.4. Database definition Tab .............................................. 20
   7.5. MVB Definition Tab .................................................. 21
   7.6. WTB configuration ..................................................... 21
   7.7. WTB static properties definition .................................... 21
   7.8. Messenger Editor ...................................................... 22
   7.8.1 Routing directories specification .................................... 23
   7.9. MVB -> WTB mapping definition ................................... 23
       7.9.1 MVB sink port definition .................................... 23
       7.9.2 MVB sink port on WTB source port mapping .................... 23
   7.10. WTB -> MVB mapping definition ................................... 24
       7.10.1 MVB source port definition ................................... 25
       7.10.2 WTB sink port to MVB source port mapping .................... 25
   7.11. ETH -> WTB mapping definition ................................... 26
       7.11.1 Ethernet sink port definition .................................. 26
       7.11.2 Ethernet sink port to WTB source port mapping ................ 26
   7.12. WTB -> ETH mapping definition ................................... 27
       7.12.1 Ethernet sink port definition .................................. 27
       7.12.2 WTB sink port to Ethernet source port mapping ................ 27
7.13. Error List .......................................................................................................................... 28

8. Internal structures and internal marshalling ................................................................. 29

8.1. Internal import marshalling .......................................................................................... 29
NADI exported structure: ................................................................................................. 29
Exported structure Gateway status.................................................................................. 30
Exported structure Domain descriptors: .......................................................................... 31
Exported structure Marshalling status: ............................................................................ 32

8.2. Internal export marshalling .......................................................................................... 32
Imported structure Marshalling control: .......................................................................... 33

9. ADSManager ..................................................................................................................... 35

9.1. Program description ....................................................................................................... 35
9.2. Menu ............................................................................................................................... 36
9.2.1 File ............................................................................................................................... 36
9.2.2 Option ........................................................................................................................ 36
9.3. List of devices ................................................................................................................. 36
9.4. List of domains ............................................................................................................... 38
9.4.1 Operations over domains .......................................................................................... 39
9.5. Status bar ....................................................................................................................... 39
9.6. XML script ..................................................................................................................... 39
9.6.1 GUI mode .................................................................................................................. 40
9.6.2 Command line mode ................................................................................................. 40

10. Loading domains to device ............................................................................................ 42

10.1. TCN-GW-WEM_REF domains .................................................................................... 42
10.2. Loading of new device ............................................................................................... 43
10.3. Loading of configuration domain ............................................................................... 43
10.4. Domain upgrade .......................................................................................................... 44
List of figures

| Fig. 1 | Example of four vehicle train composition ........................................10 |
| Fig. 2 | Example of orientation pair ..........................................................12 |
| Fig. 3 | Two tram vehicle composition ............................................................14 |
| Fig. 4 | Static variables definition ...............................................................22 |
| Fig. 5 | Messenger Editor ...............................................................................22 |
| Fig. 6 | Ethernet sink port to WTB source port mapping ....................................26 |
| Fig. 7 | WTB sink port to Ethernet source port mapping ......................................27 |
| Fig. 8 | Error List ............................................................................................28 |
| Fig. 9 | Description of the screen with target device detected on Ethernet ..........35 |
| Fig. 10 | Preferences ......................................................................................36 |
| Fig. 11 | Main window with found devices ..........................................................37 |
| Fig. 12 | List of domains on target device ........................................................38 |
| Fig. 13 | GUI XML script – domain update .......................................................40 |
| Fig. 14 | Command line XML script – domain update ...........................................41 |
History of revisions

Author: Adam Dočekal, Ondřej Maštálka, Jiří Skála

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>31. 7. 2014</td>
<td>New document</td>
</tr>
<tr>
<td>101</td>
<td>10. 12. 2014</td>
<td>Add Definition of abbreviations, correct name of Related documentation, correct English terms Add chapter Technical parameters</td>
</tr>
<tr>
<td>102</td>
<td>27.1.2015</td>
<td>Target device added</td>
</tr>
<tr>
<td>103</td>
<td>3.2.2015</td>
<td>Reference application added</td>
</tr>
</tbody>
</table>

Related documentation

1. RRU-WEM/1101 – Operation manual
   file: rv-w_g_en_xxx.pdf
2. TROL – Service libraries for procedural processor – Programmers manual
   file: trol_ms_cz_xxx.pdf
3. TrolDatGen – configuration utility – User’s manual
   file: troldatgen_ms_cz_xxx.pdf
4. IEC 61375-2-1 ed1.0:2012 – Wire Train Bus (WTB)
5. UIC 556 ed5:2009 – Information transmission in the train (train bus)
7. IEEE 802.3u – IEEE Standards for Local and Metropolitan Area Networks
9. TCN-ETH-MSG protocol description, tcn-eth-msg_ms_en_xxx.pdf
Definition of abbreviations

**ADD**  AMiT device detection

**ADS**  AMiT domain services

**CAN**  Controller area network (ISO 11898-1)

**CiA**  CAN in Automation organization (http://www.can-cia.org)

**COB-ID**  CAN object identifier (11-bit value, CiA 301 chapter 3)

**E-Telegram**  UIC event telegram which carries message data

**FTF**  Frame type format; two first bytes of R-Telegram

**MD**  Message data

**MVB**  Multifunction vehicle bus

**NSDB**  Node supervisor database

**PD**  Process data

**PDO**  CAN process data object (CiA 301, chapter 9.2.1)

**R-Telegram**  UIC regular telegram which carries process data

**SDF**  Source directory format

**SDO**  CAN service data object (CiA 301, chapter 9.2.2)

**TCN**  Train communication network

**UDP**  User datagram protocol

**WTB**  Wire train bus
1. Introduction

This manual describes the features of TCN-GW-WEM_REF software. The software is running on hardware from AMiT production which is referred in the following text as target device, eventually as device. List of compatible hardware can be found in chapter Technical parameters.

TCN-GW-WEM_REF allows transmission of process data between WTB, Ethernet, and MVB buses. In addition to process data transmission between consist buses and WTB (import and export marshalling), TCN-GW-WEM_REF is capable of internal transmission of PD between single consist busses (Ethernet -> MVB, MVB -> Ethernet). Internal buffer with 1024 bytes size is defined for this purpose (see chapter 8).

TCN-GW-WEM_REF also allows transmission of E-Telegrams between WTB, Ethernet and MVB busses.

TCN-GW-WEM_REF provides information about its status and information about current train composition. Software application can control TCN-GW-WEM_REF behaviour on WTB bus (setting / cancelling leading request, inhibiting / allowing inauguration, changing TCN operation mode).

TCN-GW-WEM_REF gateway behaves as an MVB slave on MVB network.

TCN-GW-WEM_REF software serves as reference application. It is fully functional but still there is great count of possibilities of project dependent changes. In such a case the project dependent implementation will be offered based on technical specification.
## 2. Technical parameters

### WTB

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Compatibility</strong></td>
<td>IEC 61375-2-1 ed1.0:2012, UIC 556 ed5:2009</td>
</tr>
<tr>
<td>Process data</td>
<td>Yes</td>
</tr>
<tr>
<td>Message data</td>
<td>Yes</td>
</tr>
<tr>
<td>Quick reinsertion</td>
<td>Yes</td>
</tr>
<tr>
<td>Background MD scanning</td>
<td>Configurable in NSDB</td>
</tr>
<tr>
<td>MD packets routing</td>
<td>Yes</td>
</tr>
<tr>
<td>MD routing directories</td>
<td>Configurable in NSDB</td>
</tr>
<tr>
<td>R-Telegrams</td>
<td>Yes, PD mapping configurable in NSDB</td>
</tr>
<tr>
<td>R-Telegrams individual period</td>
<td>Configurable in NSDB</td>
</tr>
<tr>
<td>Static properties</td>
<td>Configurable in NSDB</td>
</tr>
<tr>
<td>UIC vehicle ID</td>
<td>Optionally configurable in NSDB, set by PD mapping or by TCN-ETH-MSG protocol</td>
</tr>
<tr>
<td>User E-Telegrams</td>
<td>Yes, TCN-ETH-MSG protocol</td>
</tr>
</tbody>
</table>

### Ethernet

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Compatibility</strong></td>
<td>IEEE 802.3u</td>
</tr>
<tr>
<td>Process data</td>
<td>Yes, TCN-ETH-PDB protocol</td>
</tr>
<tr>
<td>Message data</td>
<td>Only E-Telegrams, TCN-ETH-MSG protocol</td>
</tr>
</tbody>
</table>

### MVB

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Compatibility</strong></td>
<td>IEC 61375-3-1 ed1.0:2012</td>
</tr>
<tr>
<td>Bus administrator</td>
<td>No</td>
</tr>
<tr>
<td>Process data</td>
<td>Yes</td>
</tr>
<tr>
<td>Message data</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Configuration

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSDB creation/editing</td>
<td>GTWconfigurator</td>
</tr>
<tr>
<td>Configuration (NSDB) download</td>
<td>ADSManager connected on Ethernet</td>
</tr>
</tbody>
</table>

### Target device

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>WTB/Ethernet/MVB gateway</td>
<td>RRU-WEM/1151</td>
</tr>
<tr>
<td></td>
<td>RV-WEM/1151</td>
</tr>
</tbody>
</table>

### Reference application

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RRU-WEM/1151</td>
<td>TCN-GW-WEM_REF</td>
</tr>
<tr>
<td>RV-WEM/1151</td>
<td>TCN-GW-RWEM_REF</td>
</tr>
</tbody>
</table>
3. **Principles of WTB networks behaviour**

WTB is a train network with dynamic topography connecting single vehicles or groups of vehicles (Consist\(^1\)) of train set compositions. Networking features are specified in IEC 61375-2-1 and UIC 556.

Inauguration on WTB bus takes place after train composition initialization, determining leading vehicle\(^2\), lengthening or shortening train composition\(^3\). NADI mapping table is created during inauguration. From the point of view of application these UIC properties are defined:

- Current number of vehicles in train
- Train master vehicle (Leading)
- UIC addresses of single vehicles
- UIC referent direction defines orientation of train (direction definition: left, right, front, reverse)
- Definition of orientation of single vehicles according to UIC reference direction
- Types of R-telegrams with process data, transmitted by single WTB gateways

Other parameters referring to TCN are:

- Current number of WTB gateways
- Current network master WTB (defined dynamically or by STRONG master request on particular WTB gateway)
- WTB gateway TCN addresses
- WTB gateway bottom and top TCN address
- Single TCN gateway orientation according to TCN Master gateway

![Diagram of WTB network](image)

**Fig. 1** - Example of four vehicle train composition

---

1) Consist is a group of vehicles with one WTB gateway. Vehicles are interconnected via TCN Consist network (e.g. CAN, MWB, etc.)

2) Leading vehicle is a leading vehicle of composition. It is defined by the presence of driver (Key inserted, etc.)

3) For example, train lengthening by coupling additional vehicles, in case that no inhibit request is set.
After inauguration WTB gateway starts to transmit source port of process data (R-telegram) over the WTB bus and in the same time it receives ports from other WTB gateways (sink ports). Type of transmitted R-telegram is a result of inauguration, and it is given by dynamical an statistical properties of vehicle.

Following types of telegrams are transmitted:

<table>
<thead>
<tr>
<th>Type of R-telegram</th>
<th>Data part</th>
<th>Telegram source</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>2 octets header FTF, 38 octets information for vehicles (the same as R3), 88 octets commands from the master vehicle</td>
<td>Master vehicle (Leading)</td>
</tr>
<tr>
<td>R2</td>
<td>2 octets header FTF, 38 octets information for vehicles (the same as R3), 88 octets messages from the tractive vehicle (vehicles of the consist)</td>
<td>Traction Vehicle (Traction)</td>
</tr>
<tr>
<td>R3</td>
<td>2 octets header FTF, 38 octets information for vehicles</td>
<td>Trailed vehicle (Follower)</td>
</tr>
</tbody>
</table>

Only one vehicle in train composition can send R1 telegram. Due this fact each vehicle can be accessed directly with filtering the certain type of R-telegram in some train compositions. There is no need to define data source using UIC or TCN address in case of two vehicles composition (see Chapter 4).
4. Mapping PD between WTB and MVB busses

Process data transmission is logically separated on data transmission from MVB bus to WTB bus (so-called export marshalling) and data transmission from WTB bus to MVB bus (so-called import marshalling).

4.1. Export marshalling

Export of process data is defined by mapping one or more MVB sink ports or its parts to WTB source port. User defines following parameters:

- **MVB port address**
- **Source offset of data in MVB sink port**
- **Target offset of data in WTB source port**
- **Data length**
- **Mapping type**: Byte, inverse byte, bit, inverse bit
  - **Byte mapping** is used for transmission of PD formed by process variables with length in multiples of bytes (for example variables byte, word, dword aligned by byte in MVB port).
  - **Inverse byte mapping** is used mostly for big/little endian conversion in process variables with length in multiples of bytes.
  - **Bit mapping** is used for transmission of process variables, which do not have length in multiples of bytes or their lengths are not aligned on byte (for example 4 bits enumeration starting on the 2nd bit in MVB port).
  - **Inverse bit mapping** is used mostly for process variables bit rotation in case of their different interpretation (for example, 0-th bit of TCN process variable has value 0x80, CAN has value 0x1).

- **Orientation pair**
Application cannot be synchronised with change of vehicle orientation because the change of orientation is the result of an inauguration on WTB. According to this fact user has an option to mark same length mapping pairs on WTB as orientation pairs. In case the vehicle has different orientation than UIC reference direction, TCN-GW-WEM_REF automatically switches values in particular orientation pair. It is ensured that values referring to orientation would be exported correctly, if the application uses orientation pairs from process variables related to orientation (for example: left and right doors), in case of vehicle orientation change.

![Fig. 2 - Example of orientation pair](image)
- **Mapping of process data or information about train composition**
  Instead of mapping PD, it is possible to map from MVB port command for particular TCN-GW-WEM_REF, for example request for Leading. It is specified in chapter 8.2.

  *Note:* Data transfer is not synchronised with reception of ports from MVB bus. For this reason, consistency of data received from multiple MVB ports is not guaranteed.

### 4.2. Import marshalling

Import of PD is defined by mapping from one or more WTB sink ports to one or more MVB source ports. User defines following parameters:

- **Source offset of data in WTB sink ports**
- **Target MVB source port address**
- **Target offset of data in MVB source port**
- **Data length**
- **Mapping type:** Byte, inverse byte, bit, inverse bit
- **Orientation pairs** – same meaning as for export marshalling
- **Filtering mapping according to type of R-telegram**
  User selects from what type of R-telegram mapping will be defined. User can select any combination of local R1, R2 or R3 telegrams.

- **Mapping according to TCN and UIC addresses (SDF)**
  - Mapping according to TCN addresses (TCN SDF) is controlled by current train composition according to TCN addresses. Single WTB gateways are mapped. First is the bottom gateway.
  - Mapping according to UIC addresses (UIC SDF) is controlled by current train composition according to UIC addresses. Single vehicles are mapped. First is the vehicle with UIC address 1.

  In both cases user defines position in MVB source port, where values from different vehicles or WTB gateways are stored. User can define more MVB source ports. User defines maximum number of vehicles and gateways in train composition, for the reason of saving space.

- **Process data or information about the train composition**
  Instead of mapping process data, it is possible to map TCN-GW-WEM_REF gateway status information or information about the current train composition into MVB source ports (specified in chapter 8.1).

  **SDF** (Source Directory Format) is the array of data section of particular mapping for single vehicles / gateways.

  In case of **UIC SDF**, the array is arranged by the vehicle UIC addresses. It means that vehicle with UIC address 1 occupies position with index 1. Vehicle with UIC address 2 occupies the position with index 2.

  In case of **TCN SDF** the array is arranged by the WTB gateway in order of TCN, direction from the Bottom (index 1) to the Top. TCN SDF is used in the case when multiple vehicles consist occurs in the composition. For example coupling
of two 6-vehicle consists creates 12-vehicle train composition with 2 WTB gateways. In this case UIC mapping contains 12 items. The first 6 items contain the same data as the second 6 items. It is more preferable to use TCN SDF, which contains only two items for both gateways.

Note: Data import takes place only if the data source (particular WTB sink port) is uniquely defined. Import of process data will not take place in case of situation when there are more or none data source after applying filter. Values in target data are always zeroed before import execution. This behaviour protects application from handling obsolete or random values, in cases, when the import is not successful. Process data import from WTB sink ports and handling R-Telegrams types doesn’t take place during inauguration. Import from local structures using “local gateway” filtration takes place even during inauguration.

Example: Let’s consider two-vehicle composition with a gateway in each vehicle. Both gateways send R2 telegram. If user sets filtering by R2 telegram in import of process data will not take place.

4.3. Mapping example

This chapter provides examples of PD mapping between WTB and MVB busses for two or more vehicle sets.

4.3.1 Rear vehicle closed doors check at two–vehicle rolling stock

For example, the rear vehicle door status will be sent by control unit in rear vehicle in MVB port with address 201h. Left door status will be mapped as 1 bit unit saved in 0.3 bit of MVB dataset. Right door status will be mapped as 1 bit unit saved in 0.4 bit of MVB dataset.

![Diagram](image.png)

Fig. 3 - Two tram vehicle composition

Export marshalling:

In TCN-GW-WEM_REF configuration, the user sets orientation pair export marshalling to pair of 1 bit units saved in PDO 0.3 and 0.4 bits of MVB port with address 201h into WTB source port, bit mapping. According to UIC 556
appendix A, the unit „All left doors locked“ must be mapped to 20.4 bit of WTB source port, and unit „All right doors locked“ to 20.5 bit of WTB source port.

Import marshalling:

In TCN-GW-WEM_REF configuration, the user sets orientation pair import marshalling to pair of 1 bit variables of R2-telegram (assuming the rear vehicle is tracking vehicle, else R3-telegram is used) on 20.4 offset (left doors) and 20.5 offset (right doors) Orientation pair will be mapped to source MVB port with address 181h (transmitted by TCN-GW-WEM_REF) with 0.3 bit and 0.4 bit offsets of data.

TCN-GW-WEM_REF configuration of both vehicles can be identical.

Left and right door offset in R3-telegram (R2 also contain R3 data) is specified in UIC 556. In case there is no need to keep conformal UIC structure of R-telegram, it is possible to map data anywhere to WTB port.

4.3.2 All doors closed check at two–vehicle rolling stock

Procedure is the same as in previous example; the only difference is in the import marshalling.

Import marshalling:

In TCN-GW-WEM_REF configuration, the import marshalling of two orientation pairs (2x2 bit variables), saved in array UIC SDF with index 1 and 2, on 20.4 (left doors) and 20.5 (right doors) offsets, is set. Orientation pairs will be mapped to MVB source port with address 181h (sent by TCN-GW-WEM_REF gateway) with 0.3 and 0.4 bit offsets for first orientation pair (UIC address 1, leading vehicle), and 0.5 and 0.6 bits offsets for second (UIC address 2, second vehicle).

4.3.3 All doors closed check on three–vehicle rolling stock

Procedure is similar to chapter 4.3.2. The difference is mapping 3 orientation pairs via UIC SDF.

4.3.4 Sending order to close the doors of the rolling stock

Control unit located in leading vehicle sends order to close all doors. Unit sends command as a bit variable over MVB in port with address 201h on 4.1 offset.

Export marshalling:

In TCN-GW-WEM_REF configuration, the export marshalling of bit variable, saved in 4.1 bit of MVB port with address 201h, to WTB source port of the unit that is transmitting R1 telegram, is set (filtering according to R-telegram = R1,

4) Except part of header of FTF telegram.
corresponds to unit in Leading vehicle). Bit mapping. According to UIC 556 appendix A, the unit „Close all entry doors“ must be mapped to 20.0 bit of WTB source port.

**Import marshalling:**

In TCN-GW-WEM_REF configuration, user sets the import marshalling of R1-telegram on 20.0 offset. Variable will be mapped to MVB source port, transmitted by TCN-GW-WEM_REF gateway, with address 181h on offset, for example, 4.1.

TCN-GW-WEM_REF gateway configuration of both vehicles can be identical.

R-telegram commands offset are specified in UIC 556. In case there is no need to keep conformal UIC structure of R-telegram, it is possible to map data anywhere to WTB port.
5. Mapping PD between WTB and Ethernet busses

PD transmission between WTB and Ethernet busses follows similar rules as a PD transmission between WTB and MVB networks. Mapping is logically divided on export, import and internal marshalling. User defines parameters according to similar rules as for PD mapping between WTB and MVB busses.

PD transmission over the Ethernet bus is realized via proprietary protocol Ethernet process data broadcasting (TCN-ETH-PDB). Detailed description of this protocol is provided in programmer’s manual for this protocol [8].

Detailed description of definition of PD mapping parameters between WTB/Ethernet is provided in chapter 7.
6. Transmission of E-Telegrams over the Ethernet

E-Telegram transmission over the Ethernet bus is realized via TCN-ETH-MSG proprietary protocol. Detailed description is provided in programmer's manual for TCN-ETH-MSG protocol [9].
7. Configuration application control description

This chapter describes the GTWConfigurator program user interface. Program is used for WTB, MVB parameters configuration, and for mutual mapping between WTB and MVB ports, or between WTB and Ethernet ports. Precondition for using this program is a basic knowledge of TCN WTB network, TCN MVB network, Ethernet network as well as knowledge of using TCN process data, messages and UIC E-Telegrams.

7.1. Elementary structuring

Configuration is divided on following parts:
- Gateway parameters on WTB bus
- Gateway parameters on MVB bus
- Reciprocal mapping of WTB ports on MVB ports
- Reciprocal mapping of WTB ports on Ethernet ports

Gateway configuration procedure is as follows:
- Database establishing
- Setting of WTB and MVB bus parameters
- Setting parameters of mapping data objects between buses
- Loading the database file into gateway

7.2. Menu

File:
- **New** – establishing of new configuration. This function performs deletion of actual configuration and allows user to create the new one.
- **Load** – loading configuration from a file
- **Save** – saving configuration with actual name. Unless the configuration has been saved, the user is prompted to name the file and select location for saving.
- **Save As** – saving configuration with new name. User can save configuration as .DOM or .XML type file
- **Exit** – program termination

Configuration:
- **Validate** – database validation

Options:
- **Preferences** – selection of default folder for saving / loading configuration files, option to create .XML equivalent to each .DOM file
Help:
- About – displays information about program

Device – gateway type selection:
- WTB – MVB

Warning:
During the change of Device item, depending on selected device type, tabs relevant for selected configuration type are shown or hidden. All previously completed mappings and definitions are kept in memory. This only applies until the user saves the database. During the saving, only the data valid for the selected device type are saved (data contained on displayed tabs). Other data are erased.

7.3. Tabs

User interface is divided into ten tabs, where parameters of individual database sections can be defined. These are the following tabs, described in following chapters:

- Database definition Tab
- MVB Definition Tab
- WTB configuration definition
- WTB static properties definition
- Messenger editor
- WTB -> MVB mapping
- MVB -> WTB mapping
- WTB -> ETH mapping
- ETH -> WTB mapping
- Error list

7.4. Database definition Tab

User defines identifying characters of database on this tab.

These are the following items:
- **Database Name** – database user name. Maximum 31 characters is allowed (independent on filename)
- **User Type** – database user type. Maximum 31 characters is allowed (does not affect functionality)
- **Date** – database creation date. The current date is used upon database establishing. User can change the date arbitrarily. Subsequently, the entered date appears in domain parameters.

Following three items are read only and filled automatically by program:
- **Database Identifier** – database numerical identifier
- **32-bit CRC** – CRC code calculated over the whole configuration

### 7.5. MVB Definition Tab

User defines MVB network parameters on this tab. These are the following items:

- **Device address** – device address on MVB bus
- **Reply timeout** – timeout for waiting for slave frame
- **MD priority** – message priority, for which the device will be called by bus administrator (low, high)
- **Lifetime start** – lifetime signal timeout after start of the device in ms
- **Lifetime operational** – lifetime signal timeout in regular operation of device in ms

### 7.6. WTB configuration

User defines WTB parameters on this tab. These are the following items:

- **WTB Compatibility**
  - *TCN release standard* – compatibility with IEC 61375-2-1
  - *Homologation tests* – compatibility with UIC 556 testbed
- **Fritting configuration** – enable / disable fritting
- **MD Idle polling configuration** – enable / disable background scanning for MD transmission

In addition, user specifies lengths and period of R-Telegram for single UIC operation modes.

### 7.7. WTB static properties definition

User defines static properties of consist (gateway) and static properties of single vehicles within consist on this tab. Static properties define, what is connected to rolling stock, what services consist provides, and what type of R-telegram is sent by TCN-GW-WEM_REF. User defines static properties via serial number.

In addition to other, static properties include international vehicle identification number – UIC vehicle ID (static property #140), number of vehicles in consist (static property #139), version of R-telegram (static property #134) or information, whether vehicle contain gateway (static property #141).

More information about static properties could be found in UIC 556 Appendix E.
If UIC vehicle ID is not set in static properties; TCN-GW-WEM_REF does not inaugurate on WTB. After start, it waits for setting of UIC vehicle ID via imported Control structure.

7.8. Messenger Editor

- **TCN Compatibility** – specification of compatibility according to particular group of manufacturers. There are three groups to select:
  - Interoperable version (FAR System, Duagon)
  - Homologated version (testbad Siemens, Bombardier)
- Standard normalized version (UniControls, EKE, Škoda)
- **Station ID** – 8 bit station identification. Takes values (1-254) Values 0 and 255 are reserved.
- **Maximum packet size** – maximum packet size specification:
  - 0 – MVB
  - 1 – WTB
  - 2 – Serial
  - 3 – Ethernet
- **Credit** – slide window size – values (1 – 7)
- **Multicast credit** – slide window size for multicast. Recommended value is 10.
- **Session timeout** – session timeout in ms

**Specification of transport level timeout [ms]:**
- Send timeout
- Acknowledge timeout
- Alive timeout
- Multicast send timeout
- Multicast pause timeout

### 7.8.1 Routing directories specification
Routing directories can be specified in tables Station directory, Function directory, Group directory.

### 7.9. MVB -> WTB mapping definition
User defines export marshalling on this tab. It means PD transmission from MVB bus to WTB bus. Export is defined by mapping one or more MVB sink ports to WTB source port.

#### 7.9.1 MVB sink port definition
- **Address** – port address
- **FCode** – port bit length 16 – 1024 bits
- **Period** – period of transmitting ports over the network

#### 7.9.2 MVB sink port on WTB source port mapping
- **MVB Byte Offset** – source byte offset in MVB sink port
- **MVB Bit Offset** – bit offset in source byte
- **Import GW Controls** – unit ID from imported Controls structure – see chapter 8.2
- **Import Marsh. Controls** – unit ID from imported Marshalling Controls structure – see chapter 8.2
• **DST Byte Offset** – destination byte offset in source WTB port
• **DST Bit Offset** – bit offset in destination byte
• **Bit Len** – mapped data length
• **Copy Type** – type of copying, see below
• **Data / Internal** – mapping data or internal structures
• **Paired** – flag defining whether data parts of orientation pairs are transmitted

**Copy Type:**
- **Byte copying** is used for transmission of PD consisting from process variables with length in multiples of bytes.
- **Inverse byte copying** is used mostly for big/little endian conversion in process variables with length in multiples of bytes.
- **Bit copying** is used for transmission of process data consisting from process variables, which do not have length in multiples of bytes or their lengths are not aligned on byte.
- **Inverse bit copying** is used mostly for process variables bit rotation in case of their different interpretation (for example, 0-th bit of TCN process variable has value 0x80, CAN has value 0x1).

**Orientation pair**
The orientation change is the result of inauguration on WTB. Due this fact application with orientation change cannot be synchronised. User has an option to mark same length mapping pairs on WTB as orientation pairs. User can do so by choosing two items with same bit length in MVB > WTB Map Items table by calling menu **Pair Items** from the context menu. Successful pairing of two mappings is indicated by colouring of two cells in Paired column in to same colour.

In case the vehicle has different orientation then UIC reference direction, TCN-GW-WEM_REF automatically switches values in particular orientation pair. It is ensured that values referring to orientation would be exported correctly if the application uses orientation pairs for process variables related to orientation.

### 7.10. WTB -> MVB mapping definition

User defines import marshalling on this tab. It means PD transmission from WTB bus to MVB bus. Import marshalling is defined by mapping one or more WTB sink ports to one or more MVB source ports.

- **Max number of gateways** – maximum number of gateways
- **Max number of vehicles** – maximum number of vehicles
- **Freshness Limit** – freshness limit value. Interval can take values from 256 to 8192 ms.
Note:

*Items Max number of gateways and Max number of vehicles are common also for WTB <-> Ethernet mapping.*

### 7.10.1 MVB source port definition

- **Address** – port address
- **FCode** – port bit length 16 – 1024 bits
- **Period** – period of transmitting ports over the network

### 7.10.2 WTB sink port to MVB source port mapping

- **Local** – mapping from own gateway. Selection option Local causes cancellation of all filtrations R1, R2, R3.
- **R1, R2, R3** – user selects, from what type of R-telegram mapping will be defined. User can select any combination of R1, R2 or R3 telegram. At least one type must be selected. Selection R1, R2 or R3-Telegram type causes cancellation of filtration Local.
- **SRC Byte Offset** – source byte offset
- **SRC Bit Offset** – source bit offset in source byte
- **Export Nadi ID** – unit ID from exported NADI structure, see chapter 8.1
- **Export GW Status ID** – unit ID from exported unit GW Status (see chapter 8.1)
- **Export Domain ID** – domain ID in item Domain_Id of Domain descriptor structure (see chapter 8.1)
- **Export Domain Descriptor** – ID of item from exported structure Domain Descriptor (see chapter 8.1)
- **SDF Type** – mapping according to TCN address or UIC address, or mapping is not SDF
- **SDF Index** – index in TCN composition or UIC address
- **WTB Byte Offset** – destination byte offset in mapped MVB source port
- **MVB Bit Offset** – bit offset in destination byte
- **Bit Len** – mapped data length
- **Copy Type** – same as for export marshalling
- **Data/Freshness/Internal** – mapping data, freshness or internal structures
  - **Paired** – meaning is similar to export marshalling

#### SDF Type

- **No** – there is no SDF mapping
- **TCN** – SDF is controlled by actual train composition according to TCN addresses. Single WTB gateways are mapped. First is bottom gateway.
- **UIC** – SDF is controlled by actual train composition according to UIC addresses. Single vehicles are mapped. First is vehicle with UIC address 1.

SDF Index Index follows the SDF Type. For SDF type:

- **No** – not defined
7.11. ETH -> WTB mapping definition

User defines export marshalling on this tab. It means PD transmission from Ethernet bus to WTB bus. Export marshalling is defined by mapping one or more Ethernet sink ports to WTB source port.

![Ethernet sink port to WTB source port mapping](image)

Fig. 6 - Ethernet sink port to WTB source port mapping

7.11.1 Ethernet sink port definition

- **Address** – port address
- **FCode** – port bit length 16 – 1024 bits
- **Period** – period of transmitting frames over the network

7.11.2 Ethernet sink port to WTB source port mapping

- **ETH Byte Offset** – byte offset in Ethernet sink port
- **ETH Bit Offset** – bit offset in source byte
- **Import GW Controls** – unit ID from imported Controls structure (see chapter 8.2)
- **Import Marsh. Controls** – unit ID from imported Marshalling Controls structure (see chapter 8.2)
- **DST Byte Offset** – destination byte offset in WTB source port
- **DST Bit Offset** – bit offset in destination byte
- **Bit Len** – mapped data length
- **Copy Type** – type of copying (see chapter 7.9.2)
- **Data / Internal** – mapping data or internal structures
- **Paired** – flag defining whether data parts of orientation pairs are transmitted.
7.12. WTB -> ETH mapping definition

User defines import marshalling on this tab. It means PD transmission from WTB bus to Ethernet bus. Import marshalling is defined by mapping one or more WTB sink ports to one or more Ethernet source ports.

![Figure 7 - WTB sink port to Ethernet source port mapping](image)

7.12.1 Ethernet sink port definition

- **Address** – port address
- **FCode** – port bit length (16 – 1024 bits)
- **Period** – period of transmitting ports over the network

7.12.2 WTB sink port to Ethernet source port mapping

- **Local** – mapping from own gateway. Selecting option Local causes cancellation of all filtrations R1, R2, R3.
- **R1, R2, R3** – user selects, from what type of R-telegram mapping will be defined. User can select any combination of R1, R2 or R3 telegram. At least one type must be selected. Selection R1, R2 or R3-Telegram type causes cancellation of filtration Local.
- **SRC Byte Offset** – source byte offset in WTB sink port
- **SRC Bit Offset** – source bit offset in source byte
- **Export Nadi ID** – unit ID from exported NADI structure (see chapter 8.1)
- **Import GW Status ID** – unit ID from exported unit GW Status (see chapter 8.1)
- **Export Domain ID** – domain ID in item Domain_Id of Domain descriptor structure (see chapter 8.1)
- **Export Domain Descriptor** – ID of item from exported structure Domain Descriptor (see chapter 8.1)
- **SDF Type** – mapping according to TCN/UIC address, or mapping is not SDF
- **SDF Index** – index in TCN composition or UIC address
- **ETH Byte Offset** – destination byte offset in Ethernet port
- **ETH Bit Offset** – bit offset in destination byte
- **Bit Len** – mapped data length
- **Copy Type** – type of copying
- **Data / Freshness / Internal** – type of mapped data
- **Paired** – flag determining whether data parts of orientation pairs are transmitted

### 7.13. Error List

This tab provides display list of errors, identified during database validation. Target line contains two columns *Location* and *Description*. In *Location* column the name of tab with error is displayed, in *Description* column the description of the error is displayed. By clicking on particular error line, the application will automatically switch to tab where error occurred.

**Error List**

<table>
<thead>
<tr>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Definition</td>
<td>Database name is empty</td>
</tr>
<tr>
<td>Database Definition</td>
<td>User type is empty</td>
</tr>
</tbody>
</table>

Fig. 8 - Error List
8. Internal structures and internal marshalling

Six internal structures are defined on TCN-GW-WEM_REF:

- four exported structures, where information about current train composition, gateway status and list of domain descriptors is saved
- two structures controlling marshalling and TCN-GW-WEM_REF behaviour on WTB

Items from exported structures can be mapped to import marshalling, items from imported structures can be mapped to export marshalling.

8.1. Internal import marshalling

Bit length of exported structures shown in table is maximum bit length. User can define shorter bit length (for example, partial import of particular unit).

All multi-byte numerical entries are little-endian ordered.

NADI exported structure:

<table>
<thead>
<tr>
<th>Title</th>
<th>Type</th>
<th>Length</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Enum8</td>
<td>8 bit</td>
<td>NADI status:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = invalid or missing (Other options are meaningless)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = actual</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = confirmed</td>
</tr>
<tr>
<td>Topo_counter</td>
<td>Unsigned8</td>
<td>8 bit</td>
<td>Current topography identification that takes values 1 to 63</td>
</tr>
<tr>
<td>Inaug_frame_ver</td>
<td>Unsigned8</td>
<td>8 bit</td>
<td>Inauguration frame version</td>
</tr>
<tr>
<td>R-data version</td>
<td>Unsigned8</td>
<td>8 bit</td>
<td>R-Telegram version</td>
</tr>
<tr>
<td>TCN_addr</td>
<td>Unsigned8</td>
<td>8 bit</td>
<td>TCN address of own gateway</td>
</tr>
<tr>
<td>TCN_orient</td>
<td>Enum8</td>
<td>8 bit</td>
<td>Relative TCN orientation of own gateway against TCN master</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = orientation is not defined</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = gateway orientation is same</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = gateway orientation is opposite</td>
</tr>
<tr>
<td>TCN_nr_nodes</td>
<td>Unsigned8</td>
<td>8 bit</td>
<td>Current number of TCN gateways</td>
</tr>
<tr>
<td>TCN_bottom</td>
<td>Unsigned8</td>
<td>8 bit</td>
<td>TCN address of bottom gateway</td>
</tr>
<tr>
<td>TCN_top</td>
<td>Unsigned8</td>
<td>8 bit</td>
<td>TCN address of top gateway</td>
</tr>
<tr>
<td>UIC_addr</td>
<td>Unsigned8</td>
<td>8 bit</td>
<td>UIC address of vehicle itself</td>
</tr>
<tr>
<td>UIC_orient</td>
<td>Enum8</td>
<td>8 bit</td>
<td>Relative UIC orientation of vehicle itself against UIC reference direction</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = orientation is not defined</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = vehicle orientation is same</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = vehicle orientation is opposite</td>
</tr>
<tr>
<td>UIC_nr_vehs</td>
<td>Unsigned8</td>
<td>8 bit</td>
<td>Current number of UIC vehicles</td>
</tr>
<tr>
<td>UIC_top</td>
<td>Unsigned8</td>
<td>8 bit</td>
<td>Top vehicle UIC address</td>
</tr>
<tr>
<td>Title</td>
<td>Type</td>
<td>Length</td>
<td>Meaning</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------</td>
<td>--------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UIC_addr_leading</td>
<td>Unsigned8</td>
<td>8 bit</td>
<td>Leading vehicle UIC address, 0 = there is no leading vehicle in the composition, UIC_addr = Vehicle with UIC_addr is a leading vehicle</td>
</tr>
<tr>
<td>Gateway_descriptors</td>
<td>Array_desc [nr_gws]</td>
<td></td>
<td>Gateway descriptor field</td>
</tr>
<tr>
<td>TCN_addr</td>
<td>Unsigned8</td>
<td>8 bit</td>
<td>Gateway TCN address</td>
</tr>
<tr>
<td>UIC_addr</td>
<td>Unsigned8</td>
<td>8 bit</td>
<td>UIC address of vehicle with gateway</td>
</tr>
<tr>
<td>R-type</td>
<td>Enum4</td>
<td>4 bits</td>
<td>Type of transmitted R-telegram, 0 = Type is not defined, 1 = R1-Telegram, 2 = R2-Telegram, 3 = R3-Telegram</td>
</tr>
<tr>
<td>R-version</td>
<td>Enum4</td>
<td>4 bits</td>
<td>Version of transmitted R-telegram</td>
</tr>
<tr>
<td>UIC_orient</td>
<td>Bitset4</td>
<td>4 bits</td>
<td>UIC consist orientation</td>
</tr>
<tr>
<td>TCN_orient</td>
<td>Bitset4</td>
<td>4 bits</td>
<td>Gateway TCN orientation</td>
</tr>
<tr>
<td>Inaug_data_len</td>
<td>Unsigned8</td>
<td>8 bit</td>
<td>Inauguration data length</td>
</tr>
<tr>
<td>Inaug_data</td>
<td>Unsigned8[ ]</td>
<td>124-byte</td>
<td>Inauguration data</td>
</tr>
</tbody>
</table>

**Exported structure Gateway status:**

<table>
<thead>
<tr>
<th>Title</th>
<th>Type</th>
<th>Length</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Life_sign_cnt</td>
<td>Unsigned32</td>
<td>32 bit</td>
<td>Gateway vitality counter</td>
</tr>
<tr>
<td>UIC_op_mode</td>
<td>Enum8</td>
<td>8 bit</td>
<td>Gateway operational mode, 0 = operational mode is not defined, 1 = operational mode leading (R1-Telegram exported), 2 = operational mode traction (R2-Telegram exported), 3 = operational mode follower (R3-Telegram exported)</td>
</tr>
<tr>
<td>TCN_op_state</td>
<td>Enum8</td>
<td>8 bit</td>
<td>TCN operational state, 0 = operational state is not defined, 1 = operational state slave, 2 = operational state strong, 3 = operational state weak, 4 = operational state passive, 5 = operational state idle</td>
</tr>
<tr>
<td>Gateway_state</td>
<td>Enum8</td>
<td>8 bit</td>
<td>Gateway state, 0 = gateway state is not defined, 1 = idle state, 2 = regular state, 3 = restricted state (inauguration in progress), 4 = passive state, 5 = single state, 6 = sleep state, 7 = init state (gateway initialized)</td>
</tr>
<tr>
<td>Inaug_state</td>
<td>Enum8</td>
<td>8 bit</td>
<td>TCN inauguration status, 0 = inauguration status is not defined, 1 = inauguration (lengthening) is allowed, 2 = inauguration (lengthening) is inhibited</td>
</tr>
<tr>
<td>TCN_conflict</td>
<td>Enum8</td>
<td>8 bit</td>
<td>Strong master conflict indication, 0 = conflict is not defined, 1 = no strong master conflict, 2 = strong master conflict</td>
</tr>
<tr>
<td>Redund_status</td>
<td>Enum8</td>
<td>8 bit</td>
<td>Redundancy state</td>
</tr>
<tr>
<td>Title</td>
<td>Type</td>
<td>Length</td>
<td>Meaning</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Buffer_exchng</td>
<td>Unsigned8[ ]</td>
<td>1024-byte</td>
<td>Buffer for data exchange between consist busses</td>
</tr>
<tr>
<td>Node_report</td>
<td>Bitset8</td>
<td>8 bit</td>
<td>WTB channel status, WTB relays settings and WTB composition change (IEC 61375-2-1 standard, part 5.5.2.2)</td>
</tr>
<tr>
<td>User_report</td>
<td>Bitset8</td>
<td>8 bit</td>
<td>User bits definition (IEC 61375-2-1 standard, part 5.5.2.3) UIC 556 standard prohibits usage of these bits.</td>
</tr>
<tr>
<td>Master_report</td>
<td>Bitset8</td>
<td>8 bit</td>
<td>Total WTB line disturbance, direction and inauguration state (IEC 61375-2-1 standard, part 5.5.2.5. Full meaning only for endpoint nodes and TCN master)</td>
</tr>
<tr>
<td>WTB_A1_cnt_tx</td>
<td>Unsigned32</td>
<td>32 bit</td>
<td>WTB channel A1 – sent frames counter</td>
</tr>
<tr>
<td>WTB_A1_cnt_rx</td>
<td>Unsigned32</td>
<td>32 bit</td>
<td>WTB channel A1 – received frames counter</td>
</tr>
<tr>
<td>WTB_A1_cnt_err</td>
<td>Unsigned16</td>
<td>16 bit</td>
<td>WTB channel A1 – error frames counter</td>
</tr>
<tr>
<td>WTB_A1_cnt_tmo</td>
<td>Unsigned16</td>
<td>16 bit</td>
<td>WTB channel A1 – timeout counter</td>
</tr>
<tr>
<td>WTB_A2_cnt_tx</td>
<td>Unsigned32</td>
<td>32 bit</td>
<td>WTB channel A2 – sent frames counter</td>
</tr>
<tr>
<td>WTB_A2_cnt_rx</td>
<td>Unsigned32</td>
<td>32 bit</td>
<td>WTB channel A2 – received frames counter</td>
</tr>
<tr>
<td>WTB_A2_cnt_err</td>
<td>Unsigned16</td>
<td>16 bit</td>
<td>WTB channel A2 – error frames counter</td>
</tr>
<tr>
<td>WTB_A2_cnt_tmo</td>
<td>Unsigned16</td>
<td>16 bit</td>
<td>WTB channel A2 – timeout counter</td>
</tr>
<tr>
<td>WTB_B1_cnt_tx</td>
<td>Unsigned32</td>
<td>32 bit</td>
<td>WTB channel B1 – sent frames counter</td>
</tr>
<tr>
<td>WTB_B1_cnt_rx</td>
<td>Unsigned32</td>
<td>32 bit</td>
<td>WTB channel B1 – received frames counter</td>
</tr>
<tr>
<td>WTB_B1_cnt_err</td>
<td>Unsigned16</td>
<td>16 bit</td>
<td>WTB channel B1 – error frames counter</td>
</tr>
<tr>
<td>WTB_B1_cnt_tmo</td>
<td>Unsigned16</td>
<td>16 bit</td>
<td>WTB channel B1 – timeout counter</td>
</tr>
<tr>
<td>WTB_B2_cnt_tx</td>
<td>Unsigned32</td>
<td>32 bit</td>
<td>WTB channel B2 – sent frames counter</td>
</tr>
<tr>
<td>WTB_B2_cnt_rx</td>
<td>Unsigned32</td>
<td>32 bit</td>
<td>WTB channel B2 – received frames counter</td>
</tr>
<tr>
<td>WTB_B2_cnt_err</td>
<td>Unsigned16</td>
<td>16 bit</td>
<td>WTB channel B2 – error frames counter</td>
</tr>
<tr>
<td>WTB_B2_cnt_tmo</td>
<td>Unsigned16</td>
<td>16 bit</td>
<td>WTB channel B2 – timeout counter</td>
</tr>
</tbody>
</table>

**Note:** Content of data exchange buffer is transferred from the item with same name, in imported structure Gateway control.

### Exported structure Domain descriptors:

<table>
<thead>
<tr>
<th>Title</th>
<th>Type</th>
<th>Length</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nr_domains</td>
<td>Unsigned32</td>
<td>32 bit</td>
<td>Number of domains</td>
</tr>
<tr>
<td>Domain_descs</td>
<td>Array_desc [Nr_domains]</td>
<td>Domain descriptor array</td>
<td></td>
</tr>
<tr>
<td>Domain_Id</td>
<td>Unsigned32</td>
<td>32 bit</td>
<td>Domain identification:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = Safe Altera</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = Altera</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = Safe application</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 = Executed application</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 = Application update</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 = NSDB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6 = EEPROM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7 = Journal</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8 = Journal main page</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9 = Journal extra page</td>
</tr>
<tr>
<td>Version_maj</td>
<td>Unsigned8</td>
<td>8 bit</td>
<td>Version major</td>
</tr>
<tr>
<td>Version_min</td>
<td>Unsigned8</td>
<td>8 bit</td>
<td>Version minor</td>
</tr>
</tbody>
</table>


### Exported structure Marshalling status:

<table>
<thead>
<tr>
<th>Název</th>
<th>Typ</th>
<th>Délka</th>
<th>Význam</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MVB</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Export_status</td>
<td>Enum8</td>
<td>8-bitů</td>
<td>Export marshalling status</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = status of export marshalling is not defined</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = export marshalling is switch off</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = export marshalling is switch on</td>
</tr>
<tr>
<td>Life_tmo_status</td>
<td>Enum8</td>
<td>8-bitů</td>
<td>Lifetime timeout status</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = lifetime timeout status is not defined</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = lifetime timeouted (lifetime signal is not received)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = lifetime ok (lifetime signal received)</td>
</tr>
<tr>
<td>Trusted_status</td>
<td>Enum8</td>
<td>8-bitů</td>
<td>Trusted status</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = trusted status is not defined</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = bus is trusted</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = bus is observed</td>
</tr>
<tr>
<td><strong>Ethernet</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Export_status</td>
<td>Enum8</td>
<td>8-bitů</td>
<td>Export marshalling status</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = status of export marshalling is not defined</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = export marshalling is switch off</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = export marshalling is switch on</td>
</tr>
<tr>
<td>Life_tmo_status</td>
<td>Enum8</td>
<td>8-bitů</td>
<td>Lifetime timeout status</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = lifetime timeout status is not defined</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = lifetime timeouted (lifetime signal is not received)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = lifetime ok (lifetime signal received)</td>
</tr>
<tr>
<td>Trusted_status</td>
<td>Enum8</td>
<td>8-bitů</td>
<td>Trusted status</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = trusted status is not defined</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = bus is trusted</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = bus is observed</td>
</tr>
</tbody>
</table>

### 8.2. Internal export marshalling

Control of TCN-GW-WEM_REF using Gateway control structure is possible from only one consist bus which is set as trusted. Bus trusted status is possible to check from Marshalling status structure.
<table>
<thead>
<tr>
<th>Title</th>
<th>Type</th>
<th>Length</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>UIC_leading_req</td>
<td>Enum8</td>
<td>8 bit</td>
<td>Requested leading state</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = leading state is not defined</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = request to set leading in direction 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = request to set leading in direction 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 = request to cancel leading</td>
</tr>
<tr>
<td>TCN_op_state_req</td>
<td>Enum8</td>
<td>8 bit</td>
<td>Requested TCN operation state</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = request operation state is not defined</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = request slave operation state</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = request strong operation state</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 = request weak operation state</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 = request passive operation state</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5 = request idle operation state</td>
</tr>
<tr>
<td>Inauguration_ctrl</td>
<td>Enum8</td>
<td>8 bit</td>
<td>Inauguration control</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = inauguration control is not defined</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = request allow of inauguration (lengthening of composition)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = request inhibit of inauguration (lengthening of composition)</td>
</tr>
<tr>
<td>UIC_vehicle_ID</td>
<td>Unsigned8</td>
<td>5-byte</td>
<td>Vehicle UIC address</td>
</tr>
<tr>
<td>Buffer_exchng</td>
<td>Unsigned8</td>
<td>10-24</td>
<td>Buffer for data exchange between consist busses</td>
</tr>
</tbody>
</table>

**Note:** Vehicle ID can be set from NSDB (static vehicle properties) or via Imported Controls structure. If the UIC vehicle ID is not set in NSDB, TCN-GW-WEM_REF will not inaugurate on WTB after start, it waits for setting of UIC vehicle ID via imported Control structure.

UIC vehicle ID can be changed during run via imported Controls structure; UIC vehicle ID change will cause UIC inauguration on WTB.

During setting UIC vehicle ID in to inauguration frame, no big/little endian conversion takes place. UIC vehicle ID is understood as a 5 byte bytestream.

Content of data exchange buffer is transferred to the item with the same name, in imported structure Gateway status.

### Imported structure Marshalling control:

<table>
<thead>
<tr>
<th>Title</th>
<th>Type</th>
<th>Length</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Export_ctrl</td>
<td>Enum8</td>
<td>8 bit</td>
<td>Export marshalling control</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = export marshalling control is not defined</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = request for turn off running of export marshalling</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = request to turn on running of export marshalling</td>
</tr>
<tr>
<td>Lifetime_cnt</td>
<td>Unsigned32</td>
<td>32 bit</td>
<td>Remote device vitality counter (VCU)</td>
</tr>
<tr>
<td>Trusted_ctrl</td>
<td>Enum8</td>
<td>8-bitu</td>
<td>Trusted/observed control</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = trusted/observed control is not defined</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = bus requested trusted</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = bus requested observed</td>
</tr>
<tr>
<td>Ethernet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Export_ctrl</td>
<td>Enum8</td>
<td>8 bit</td>
<td>Export marshalling control</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = export marshalling control is not defined</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = request for turn off running of export marshalling</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = request to turn on running of export marshalling</td>
</tr>
<tr>
<td>Lifetime_cnt</td>
<td>Unsigned32</td>
<td>32 bit</td>
<td>Remote device vitality counter (VCU)</td>
</tr>
<tr>
<td>Title</td>
<td>Type</td>
<td>Length</td>
<td>Meaning</td>
</tr>
<tr>
<td>-------------</td>
<td>----------</td>
<td>--------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>Trusted_ctrl</td>
<td>Enum8</td>
<td>8-bitu</td>
<td>Trusted/observed control</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = trusted/observed control is not defined</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = bus requested trusted</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 = bus requested observed</td>
</tr>
</tbody>
</table>

**Note:** Turn off export marshalling stops export on WTB. This setting doesn't affect internal export marshalling.

Vitality counter timeout is set in configuration of particular MVB bus. If the vitality counter does not change during defined period export on WTB is stopped.

Particular trusted request cancels previous trusted states on other buses.
9. ADSManager

9.1. Program description

Program provides loading and retrieval of domain files from/to TCN-GW-WEM_REF and for domain management. Domain is considered as part of memory, which is defined by offset and size.

Program itself consists of several parts:

- Menu
- List of devices
- Device search
- Status bar

Fig. 9 - Description of the screen with target device detected on Ethernet
9.2. Menu

9.2.1 File

**Exit** program termination

**Search** searching for device on selected communication channel

**Search cancel** cancel device searching

9.2.2 Option

![Fig. 10 - Preferences](image)

**Communication channel** – communication channel type (for target device only Ethernet)

- **Port** – Ethernet port number
- **COM** – serial port number
- **View Stamp Data** – when the box is checked, the domain CRC will be displayed in domain list
- **Domain File Source Dir** – folder from where the .dom files will be read
- **Domain File Destination Dir** – folder where the .dom files will be saved

9.3. List of devices

All found devices are displayed on selected communication line.

Five basic operations with device can be performed when clicking on particular device:

- **Domain services** – reading the list of domains from the device (see below)
- **Switch to service mode** – switching the device into the service mode; Safe application starts
- **Reset device** – resetting device
- **Load XML** – displaying window for reading XML script. XML script serves, for example, for multiple domain update
- **Ethernet settings** – change of IP address, mask or default IP gateway

![Fig. 11 - Main window with found devices](image)
9.4. List of domains

**Main table** The main table shows the actual list of domains.
If the domain is active, the line is blue, when damaged or free, the line is red. Otherwise the line is black.

- **ID** – the domain ID
- **Memory** – type of used memory (ROM, Flash, …)
- **User rights** – RE = reading, WR = writing, DW = download, SR = sequential reading, RU = rights to run, DI = dictionary with parameters
- **Version** – domain version
- **Description** – domain description

**Sub-table** The sub-table shows detail parameters of particular domain. Sub-table content varies according to line selected in main table.

- **File** – file name specified by AMiT company
- **DB name** – database name specified by user in GTWConfigurator tool (Tab Database definition)
- **MAC** – network interface MAC address
- **Offset** – retrieved domain offset (0 to DMNS – 1)
- **DMNS** – overall number of domains included in device
- **Max Size** – domain total size
- **Current Size** – occupied domain size
- **Free Size** – domain free size in %
- **Time** – time of domain creation/ change
- **Usage** – domain occupation type (empty, debug, internal, release)
- **State** – domain data status
- **Appl ID** – bottom Word = HWID, Upper Word = PID
- **Hwid ID** – station identifier
- **PID** – process/application identifier.
- **Cfg ID** – station modification identifier
- **Cfg Mask** – Cfg ID mask for given domain
- **Sig** – domain signature (specifies meaning of domain content)

### 9.4.1 Operations over domains

- **Read Domain** – domain contents retrieval. Provides retrieval of domain content from the device and its storing to *.dom type file. In case of NSDB type domain, it is possible to edit file directly via program GTWConfigurator. For Journal type domain – the text form *.log can be displayed.
- **Download Domain** – loading the domain into device. Provides loading of selected domain into the device. After successful downloading, the user is prompted to activate data. When refused, the domain will be inactive up to next device restart.
- **Compute CRC** – retrieving/calculating CRC. When user activates the item, program first tries to read CRC from the device, if available. If unavailable, calculation is made. User can also force the calculation. When presses the Compute button in window for calculation.
- **Compute SHA1** – retrieving/calculating SHA1. The similar process as for CRC calculation.

### 9.5. Status bar

The status bar shows the result of recent operation with device.

### 9.6. XML script

XML script contains list of domains for update. Script is provided by AMiT company.
XML script can be loaded either from GUI or it is possible to run ADSManager in command line mode.

### 9.6.1 GUI mode

Window for working with XML scripts can be invoked from context menu “Load XML” after the device is found.

- **Browse script** – browse script
- **Clear** – clear the command line window
- **Run script** – run script

![GUI XML script - domain update](image)

**Fig. 13 - GUI XML script – domain update**

### 9.6.2 Command line mode

Description of ADSManager in command line mode.

**ADS CMD** format:

```
ADSManager -c<port> -x <xml>
ADSManager -c<port> -d<id> <dom>
```

**Parameters:**

- `-h` – help
- `-c<number>` – serial channel number
- `-i<IP>` – IP address, possibly also device UDP port
- `-x <*.xml>` – name of XML file with script
- `-d<number> <*.dom>` – domain ID and DOM file for update
Fig. 14 - Command line XML script – domain update
10. Loading domains to device

This chapter describes how to load the actual domain version into new device, or how to perform the domain upgrade for used device. Domain can be considered as contiguous memory block.

10.1. TCN-GW-WEM_REF domains

TCN-GW-WEM_REF on target device contains following domains:

<table>
<thead>
<tr>
<th>Domain</th>
<th>Manufacturing status</th>
<th>Description / Upgrade</th>
</tr>
</thead>
<tbody>
<tr>
<td>WTB Controller Safe</td>
<td>Loaded</td>
<td>WTB controller – secure (backup) copy. Upgrade Yes</td>
</tr>
<tr>
<td>WTB Controller</td>
<td>Free</td>
<td>WTB controller – runtime copy. Upgrade Yes</td>
</tr>
<tr>
<td>Safe Application</td>
<td>Loaded</td>
<td>Safe application      Upgrade: Yes</td>
</tr>
<tr>
<td>Executed application</td>
<td>Free</td>
<td>Just running application Upgrade: Information only</td>
</tr>
<tr>
<td>Application update</td>
<td>Free</td>
<td>Valid application     Upgrade: Yes</td>
</tr>
<tr>
<td>NSDB</td>
<td>Free</td>
<td>Configuration database Created by user Upgrade: Yes</td>
</tr>
<tr>
<td>EEPROM</td>
<td>Loaded</td>
<td>Gateway runtime configuration (Ethernet, TCN quick reinsertion variables) Upgrade Yes</td>
</tr>
<tr>
<td>Journal – Main Page</td>
<td>Journal</td>
<td>Operating journal.    Upgrade No</td>
</tr>
<tr>
<td>Journal – Extra Page</td>
<td>Journal</td>
<td>Operating error journal Upgrade No</td>
</tr>
</tbody>
</table>

All domains with exception of NSDB are supplied by AMiT company. Actual versions are available here:


The NSDB domain is created by user through GTWConfigurator PC tool.

Only correct domain file can be loaded into appropriate domain. File headers are controlled during loading. Wrong data cannot be loaded.

Domain file names:

<table>
<thead>
<tr>
<th>Domain</th>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application update</td>
<td>wtb-wem_ref_1_a_100.dom</td>
</tr>
<tr>
<td>Safe application</td>
<td>wtb-wem_ref_1_s_100.dom</td>
</tr>
<tr>
<td>WTB controller</td>
<td>wtb-wem_ref_1_f_100.dom</td>
</tr>
<tr>
<td>WTB controller safe</td>
<td>wtb-wem_ref_1_e_100.dom</td>
</tr>
<tr>
<td>NSDB</td>
<td>Created by user</td>
</tr>
</tbody>
</table>
10.2. Loading of new device

Target device comes from production with station firmware loader (marked as Safe Application) and safe WTB controller copy. It is necessary to load Application and WTB controller before the first use.

To load files, it is necessary to download, from web address mentioned in chapter 10.1, the ADSManager program, actual versions of application program and WTB controller.

**Procedure**

- Connect device to Ethernet network with connected PC
- Launch the ADSManager program
- Set in the Menu – Option – Preferences
  - Communication channel to **Ethernet**
  - Port to value **75**
  - Optionally you can choose default folders with *.dom files.
- Power on target device
- Start auto detection in main program window
- In Device list, the target device will appear (if not, click Search again). Premature search termination can be done with **Esc** key.
- Launch the Domain services by click on device (context menu is activated by right mouse button)
- Click on WTB controller domain and start the Download domain service from the context menu. After loading, the device restarts itself
- Click on Application Update domain and start the Download domain service from the context menu. After loading, the device restarts itself

Now the device is ready for loading domain with configuration created by GTWConfigurator PC tool.

10.3. Loading of configuration domain

**Procedure**

- Connect device to Ethernet network with connected PC
- Launch the ADSManager program
- Set in the Menu – Option – Preferences
  - Communication channel to **Ethernet**
  - Port to value **75**
  - Optionally you can choose default folders with *.dom files.
- Power on target device
- Start auto detection in main program window
- In the Device list, the target device will appear (if not, click Search again)
- Launch the Domain services by click on device (context menu is activated by right mouse button)
Click on NSDB domain and start the Download domain service from the context menu. After loading, the device restarts itself.

Now the device is completely ready for work.

10.4. Domain upgrade

When any domain needs to be upgraded on device in operation, follow the procedure below.

**Procedure**
- Connect device to Ethernet network with connected PC
- Launch the **ADSManager** program
- Set in the Menu – Option – Preferences
  - Communication channel to **Ethernet**
  - Port to value **75**
  - Optionally you can select default folders with *.dom files.
- Power on target device
- Start auto detection in main program window
- In the Device list, the target device will appear (if not, click Search again)
- Launch the Domain services by click on device (context menu is activated by right mouse button)
- Click on domain which is to be upgraded and start the Download domain service from the context menu. After loading the domain, the device restarts itself