UIC CODE 556

5th edition, August 2009

Translation

OR

Information transmission in the train (train bus)

Transmission d'informations dans le train (bus de train)
Informationsübertragung im Zug (Zugbus)

UNION INTERNATIONALE DES CHEMINS DE FER
INTERNATIONALER EISENBAHNVERBAND
INTERNATIONAL UNION OF RAILWAYS

Leaflet to be classified in Volumes:

V - Rolling stock

VI - Traction

Application:

With effect from 1. August 2009
All members of the International Union of Railways

Record of updates

1st edition, June 1998	First issue, was not published in the UIC Code.
2nd edition, May 1999	Redactional corrections without modification of the contents, addition of new telegrams in Appendices 1 and 2.1 to 2.3, first issue of Appendices 5.3, 7, 8 and 9 and approval of new train bus nodes; has not been published in the UIC Code.
3rd edition, October 2004	Adaptation to the editor's guide M1 coordination and insertion of the new content of the applications traction and doors; insertion of new appendix G (prev. 7) "Homologation procedure of train bus nodes", editorial changes; has not been published in the UIC Code.
4th edition, August 2005	Publication in electronic format with Appendices A, B, E, I and J published on the Internet site
5th edition, August 2009	Error corrections and additions in the text part (structure of the E-telegram) as well as in the appendices A, B, C.1, C.2, E.1, I, J, revised appendix G, new appendix L (safe data transmission)

The person responsible for this leaflet is named in the UIC Code

Contents

Summary

This leaflet defines the requirements for data transmission equipment on RIC coaches.

The requirements may also be applied to other coaches, driving trailers, traction vehicles and multiple units.

The purpose of this leaflet is to:

- fully document the requirements of all users, align them and set them out in standard form,
- provide guidelines for the technical solution adopted for the train bus and the physical characteristics of the transmission cable,
- Lay down and specify the details of the requirements of the various UIC Committes as listed in Appendix A - page 35.

Concrete train bus applications for certain functionalities are not dealt with in *UIC Leaflet 556*. They are contained in other leaflets administered by the UIC bodies responsible for the respective vehicle components.

1 - General

The object of the train bus is to transmit information of all types from any vehicle of a train or multiple units in passenger service into one or more other vehicles of the same train.

With the help of the train bus

- the operating process should be made more flexible by increased and differentiated remote control possibilities,

- the work of the Operating Department staff should be made easier,

- the safety of passenger traffic should be increased,

- additional functionalities should be opened up in passenger traffic as, for example,

  - automatic brake testing,
  - comprehensive information and service for the passengers,
  - general use of diagnostic techniques in the vehicles,
  - saving of energy.

2 - Scope

2.1 - Basis

The present requirements apply to a **homogenous** train formation, in which in the undamaged condition all vehicles are either completely bus compatible or only suitable for train inauguration and participate in bus traffic.

2.2 - Special cases

The equipment should be partly usable if cable vehicles are fitted between bus capable vehicles, which do not actively take part in the bus traffic, but allow the unhindered transmission of information within the train composition.

If passenger trains are made up of individual coaches or vehicles with damaged bus connections, then the conventional functions of the remote control and information as specified in *UIC Leaflet 558* (see Bibliography - page 236) are controlled by it and not by the train bus.

In detail these are:

- remote control of the doors as specified in *UIC Leaflet 560*;
- remote control of the lighting as specified in *UIC Leaflet 555*;
- passenger address throughout the train by means of loudspeaker equipment as specified in *UIC Leaflet 568* (see Bibliography - page 236).

Should, in this special case, further functions be necessitated over the train bus, then operating regulations should be introduced and application side precautions taken.

On trains made up of vehicles that do not have transmission cables the operation should be done conventionally without using the train bus.

3 - Laying down the Standard

3.1 - Classification of *UIC Leaflet 556*

This Leaflet has the function shown in Fig. 1 of a connecting element between the functions of the technical applications (application) in passenger coaches, traction units and multiple unit trains e.g. for traction control, brake control, door control, passenger services, diagnostics etc. and the data transmission specified in *IEC Standard 61375 Train Communication Network TCN* for the train bus - Wire Train Bus (WTB) (see List of abbreviations - page 232 and Bibliography - page 236).

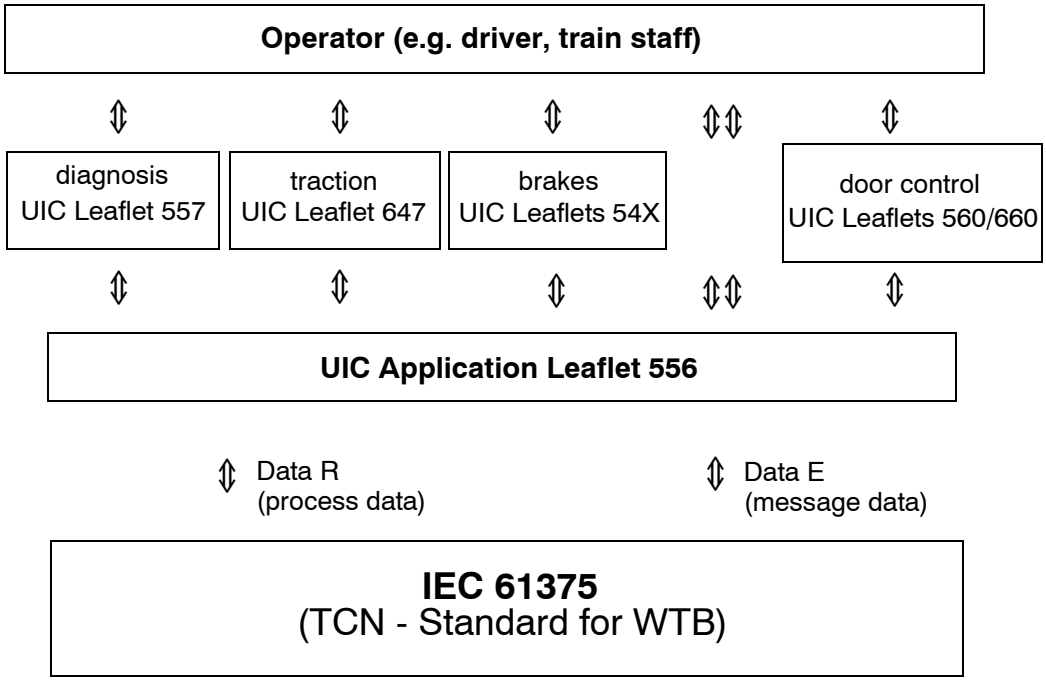


Fig. 1 - UIC-Leaflet 556 as connecting element between IEC Standard 61375 (TCN) and the functions of the UIC applications

The functionalities of the application of the UIC train bus are shown in Appendix A - page 35. It is not possible in most cases to allow the signals or data transmitted over the train bus to directly affect the individual control systems or vehicle components or to obtain signals from these. Signals which contain the necessary information must, therefore, be defined with a view to the use of the train bus for the individual functionalities. These signals must be translated by the application, in order to achieve the necessary independence from concrete technical solutions. In addition it should be specified what the time response for the individual pieces of data must be and how the application should behave when faced with irregularities and defects in the data traffic.

3.2 - Train Communication Network - TCN

The *IEC-Standard 61375 Train Communication Network TCN* specifies the data transmission in trains through the train bus (Wire Train Bus). *UIC Leaflet 556* is based on this standard. Consequently it is mandatory to use the TCN standard in its Wire Train Bus (WTB) elements for the achievement of a data transmission system. The use of the data formats, protocols etc. defined in it or the functions offered through the user interface (application interface) is binding.

3.3 - *UIC Leaflet 556*

This Leaflet specifies the parameters with which the functions of *IEC 61375* are used in a standard way in the case of a "UIC" use. This ensures that all vehicles built to this specification speak the same language over the train bus.

The Leaflet also specifies all the necessary details of all the information to be exchanged on the train bus of trains and vehicles that run in international traffic (see Appendix A - page 35 and B - page 36 as weel as point C.3 - page 84. These definitions apply up to the interface, from and to the applications, and are kept so general that they are not dependent on concrete technical solutions. In addition it is specified how the time behaviour for the individual information must be and how the application must react to irregularities and defects in the data flow.

The modification procedure for the leaflet is described in point 7 - page 29.

In addition, the UIC train bus offers the possibility of achieving national or bi- or multinational applications through the use of the UIC parameters. This is explained in Appendix B - page 36 and can be done independently or in addition to the UIC application.

3.4 - Applications

The functions of the applications of the UIC train bus, that is to say the regulations of how to achieve a uniform process of generating and processing the information to be transmitted, are specified in special UIC leaflets for such applications, some of which still have to be written (see *UIC Leaflet 557* for diagnostics in passenger coaches, *UIC Leaflet 647* for traction control, *UIC Leaflet 541-5* for EP brakes and emergency brake shorting out, *UIC Leaflet 560* for doors in passenger coaches, *UIC Leaflet 568* for public address, *UIC Leaflet 176* for passenger information systems etc. see bibliography - page 236).

These UIC leaflets are prepared and updated by the UIC technical Committees responsible for these applications. These leaflets also contain the conditions for the necessary tests of these applications for conformity.

4 - Specifications for the hardware

4.1 - Bus structure

4.1.1 - Principles

The train bus must be able to operate with between 2 and 22 vehicles with a maximum of 32 train bus nodes and a transmission cable length (source ⇔ sink) of up to 850 m.

At least one train bus node is planned per vehicle. For redundancy purposes it is recommended that two train bus subscribers are used on traction units (see Glossary - page 234) and for the bus connection of the driving cab equipment of driving trailers (this also applies to multiple units). In this case, one of the two redundant train bus subscribers is always active and the other is in cold standby mode.

In the case of failure the activation of the train bus subscriber in cold standby is done automatically. Faults and remedies should however be diagnosed, displayed and stored.

The following maximum damping values based on the bit rate should be met:

per train bus node	0,3 dB
per vehicule (with short-circuited train bus nodes)	0,5 dB
per train	20,0 dB

4.1.2 - Types of UIC train bus connection

For the type of bus connection (see Fig. 2) a distinction should be made between

- vehicles with individual train bus nodes,
- traction units, driving trailers and multiple units fitted with two redundant train bus nodes,
- trainsets, i.e. groups of vehicles with a common train bus node for several vehicles (these can also be designed as redundant, if, for example, a driving trailer function is available).

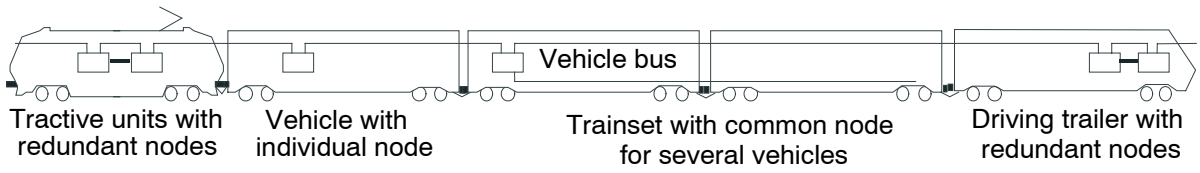


Fig. 2 - type of UIC train bus connection

A special form of the UIC train bus connection is given, if in one vehicle several different functions (traction or vehicle functions) are allotted to the train bus nodes present. In such a case care should be taken that on the train bus level no functional conflicts occur.

To meet the error tolerance requirement of point 6.2 - page 27 two synchronously working bus lines with separate cables and plug devices between the vehicles shall be provided as shown in Fig. 3.

o 4.2 - Cable for the transmission of information

4.2.1 - Physical features

The cable for the transmission of information within the trains shall be designed as a shielded 2 core cable to guarantee defect-free transmission. Two redundant cable ways (line A and line B) shall be provided (see Fig. 3).

The physical features, the principle routing and the coupling in coaches of passenger trains are described and specified in *UIC Leaflets 558 and 541-5*.

For multiple units these conditions apply also in respect of the coupling between vehicles.

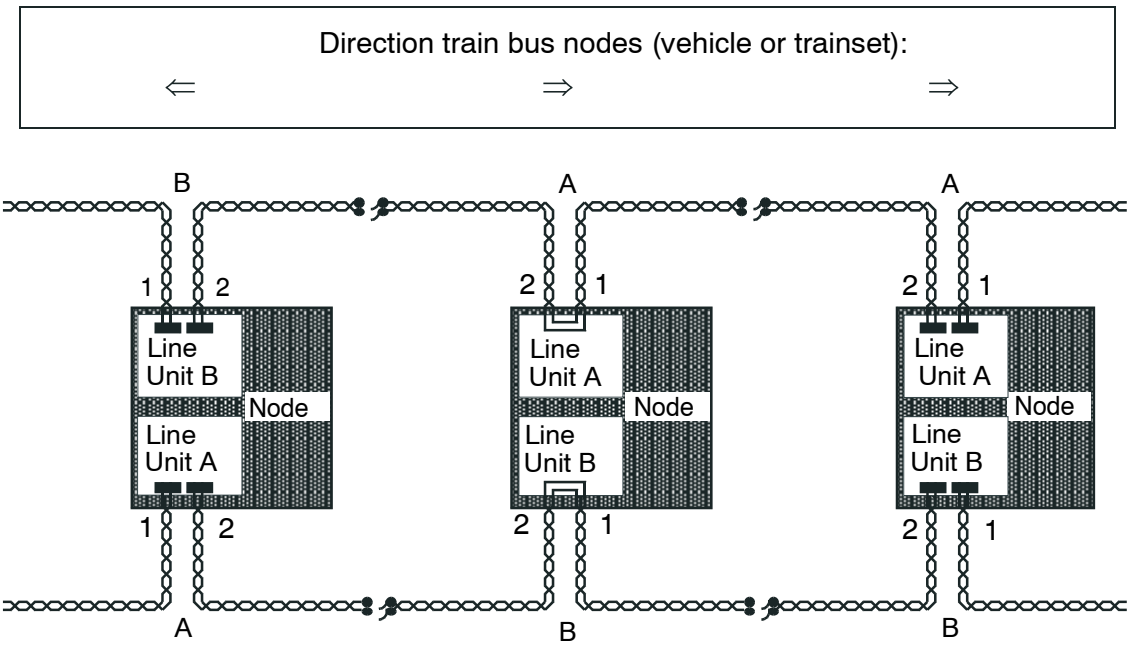


Fig. 3 - Principle coupling of the train bus nodes on the cable

4.2.2 - Additional features

- The train bus cable must be designed as electrically isolated. This means a separated-potential subscriber interface. The cable used shall be kept exclusively for information transmission.

- The shield of the train bus cable at the train bus nodes in the vehicles shall be connected to earth and may not be coupled through or connected between the vehicles.

- The shields must be earthed whenever possible.

The shield concept is specified in *UIC Leaflet 558*.

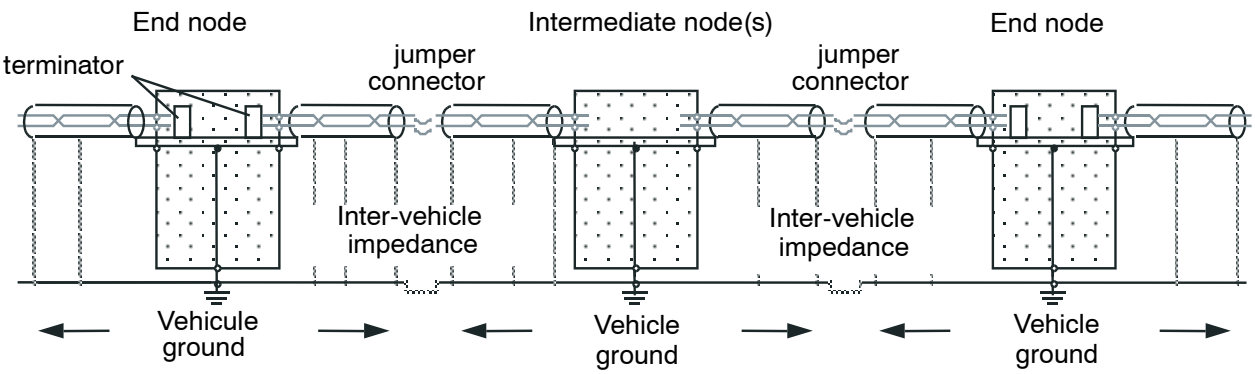


Fig. 4 - Shield concept for the trains bus nodes

4.3 - Train bus nodes

The electrical consumption of a train bus node from the battery supplied DC system of the vehicle shall not exceed 10 W when it is in sleep mode (see point 5.1 - page 10).

The train bus nodes must be able to operate in the voltage tolerance range of 0,7 to 1,25 x $U_{nominal}$. Moreover, they must meet the other conditions specified in *EN 50155* (see Bibliography - page 236).

The connection of the train bus nodes in the vehicles must result in the clear recognition of the forward, backward, right and left directions of the train as specified in Fig. 5 - page 9.

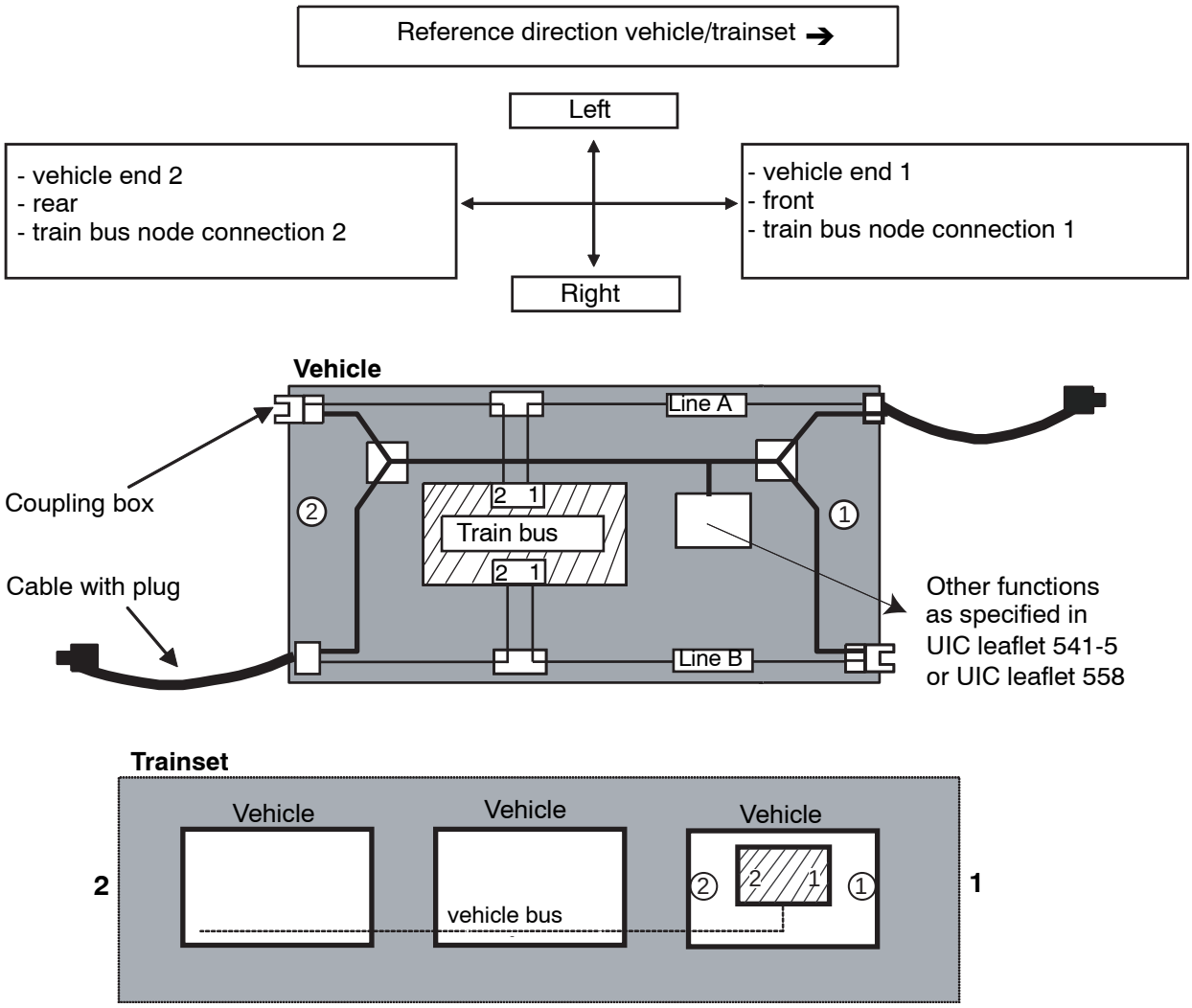


Fig. 5 - Installation of the train bus node in the vehicle/trainset (plan view)

5 - Specification of the software

5.1 - Operating conditions

The train bus nodes in an individual vehicle should be able to take up the following conditions:

Operating conditions		Properties
N°	Designation	
0	Off	A train bus node is not able to work when the voltage of the DC system of the vehicle concerned falls below 70% of the nominal voltage.
1	Sleep mode	The train bus node of a vehicle can receive signals on the train bus cable, e.g. inauguration commands.
2	Inauguration	The train bus nodes of the individual vehicles are polled from the TCN master. They export the specified vehicle properties (see point C.2 - page 55) according to the inauguration concept (see Appendix E - page 99) as part of the inauguration frame (see point C.3 - page 84). The vehicles are numbered in the sequence defined in point C.1 - page 37. The result of the inauguration is the listing of all vehicles with their data. It is determined from the inauguration frames of all vehicles and filed in the NADI (see point C.1).
3	Full operation	Information exchange as specified in Appendix A - page 35.

A new configuration of the telegram structures because of modifications of the status of individual vehicles (leading, not leading, driven, not driven (see point 5.2 - page 14) must take place during full operation and may only restrict the usability of the bus for a maximum of 1 second (this may be a maximum of 1,4 seconds for errors and defects in the inauguration command - see point 6.2 - page 27).

The following requirements determine the transitions of the operating and internal conditions of the train bus node:

- With battery charge available, the train bus node goes into the inauguration and full service operation.

- If the train bus node is in the sleep mode (operating condition 1), the operation of the push button "coach lighting on" on the operating switch desk as specified in *UIC Leaflet 550-1* (see Bibliography - page 236) initiates the transition into the inauguration and full operation (operating conditions 2 and 3).

- If bus activity is recognised during sleep mode, the train bus node goes into the inauguration and full service operation (operating conditions 2 and 3).

- 45 minutes after the termination of battery charging throughout the train (e.g. switching off of the train line from the traction unit), the train bus goes into the sleep mode (operating condition 1) corresponding to the sleep concept of *IEC standard 61375*. 45 minutes after the loss of battery charge in the vehicle concerned, each train bus node thus sends to the train bus master the order to go to sleep.

- If the protection of the battery against deep discharge is activated (see also *UIC Leaflet 550* - see Bibliography - page 236), the train bus node is switched off (operating condition 0), since all remote-controlled loads are likewise switched off.

- For emergency operation the train bus node can also be switched on below the lowest switch off stage of the minimal voltage with the push button "coach lighting on" on the operating desk, provided that the minimum voltage of the DC system (70% of the nominal voltage, *EN 50155*) is available.

- The UIC Mapping Server telegram "request to change to sleep mode" controls the transition between node operation and waiting in sleep mode.The DC system of the vehicle is monitored with a suitable device, in order to derive the conditions for the change of condition of the train bus node.

The following conditions are required:

- battery charging in the vehicle (available or not available);

- monitoring of the protection of the battery for deep discharge (undershooting or overshooting);

- local inauguration request by the falling edge when operating the push button "coach lighting on";

- remote inauguration request by bus activity on the train bus (is recognised by the train bus node);

- it is possible to go from the operating condition "full operation" directly into the condition "wait in sleep mode" by sending a suitable E telegram.

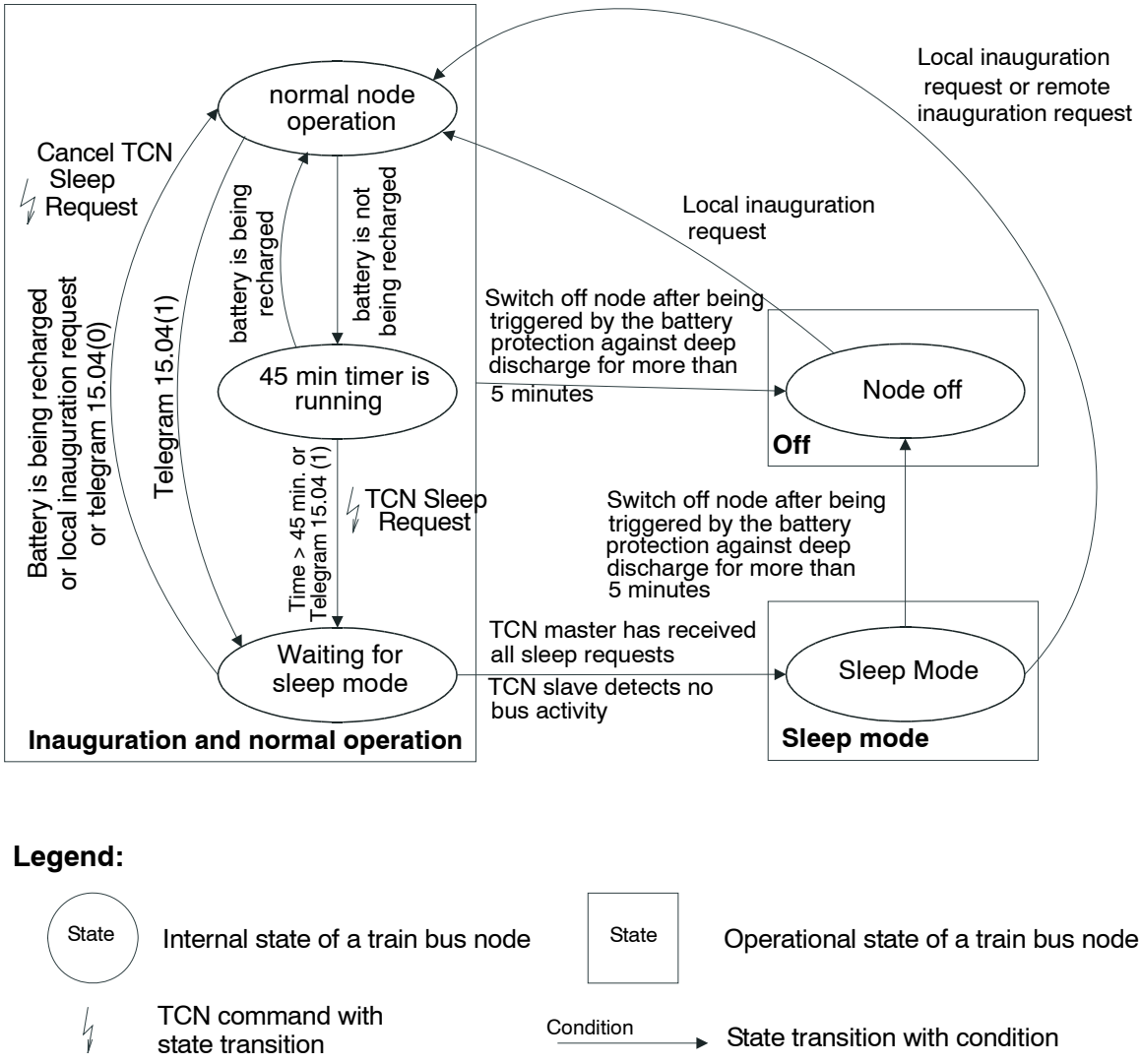


Fig. 6 - Operating conditions and condition transitions of the train bus nodes

Terms:

Minimal voltage lowest stage:	The lowest stage for loads on the battery achieved as specified in *UIC Leaflet 550* in the vehicle.
Local inauguration command:	Local operation of the push button coach lighting or a battery charge in the vehicle.
Remote inauguration command:	Bus activity recognised on the train bus.
Telegram 15.04 (X):	UIC Mapping Server Telegram "request to change to sleep mode" (1) request (0) cancel, (see Appendix A - page 35).

The following table describes the operating modes shown in Fig. 6 - page 12 and changes of mode:

Mode	Condition	Resulting mode	Explanation
Inauguration and full operation	Operation of the protection of the battery for deep discharge for more than 5 minutes	Out (mode off)	Switching off after operation of the protection for deep discharge of the battery.
Node operation	No battery charging in the vehicle	Timer runs 45 mins	After the end of battery charging a timer starts with a duration of 45 min. before the sleep mode is ordered.
	Telegram 15.04(1)	Wait for sleep mode	With the UIC Mapping Server telegram "Request to change to sleep mode change" (request), the sleep mode can be entered without timer waiting delay.
Timer runs 45 min.	Time > 45 min. or telegram 15.04(1)	Wait for sleep mode	After the timer has run out, the train bus node sends the request "sleep request" over the train bus, in order to request the sleep mode as specified in *IEC Standard 61375*.
	Battery charging in the vehicle	Node operation	If battery charge is restored the timer is reset.
Waiting for sleep mode	All sleep requirements receive (TCN-Master) *or* no bus activity (TCN-Slave)	Sleep Mode)	Corresponding to the sleep concept of the *IEC standard 61375*, the train bus master ends its activity as soon as it has received the sleep request from all bus nodes. The train bus slaves go into sleep mode, if there is no further train bus activity.
	Battery charging in the vehicle *or* local inauguration request *or* telegram 15.04(0)	Node operation	If battery charging returns, if the push button "coach lighting on" is pressed or if the UIC Mapping Server telegram "request to change in sleep mode" (cancel) is received, the sleep request is withdrawn, that is to say "sleep request cancel" is sent out. Train bus activity does not lead to the withdrawal of the sleep request with the TCN request "sleep request cancel".
Sleep mode	Local inauguration request or remote inauguration	Node operation	The train bus node is woken up by battery charging on the train line. With the push button "coach lighting on", the train bus node in a vehicle can be switched on, which then wakes up the other train bus nodes with an inauguration.
	Operation of the protection of the battery for deep discharge for more than 5 min.	Off (Node off)	Switching off after operation of the protection of the battery for deep discharge.
Off	Operation of the protection of the battery for deep discharge *or* local inauguration request	Node operation	The train bus node in a vehicle can be switched on with the push button "coach lighting on", even when the protection of the battery against deep discharge has been activated. Thus the train bus can be woken up from this vehicle.

5.2 - Inauguration

5.2.1 - Concept

A precondition for the smooth operation of the data transmission system over the UIC train bus in trains made up of individual coaches and in multiple units, which can also include trainsets, is the carrying out of the inauguration.

The inauguration is divided into the two parts given below:

- TCN inauguration to guarantee the data transmission as specified in *IEC Standard 61375*;

- UIC inauguration for the formation of the actual vehicle configuration regarding the data traffic.

The concept of the train inauguration, especially the UIC inauguration is described in detail in point C.1 - page 37.

5.2.2 - UIC Mapping Server

The purpose of the UIC Mapping Server is to guarantee the standard procedure of the inauguration with distribution of the TCN and UIC inauguration numbers throughout the train, the static and dynamic vehicle properties in the NADI (Node Address and Attribute Directory) etc. as well as to carry out the special functions such as collective and group addressing. This is a software module in the UIC train bus node, which ensures the function of the UIC train bus while considering the real vehicle configuration.

The specification of the UIC Mapping Server is given in point C.2 - page 55.

The use of the UIC Mapping Server specified in point C.2 is a precondition for guaranteeing the function of the UIC train bus according to this leaflet.

A further software model for carrying out special functions for certain information to be transmitted over the UIC train bus is the Process Data Marshalling (PDM). This guarantees the possibility of combining and assessing information and data according to the criteria specified by the application. The specification of the Process Data Marshalling is given in Appendix D - page 91.

The conformity test to be carried out for vehicles in international traffic for train bus nodes with implemented function of the UIC Mapping Server is described in detail in point 8 - page 32.

5.2.3 - Vehicle sequence

5.2.3.1 - Determining the forward, backward, left and right direction

A "reference direction train" is specified for the complete train. This is calculated from each train bus node of the UIC sequence specified with the train inauguration and stored in the NADI.

The detailed description is contained in point C.1.

5.2.3.2 - Specifying the vehicle sequence number

Independent of the technology of the "UIC train inauguration", the vehicles in the inauguration result are numbered in sequence - starting from the leading vehicle at one end of the train composition:
- 01 - 02 - 03 - ... - MN.

If the inauguration is initiated from the leading vehicle, the inauguration result should be shown there. The traction unit driver can correct the inauguration result in the leading vehicle if there are vehicles which do not take part in the inauguration in between bus-fitted vehicles e.g. vehicles which cannot be automatically inaugurated due to defects in the train bus node. Detailed specifications are given in *UIC Leaflet 647*.

If a train inauguration is carried out without the presence of a leading vehicle by the application master in any other vehicle of the train, then in conjunction with the application (e.g. initialising of functions of the passenger information and seat reservation) a correction of the inauguration must be possible on a suitable man-machine interface. Detailed specifications for this are contained in specific new UIC leaflets (to be produced).

The coach order numbers for seat reservations are independent of this numbering as part of the UIC train inauguration. The coach order numbers are static vehicle properties (see point E.2 - page 99) as far as the data traffic over the train bus is concerned. They must be announced to the system for passenger information systems.

For the specification of the vehicle with the serial number 01 the following cases which are described in detail in point C.1 - page 37 apply:

1. Leading vehicle at one end:
   This vehicle = vehicle 01

2. Leading vehicle within a train composition:
   The vehicle 01 is the end vehicle, which is located nearer to the leading vehicle, when counting the number of intermediate vehicles. If there is an identical number of vehicles, 4. applies.

3. No leading vehicle within the train composition but a traction unit at one end of the train composition:
   This vehicle = vehicle 01

4. All other cases:
   Vehicle 01 is at one of the two ends of the train and so the TCN sequence tends to correspond to the UIC sequence

Example:	TCN :	61	62	63	01	02	...
	UIC :	01	02	03	04	05	...

5.2.4 - Multiple inaugurations

If the inauguration of two or more vehicles starts at the same time, then TCN resolves the conflict automatically. As a result a completely inaugurated train is produced.

5.3 - Train structure

5.3.1 - Leading vehicle

The leading vehicle controls the movements of the train, especially the brake. It is generally occupied (except, for example, when radio-controlled). The property "leading vehicle" is transmitted to the vehicle concerned and consequently to the UIC train bus by a precise operation, in general by the activation of the driver's brake valve (see *UIC Leaflet 647*).

There is always a maximum of one leading vehicle in the train. Details of how the leading vehicle in the train is determined are given in point C.1.3.2 - page 39.

Traction units, railcars, multiple units, locomotives and driving trailers (see Glossary - page 234) can take on the property "leading vehicle".

5.3.2 - Vehicle properties

Further features must be given for the accurate description of vehicle properties. For this a distinction should be made between static and dynamic properties:

- Static properties are features built into the vehicle and thus only dependent on its class or design.
- Dynamic properties depend on the operating condition of the vehicle and/or on the way it is operated.

The static vehicle properties are listed in point E.1 - page 99. The table gives,

- whether the property depends on the train bus nodes (gateway) (= trainset) or on the individual vehicle;
- in which octet/bit of the inauguration frame this property is given to the other train bus nodes;
- which telegrams are supported by the train bus node concerned that is to say sent and/or processed, if this property is available.

The dynamic properties of the vehicles are given in point E.2 - page 99.

5.4 - Transmission cases

The individual applications of the UIC train bus, which require a general train data transmission, are given in Appendix A - page 35. The list is divided into the following functional areas:

0	Test purposes
1	Door control and door monotoring
2	Lighting control
3	Public address
4	Traction control
5	Brake functions
6	Completeness of the train
7	Central control of the climatic equipment
8	General train diagnosis
9	Control of equipment for passenger information
10	Central control of the power supply equipment
11	Others
...	
15	UIC Mapping Server

Even if not all applications are installed at the same time and from the beginning, the UIC train bus system is basically designed so that it controls all the applications shown and still has reserves for additional applications which may become necessary later.

This list is, therefore, continuously updated (see also point 7 - page 29).

In the functional range "public address" only the **control** of the participating equipment in the vehicles of the train is operated with the help of the UIC train bus. The sound transmission itself is done as specified in *UIC Leaflet 568* using wires 1-4 of the remote control and information cable specified in *UIC Leaflet 558*.

5.5 - Vehicle addressing

5.5.1 - Addresses

Each telegram carries the source address (UIC inauguration number of the sender of the information) and the destination address (UIC inauguration number of the receiver of the information).

The source address always allows determining the UIC vehicle number corresponding to the inauguration result.

Each receiver can, therefore, always ascertain from which vehicle the incoming information comes. The cases in which this knowledge is of significance for carrying out the operation are marked in Appendix A - page 35 in columns 16 and 19 by brackets.

On the users level a collective or group address as specified in points 5.5.2 and 5.5.3, can be used as the destination address besides the UIC inauguration number of a given vehicle. Thus information can be transmitted at the same time to several vehicles involved.

5.5.2 - Collective addresses

Collective addresses are used for

- static vehicle properties as specified in point E.1 - page 99 or
- dynamic vehicle properties as specified in point E.2 - page 99 or
- certain states transmitted by R telegrams.

The collective addresses used are listed in point E.3 - page 99.

5.5.3 - Group addresses

Independent of vehicle properties or because of operating events, certain vehicles of different groups can be freely selected and arranged to have defined group addresses (e.g. different journey routes of vehicle groups, travel companies in certain vehicles). The system is set up to support a minimum of 15 of these freely selectable groups. The detailed arrangements for the selecting, advising, reading and deleting of groups are contained in the relevant UIC leaflets which describe the application.

A group is characterised as follows:

- *Group number (201 to 254)*
  Each group number may only be present once in the group database.
- *Number of vehicles in the group*.
- *Group description*
  A data field with a length of 32 characters (UNICODE16) is specified for the verbal description of a group and the proposed functionality including eventual user restrictions for certain functions.
- *Vehicle identification numbers (ID) of the group members*
  Dynamic groups are independent of any train inaugurations that become necessary during the period in service. Therefore the characterisation of the group members is done through the vehicle identification numbers (ID).

In order to prevent the parallel input of modification of groups within a train, a token-bit set to "0" is used in the R data (R3 telegram, octet 10, bit 3). This token-bit indicates that group data is sent over the train bus from any subscriber. It can be placed by any subscriber. The detailed mechanism for the use of this token-bit has to be defined in the leaflets for the use of group addresses.

The reading, writing and deleting of individual groups, as well as of all groups over the UIC train bus, is done through telegrams 11.01 / 11.01A to 11.06 / 11.06A and is specified as follows:

- The management of the groups is done with confirmed E data traffic (see List of abbreviations - page 232).

- The reading of a group is done with telegrams 11.01 / 11.01A, the reading of all groups is done with telegrams 11.02 / 11.02A.

- The writing (definition) of a group is done with telegrams 11.03 / 11.03A, the writing of all groups is done with telegrams 11.04 / 11.04A.

- The cancelling of a special group is done with telegrams 11.05 / 11.05A, the cancelling of all groups is done by sending telegram 11.06 / 11.06A

5.6 - Functional addressing

5.6.1 - Internal vehicle structures

CEI 61375 is a standard which also deals with the processing of the information to be carried by the UIC train bus inside the vehicle, thus, before it is transmitted:

- production of the information in the sub-systems,

- collecting the information in the traffic memory of the train bus node after the transmission,

- distribution of the information to the sub-systems,

- processing of the information in the sub-systems.

CEI 61375 provides a vehicle bus MVB (Multifunction Vehicle Bus) corresponding to the UIC train bus WTB.

Like *UIC Leaflet 556*, *IEC 61375* also assumes that at the same time other vehicle structures can and are allowed to occur. A clear interface between the train bus nodes and vehicle internal structure must, therefore, be defined, which in all cases enables the above mentioned tasks to be achieved. This is done by defining standard function addresses for the data transmission over the train bus.

5.6.2 - Function addresses

Information is exchanged between functions. Functions are carried out by the train bus nodes themselves or by differently connected sub-systems. All occurring functions are, therefore, given function addresses that is to say they are numbered in a standard way.

IEC 61375 specifies the following number frames:

- Universal functions (e.g. UIC, underground in general, ...) 0 - 127
- Private functions (e.g. special applications, individual railways, underground, tramways, ...) 128 - 239

This leaflet specifies the following function addresses for the UIC international mandatory applications:

01	Cab
02	Train control
03	Traction unit control
04	Traction unit auxiliary operation
05	Drive
06	Brakes
07	Power supply
08	Data radio/radio modem
09	Diagnostics
10	Doors
11	Lighting
12	Public address
13	Heating and air conditioning equipment
14	Passenger information, passenger service
15	Train bus nodes (UIC Mapping Server)
16	Distance/speed measurement
17	Train protection
18	Sanitary equipment
19	Cab display
20	Tilting equipment
21	Train bus nodes (general services)
22-99	Reserves

5.7 - Telegram structures

5.7.1 - General data structures

The information to be transmitted is packed in accordance with the following rules into regular (R) or event based (E) telegrams and transported over the UIC train bus.

5.7.2 - R telegrams (regular telegrams)

R telegrams are sent every 100 ms over the UIC train bus. One vehicle always sends to all vehicles. Each vehicle takes and assesses the information addressed to it, the other information is ignored.

The basic structure as specified in the IEC standard for the UIC train bus WTB is as follows:

"Transport part"			"Application part"
	Source vehicle 1 octet	Head 2 octets	Individual information place orientated maximum 126 octets

The UIC train bus uses the following structures:

Type of telegram	Application part		
R1	Head 2 octets	Coach Information (identical to R3) 38 octets	Traction information (orders of the leading vehicle) 88 octets
R2	Head 2 octets	Coach Information (identical to R3) 38 octets	Traction Information (reports of the vehicles hauled) 88 octets
R3	Head 2 octets	Coach information 38 octets	

The types of telegram to be sent out from a vehicle are made up of combinations of the following static and dynamic vehicle properties:

Property as specified in points E.1 or E.2 - page 99	Type of telegram 1	Type of telegram 2					Type of telegram 3
38	any	1	1	1	0	1	any
39	any	1	0	1	1	1	any
49	any	1	1	1	0	0	0
50	any	1	0	0	1	1	0
512	1	0					0

Key (see also points E.1 or E.2 - page 99) :

38 Traction unit with electric drive

39 Traction unit with diesel engine drive

49 Drive of the electric traction unit can be remote-controlled by means of the train bus with control procedure 1e

50 Drive of the diesel traction unit can be remote-controlled by means of the train bus with control procedure 1d

512 Vehicle is the leading vehicle (this dynamic property is given by the UIC Mapping Server)

The make up of the 3 types of telegrams as well as the source and destination functions of the position-orientated information packed in them are given in Appendix B - page 36.

The following general structure applies to the R telegrams and provides for the applications stated in *UIC Leaflet 556*, a reserve for the future definition of further international applications as well as for applications to be used nationally only:

Structure range	***UIC Leaflet 556***		
		Octet range	**Number**
	Head information	1 - 2	2
R3-telegram	Internationally defined	3 - 30	28
	International reserve	31 - 34	4
	National reserve	39 + 40	2
R1-telegram	Internationally defined	41 - 69	29
	International reserve	70 - 84	15
	National reserve	85 - 128	44
R2-telegram	Internationally defined	41 - 71	31
	International reserve	72 - 84	13
	National reserve	85 - 128	44

The readability for the individual subscriber vehicle results from the information received in the telegram header:

- Train bus user (here "UIC");
- Type of telegram (here "R1", "R2" or "R3");
- Version number (= highest version number of the *UIC Leaflet 556* which still supports the train bus subscriber involved);
- Dynamic vehicle properties;
- Validity.

The static vehicle properties are communicated over the "Inauguration Frame" (see point C.3 - page 84) at the inauguration.

NB: The version number ensures only the readability of the R telegrams in the international area. It does not say that the sending train bus subscriber has and supports all the functions contained in the telegram. These depend much more on the static vehicle properties given in the inauguration frame (see point C.3).

The use of the national reserves (octet 39 and 40 as well as 85 - 128) is otherwise completely free, if necessary national regulations shall be observed. International and unused national reserves shall be set bitwise to "1".

5.7.3 - E Telegrams (event based telegrams)

An E telegram has the structure shown in Table 1 - page 24 and is transmitted once as a result of an event that triggered it.

There are two procedures for E telegram traffic:

- Bilateral traffic:
  The E telegram is transmitted from the sending train bus subscriber to one specific vehicle and for this purpose addressed with its UIC sequence number.

- Multicast or Broadcast traffic:
  The E telegram is transmitted from the sending train bus subscriber at the same time to several (multicast) or all (broadcast) vehicles. In this case collective addresses (e.g. 66 = all vehicles, 67 = all coaches, ...) or group addresses are used.

Since E telegrams are only sent once, TCN provides a special procedure for safe transmission.

The computer in the train bus node is informed whether the transmission is successful through reports, which run in the so-called transport layer (only in bilateral traffic). If this transport acknowledgement is not received within a certain time, the transmission is repeated up to twice more. If the transmission cannot be completed successfully even after the third attempt, the transmission is interrupted and the sending application must be informed.

Each vehicle that receives an E telegram, regardless of whether it is bilateral, multicast or broadcast traffic, must, after receipt and processing, send an answer telegram to the sending train bus subscriber. This monitors the receipt at the right time of the answer telegram or telegrams and repeats these to the sending user. The latter then knows not only that the telegram has been received, but also how the telegram was processed by the receiver. The answer telegram should be transmitted by using the reply mechanism of the TCN.

The answer telegram is for this reason produced by the destination function and not from the computer in the train bus node of the destination vehicle. The answer telegrams are, therefore, without exception defined in Appendix A - page 35. They can be recognised by the "A" after the running number (e.g. 1.7A is the answer to telegram 1.7). The answer telegrams are also E telegrams. They are constructed with the same structure as the original telegrams. The processing status must be called up from octet 9 (see Table 1 : page 24).

1. International binding E telegram												
Structure part	**Key code**		**Adress part**				**Telegram code**		**Status**		**Variable content**	**Check sum (optional)**
Octet No	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11ff**	
Content	RU code (UIC)	Reserve	Target vehicle (single or collective address)	Target function	Source vehicle	Source function	UIC telegram code		Status (or command)	Reserve	Structure according to Appendix A - page 35	CRC32 according to Appendix L
Data format	ENUM8	ENUM8	ENUM8	ENUM8	ENUM8	ENUM8	ENUM8	ENUM8	ENUM8	ENUM8		UNSIGNED32
Codes	00		NN	HH	NN	HH	HH	HH				
Examples	00		05 = veh. 05 64 = leading veh. 67 = all pass. veh.	10 = door	02 = veh. 02	10 = door	0x1004					
2. National or bilateral usable E telegram												
Structure part	**Key code**		**Address part**				**Telegram code**		**Status**		**Variable content**	**Check sum** (optional)
Octet No	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11ff**	
Content	RU code (UIC)	Reserve	Target vehicle (single or collective address)	Target function	Source vehicle	Source function	application code	UIC telegram code	Status (or command)	Reserve	Structure according to Appendix A - page 35	CRC32 according to Appendix L - page 224
Data format	ENUM8	ENUM8	ENUM8	ENUM8	ENUM8	ENUM8	ENUM8	ENUM8	ENUM8	ENUM8		UNSIGNED32
Codes	NNN		NN	HH	NN	HH	HH	HH				
Examples	80 = DB 87 = SNCF 101ff = bilat.		05 = veh. 05 64 = leading. veh. 67 = all pass. veh.	10=door	02=veh. 02	10=door	0X10	0X04				

Table 1 : Structure of the E telegram

The following code frames are specified for this:

255	Processing not possible, since the function is not available
254	Processing not possible, since the function is switched off
253	Processing not at present possible, since the function is defective
252	Processing possible later, computer at present overloaded
251	Processing possible later, function is in test mode
200-250	Reserved for information about further processing obstacles
100-199	Reserved for information about further regular types of processing
0-99	Confirmation of the proper receipt and the type of proper processing. The coding for this is given in Appendix A - page 35.

If no code is shown in Appendix A - page 35 for the regular processing then it should be coded:

1 answer telegram received and duly processed.

Further contents can also be packed in the octets 11 ff of the answer telegram. This is shown in detail in Appendix A - page 35 as well.

There is no further answer to an answer telegram.

Like R telegram information, E telegrams are also sent through function addresses directly to the function concerned. However, an E telegram can only be addressed to one function.

The use of the TCN topocounters (see point C.1.3.8 - page 53) for E telegrams is specified as mandatory, so that if there is a new inauguration in the meantime, long E telegrams, including the appropriate answer telegrams, can be transmitted correctly.

The obligatory part of the E telegram consists of the octets 1 to 10.
The E telegram header consists of the octets 1 to 8.

The following general rules apply to the production and use of E telegrams:

- The E telegram format shall always be completed on the total number of octets.

- All reserve bits and octets shall be set to "0" if they are not used.

The individual elements of the header of the E telegrams have the following meaning:

- Key code: The key code consists of two octets.
  The first contains the RU code concerned.
  For the UIC application it should be coded "00".
  The second is at present reserved.

- Address part: The address part consists of four octets. The information shown for the address part is used in order to forward the information to the competent sub-system and to enable the processing by the sub-system on the basis of the knowledge of the origin.

The terms below have the following meaning:

Destination vehicule:	UIC address of the destination vehicle or collective or group address (see Appendix A - page 35, Column 9a).
Destination function:	Function by which the information is to be processed (see Appendix A, Column 14).
Source vehicle:	UIC address of the source vehicle (see Appendix A, Column 3).
Source function:	Function from which the information was produced (see Appendix A, Column 13).

The function addresses are the same as for the R telegrams (see point 5.6.2 - page 19):

RU/UIC:	The first octet of the key code gives the UIC code of the owning RU of the information source. It gives the RUs the opportunity of making a national or bilateral use of the UIC train bus besides the international obligatory specification. The code "00" applies for international obligatory information. This and only this is given in Appendix A. It is recommended that RUs produce a similar list for national use and to exchange this, if necessary, with individual (neighbouring) RUs for possible bilateral use.
UIC telegram code:	The 7th and 8th octet in the telegram header define the UIC telegram code, which consists of 4 hexadecimal figures. For each E telegram given in Appendix A, Column 13, a separate UIC telegram code should be given, from which the receiving train bus subscriber, the type of the sub-system and the type of the processing can be derived on the basis of tables deposited there. With the telegram code the "most significant octet" is the first to be transmitted (octet 7 before octet 8).
Application identification:	The first octet of the UIC telegram code (7th octet of the E telegram) is used as the application identification for national/private use of the UIC train bus. National regulations can be made for this.
Status :	In the 9th octet of the E telegrams "Status" different contents (e.g. 1 = ON!, 0 = OFF!) can be packaged corresponding to the definitions in Appendix A. The coding is not subject to any regulation. If no different contents are passed on, it should be coded "1".

Longer E telegrams are divided into pieces of suitable length (e.g. 128 bytes).

6 - Reliability

6.1 - Failure rate

The consistency of the functional ability of the equipment participating in the bus operation over time is of particular importance for defect-free operation.

6.2 - Error tolerance

If there are errors in traction units and driving trailers

> the 1st error that occurs should not result in any lasting effect on the data exchange on the train bus. Temporary impairment shall not last longer than 1,4 seconds and must automatically be overcome.

If there are errors in the other vehicles

> the maximum result of the first error shall be that a vehicle can no longer be reached over the train bus. This fact must be known to the vehicle concerned, so that safe conditions can be brought about in its sub-systems and the leading vehicle and all other vehicles must be informed.

If the defective/separated vehicle is in the bus line, this should be classified as "defective" and indicated to the traction unit driver in the leading vehicle.

If the defective/separated vehicle(s) are at the end of the train, this should be immediately and neutrally indicated to the traction unit driver in the leading vehicle. The traction unit driver then decides whether it is a planned train shortening or a failure and enters this in the system by corresponding actions.

If the train has been shortened, the train bus system goes into the inauguration operation. If a defect has occurred, the bus operation is continued with the remaining vehicles. Attempts are periodically made to get the lost bus subscriber(s) back on the bus.

6.3 - Transmission reliability in undisturbed operation

6.3.1 - Transmission of regularly recurring information

Because of the regular repetition in the rhythm of the time based transmission priority no special safety precautions are normally required (R telegrams).

No acknowledgement of the individual pieces of information is given.

For special safety critical information antivalent data capture + data transmission + data evaluation is recommended on the user level.

6.3.2 - Transmission of event-based information

The bus shall be designed in such a way that the sender is told in each case whether the information that has been sent has arrived. It does not have to be told whether it was understood. The transmission tests shall be continued until it is reported that the E telegram has been received or until it appears to be successful, in general three times.

If communication is lost, the sending application is informed. The application decides whether there should be a further repetition of the complete report, e.g. after a suitable stabilisation period.

The processing status is coded in the 9th octet of the answer telegram corresponding to the provisions in point 5.7.3 - page 23.

7 - Modification procedure

In order to enable the data traffic to be extended in accordance with technical progress and the requirements of the RUs, the following procedure is laid down for modifications and extensions of the leaflet that become necessary.

7.1 - Modification competence

The UIC Steering Group "Train bus" is responsible for preparing modifications and additions to the leaflet.

It will always take care to guarantee the downwards compatibility of the individual leaflet versions and check to what extent an update of the leaflet is necessitated by other standardising work, especially in respect of the TCN standard. With the same objective the Control Group follows up the work of the UIC Committees and supports it by applying the relevant decisions to the UIC train bus.

Modifications to the leaflet can be initiated by

- decisions of other UIC Committees,
- applications from interested RUs or firms and
- the initiative of the UIC Steering Group "train bus".

When it is absolutely necessary to make the UIC train bus conform to the application, it is very important to make quick modifications. Therefore, authority to take decisions is delegated as follows depending on the scope of the modifications:

Modified part of the leaflet	**Modified item**	**Responsibility for taking decisions**
Appendix A - page 35	New or modified E telegrams	UIC Steering Group "Train bus"
Appendix A and B - page 36	New or modified R telegrams	UIC Working Group commissioned by the UIC Steering Group "train bus"
Point C.3 - page 84 and Appendix E - page 99	Modification in the inauguration frame	UIC Working Group commissioned by the UIC Steering Group "train bus"
Appendix D - page 91	Modifications to PDM	UIC Working Group commissioned by the UIC Steering Group "train bus"
Other parts or modification items		Technical and Research Forum (FTR)

7.2 - Modification procedure

Any modification may have serious effects on the data traffic of already existing products and vehicles. Therefore, in order to maintain an undisturbed operation and the ability to freely couple different vehicles, it is necessary to insist on two principles:

7.2.1 - Absolute downwards compatibility

Each implementation based on a later version must be able to exchange undistorted data with all earlier versions.

7.2.2 - Exact documentation of the modification version

The text and each Appendix of *UIC Leaflet 556* can be modified separately and independently from one another, therefore, each part is given a separate version number beginning with "1" and increasing from modification to modification. All modifications are accurately documented with the date of the modification advice and modified version number in the "modification certificate" see Appendix J - page 222.

Advice of modifications is issued as required - up to twice a year - by the Steering Group "Train bus" and contains the following elements:

1. New version of the appendix;
2. Modified parts of the leaflet in the new form;
3. Documentation of all changes (= comparison old-new) for the modified parts of the leaflet;
4. Date, on which the modifications come into force.

In each vehicle, it must be exactly documented by suitable lettering on the train bus node which modification condition of the leaflet is fitted, that is to say which version of the leaflet or which version of the individual leaflet parts is supported. Thus the version number of Appendix J applies as the complete version of the leaflet. For an implementation, it is sufficient to give this number.

In exceptional cases e.g. with later design modifications, implementations with different modification conditions of the individual leaflet parts are allowed. Then the supported version numbers of all parts of the leaflet must be given in matrix form.

The version of points B.1, B.2, B.3 - page 36 and C.3 - page 84 fitted and used for the application are carried in the telegram traffic.

7.2.3 - Announcement of the modifications

The leaflet and all modifications can be obtained as hard copy and as electronic file from

Direction Générale de l'UIC
Département Système
16, rue Jean Rey - 75015 Paris

or on the UIC website: http://www.uic.asso.fr

All UIC member automatically receive all updates to the leaflet.

7.3 - UIC-Steering Group "Train bus"

Members of the UIC Steering Group "Train bus" are selected amongst applicants from interested RUs and firms which develop or deliver train bus equipment.

The UIC Steering Group chooses amongst its members for two year cycles the people responsible for the following departments:

Department	**Area**
Chairman	Management, basic questions Points 1 - page 2 to 9 - page 33 and Appendices G - page 112, H - page 123, and I - page 221
Telegram traffic	Appendices A - page 35 and B - page 36
Inauguration procedure	Appendices C - page 37 and E - page 99, UIC-Mapping-Server
Standardisation principles	TCN-questions, other committees, PDM, Appendices D - page 91 and E

8 - Test procedure for train bus nodes (gateways)

8.1 - Homologation

Only type approved train bus nodes may be used.

After having completed conformity testing type approved train bus nodes are listed in Appendix I - page 221.

8.2 - Homologation procedure

The homologation of the train bus nodes for vehicles that run in international traffic is carried out by the UIC Steering Group "Train bus" which is the body with experience of the use of the UIC train bus for the transmission of information in trains. Representatives of all UIC Committees which use the UIC train bus participate in this Control Group.

The type approval of new train bus nodes is done by conformity testing which checks the conformity of the inauguration mechanism and the data traffic (R-data and E-data) of the new train bus node including the use of the UIC Mapping Server with at least one already approved train bus node.

The conformity testing shall be carried out in suitably equipped laboratories e.g. by the RUs or by the industry.

For each type approval of a new train bus node by the UIC Control Group "Train bus", an application shall initially be made to the UIC Office in Paris.

The detailed content of the conformity testing is laid down in Appendix H - page 123.

8.3 - Approval of the test laboratory

The test laboratories will be approved by the UIC Steering Group "Train bus". The approval procedure is yet to be laid down.

9 - Other matters

Addresses

The features of the bus capability (see Glossary - page 234) of vehicles that run in passenger service are characterised by addresses.

For passenger coaches that can be used on any international service in which the data transmission for the UIC train bus is done over the remote control and information cable as specified in *UIC Leaflet 558*, the marking is done while retaining the train remote control functions for remote control of the door closing, as well as the central side selective door release as specified in *UIC Leaflet 560* over wires 9, 14, 15, 16 and 12 of the cable specified in *UIC Leaflet 558*, and for remote control of the lighting as specified in *UIC Leaflet 555* over the wires 10, 11 and 12 of the cable specified in *UIC Leaflet 558* according to RIC sheet 10 as follows:

3 Sign for coaches with:

- 18 wire remote control and information cable as specified in *UIC Leaflet 558* (Cable vehicle as specified in *UIC Leaflet 556*)
- Remote control of door closing as specified in *UIC Leaflet 560*
- Remote control of the lighting as specified in *UIC Leaflet 555*.

4 Sign for coaches with:

- 18 wire remote control and information cable as specified in *UIC Leaflet 558* (Cable vehicle as specified in *UIC Leaflet 556*)
- Remote control of door closing and the side selective door release as specified in UIC Leaflet 560 over the wires 9,14,15,16 and 12 of the cable specified in *UIC Leaflet 558*
- Remote control of the lighting as specified in *UIC Leaflet 555*.

5 Sign for coaches with:

- 18 wire remote control and information cable as specified in *UIC Leaflet 558*
- Train bus nodes as specified in *UIC Leaflet 556* only capable for train inauguration
- Remote control of door closing as specified in *UIC Leaflet 560*
- Remote control of lighting as specified in *UIC Leaflet 555*.

6 Sign for coaches with:

- 18 wire remote control and information cable as specified in *UIC Leaflet 558*
- Train bus node as specified in *UIC Leaflet 556* only capable of train inauguration
- Remote control of door closing and the side selective door release as specified in *UIC Leaflet 560* over wires 9,14,15,16 and 12 of the cable specified in *UIC Leaflet 558*
- Remote control of the lighting as specified in *UIC Leaflet 555*.

7 Sign for coaches with:

- 18 wire remote control and information cable as specified in *UIC Leaflet 558*
- Train bus node as specified in *UIC Leaflet 556* fully bus compatible
- Remote control of door closing as specified in *UIC Leaflet 560*
- Remote control of the lighting as specified in *UIC Leaflet 555*.

8 Sign for coaches with:

- 18 wire remote control and information cable as specified in *UIC Leaflet 558*
- Train bus node as specified in *UIC Leaflet 556* fully bus compatible
- Remote control of door closing and the side selective door release as specified in *UIC Leaflet 560* over wires 9, 14, 15, 16 and 12 of the cable specified in *UIC Leaflet 558*
- Remote control of the lighting as specified in *UIC Leaflet 555*.

9 Sign for coaches with:

- 18 wire remote control and information cable as specified in *UIC Leaflet 558*
- Train bus node as specified in *UIC Leaflet 556* fully bus compatible
- Remote control of door closing as specified in *UIC Leaflet 560*
- Remote control of the lighting as specified in *UIC Leaflet 555*.
- Side selective door release over the train bus as specified in *UIC Leaflets 556-0 to 556-3*.

If the data transmission on the UIC train bus is done over the cable defined in *UIC Leaflet 541-5*, then the markings should be done according to the specifications of this leaflet.

Appendix A - List of information to be transmitted (Version 002.03, valid from 01.03.2009)

The list of information to be transmitted is to be found on the UIC Website (http://www.uic.org) Technology and Research/Products/UIC leaflets/Appendices/556.

Appendix B - Composition of the R telegrams

B.1 - Composition of the R1 telegram (Version 002.03, valid from 01.03.2009)

The composition of the R1 telegram is to be found on the UIC-Website (http://www.uic.org) Technology and Research/Products/UIC leaflets/Appendices/556.

B.2 - Composition of the R2 telegram (Version 002.03, valid from 01.03.2009)

The composition of the R2 telegram is to be found on the UIC-Website (http://www.uic.org) Technology and Research/Products/UIC leaflets/Appendices/556.

B.3 - Composition of the R3 telegram (Version 002.03, valid from 01.03.2009)

The composition of the R3 telegram is to be found on the UIC-Website (http://www.uic.org) Technology and Research/Products/UIC leaflets/Appendices/556.

Appendix C - Inauguration and Mapping Server

C.1 - Concept of the train inauguration (Version 002.03, valid from 01.03.2009)

C.1.1 - Foreword

The train inauguration is an important feature of the train bus as specified in the TCN standard. Beside the train inauguration itself with a view to data transmission as specified by TCN, a UIC train inauguration is necessary, which considers the actual train composition, this means the vehicles present in the train with their special properties (see points E.1 and E.2 - page 99). A condition both for the train inauguration for the purpose of data transmission (TCN inauguration) and for the UIC train inauguration is that it should be carried out in an acceptable time for the applications, which are operated over the train bus.

The concept described below considers these requirements, however, by using the application defined inauguration data of the TCN standard. The algorithm for the UIC inauguration could consequently be considerably simplified.

C.1.2 - States in the train bus operation

Fig. 1 shows the process and states in connection with the UIC inauguration.

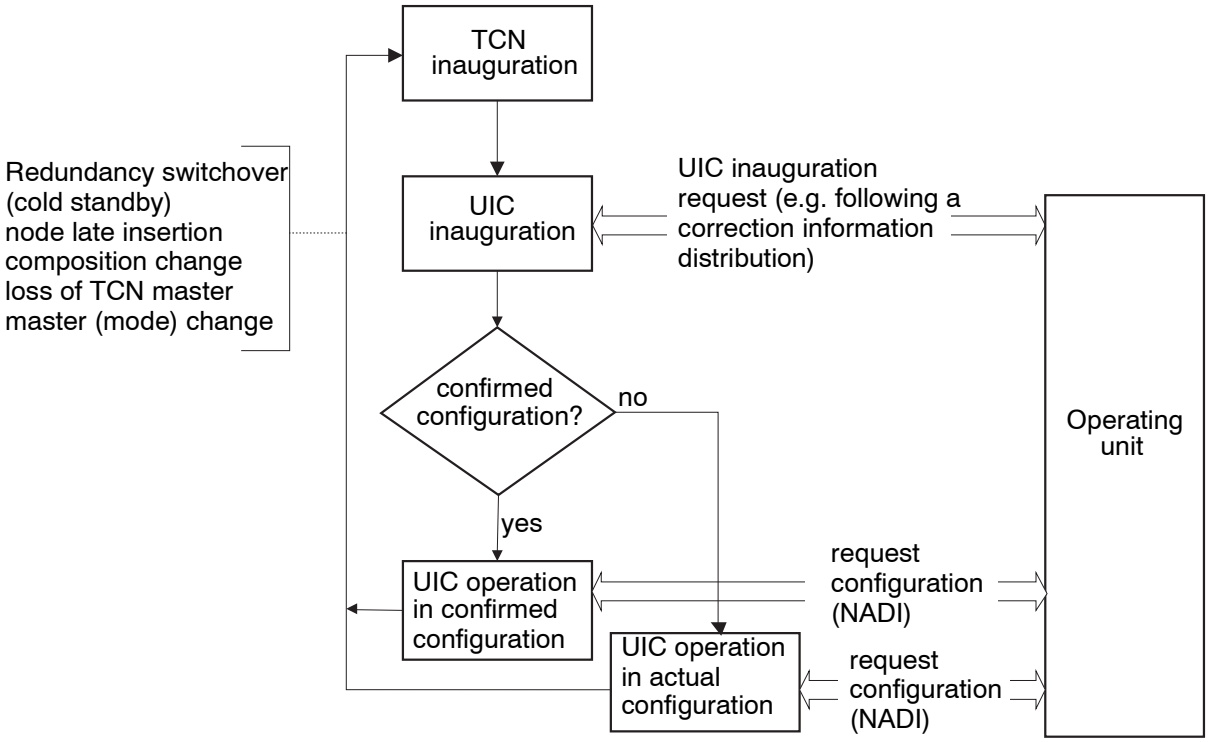


Fig. 1 - Process and states for train bus traffic in connection with the UIC inauguration

Regarding the data traffic on the train bus the following applies from the point of view of the **Application** for the individual states:

State	Process data traffic	E-telegram traffic
TCN inauguration	not possible	not possible
UIC inauguration	not possible	not possible
UIC operation with actual configuration [a]	acc. to UIC	acc. to UIC
UIC operation with confirmed configuration	acc. to UIC	acc. to UIC

a. Scope of the operation allowed with an actual configuration is laid down in the UIC leaflets based on the different applications or functionalities.

A UIC inauguration is always triggered if a TCN inauguration occurs. Moreover the application has the possibility of triggering a UIC inauguration. In this case no relays are switched in the train bus nodes (gateways), and the phase of the naming/unnaming of the TCN nodes is not carried out. The TCN Topography_Request/Response phase described below is begun directly. The UIC inauguration is always successfully completed when the distribution of the inaugural information ensured by the TCN is assessed decentrally on all nodes. If a node fails during the TCN inauguration, then it is not contained in the configuration. A failure directly after the inauguration phase is the same as a failure in normal operation.

The result is a NADI representing either an actual configuration on all nodes or a confirmed configuration on all nodes. If the mapping service cannot calculate either of the two configurations, e.g. because of the non-UIC conforming vehicles in the train, the NADI determined is not valid (see also point C.1.3.4.2 - page 45, second paragraph).

If there is a valid NADI, the UIC operation is started. The confirmation/correction of the inauguration result can be carried out by an operating device, which sends the correcting information by means of message data and implicitly triggers a request for the UIC inauguration.

The model used includes a redundant train bus connection in cold-standby. With cold-standby should be meant that respectively only one active gateway of a redundant gateway pair takes part in the train bus traffic. The passive gateway does not take part in the train bus traffic. If there is a redundancy switch over, the previously passive gateway becomes an active gateway and so triggers a TCN inauguration. It is, therefore, required that a TCN inauguration should not be prevented during the journey (inhibit bit is not set in normal operation). A train bus node designed to be redundant must recognise whether it was started as a result of a redundancy switch over.

From the point of view of the UIC inauguration, a redundancy with hot standby is only allowed if it satisfies the TCN standard and the inauguration algorithm described here. A model used for **hot standby** with a redundant train bus connection is not explicitly described.

C.1.3 - UIC inauguration

C.1.3.1 - Requirements

The UIC inauguration is based on the fact that an inauguration frame to be specified by the application is sent as TCN Topography_Response per broadcast and received at all other nodes. The TCN Topography_Request/Response mechanism thereby guarantees the following:

1. All intact nodes have identical information regarding the content of the inauguration frame exchanged at the end of this sequence.

2. A node which has not been properly through the exchange of topography does not participate in the train bus traffic without renewed inauguration.

3. TCN-inauguration and associated UIC inauguration last for 32 nodes **in the normal case 1 s**. If it is necessary to repeat the inauguration protocol because of errors, the duration of the complete inauguration is limited to a **maximum of 1,4s**. The inauguration time is defined as follows: Time from the first unnaming-request in the TCN inauguration until the completion of the calculation of the NADI.

NB : In the implementation a suitable time protocol shall be considered e.g. by means of an event recorder.

Next an assessment process runs at each node, which constructs the NADI on the basis of the information in the inauguration frame.

It should be guaranteed that all nodes construct identical information in the NADI. This is achieved by the identical algorithm running on all nodes and that for the construction of the NADI only the identical results of the TCN Topography_Request/Response phase is processed at all nodes. No node may use additionally stored data, for this can lead to inconsistent NADIs. In this connection the case that a node has not run properly through the exchange of topography, is the same as a failure shortly after the conclusion of the inauguration. If a node does not react any more immediately after the inauguration, it can only be taken into the system again by a new inauguration or by node reinsertion.

C.1.3.2 - Determination of the leading vehicle in the train

In a train there can never be more than one leading vehicle. This is a global property in relation to the train. In order to be able to deal sensibly with cases of conflict at the UIC inauguration the property 512 "vehicle is leading vehicle" is only assigned at the end of the UIC inauguration. The sending out of the R1 telegrams is associated with property 512 "vehicle is leading vehicle".

A vehicle[1] can declare the following properties at the beginning of the UIC inauguration:

- Property 510: "Vehicle was leading vehicle" (had in the last inauguration the Property 512 "Vehicle is leading vehicle")

- Property 511: "Vehicle wants to become leading vehicle"

1. For a vehicle with several train bus nodes, only that train bus node which is assigned to the traction function, can give these properties.

The following provisions apply:

1. In the determination of the leading vehicle, the only vehicles considered are those which were detected on the train bus and to which a valid UIC address can be assigned.

2. If, because of an operation, a leading vehicle no longer has the property 511 "Vehicle wants to become leading vehicle", it has at the same time lost the property 510 "Vehicle was leading vehicle"[1]

3. If at the beginning of the UIC inauguration there is a vehicle with the properties 510 "Vehicle was leading vehicle" and 511 "Vehicle wants to become leading vehicle", then this vehicle receives the property 512 "Vehicle is leading vehicle" after the UIC inauguration. This property is not assigned to any other vehicle.

4. If at the beginning of the UIC inauguration, there is no vehicle with the property 510 "Vehicle was leading vehicle" then it depends on how many vehicles report the property 511:

   a. exactly one vehicle: This and only this vehicle receives the property 512 "Vehicle is leading vehicle" after the UIC inauguration.
   b. more than one vehicle or no vehicle: no vehicle receives the property 512 "Vehicle is leading vehicle" after the UIC inauguration.

5. If at the beginning of the UIC inauguration several vehicles have the property 510 "vehicle was leading vehicle" (e.g. when trains are coupled), then no vehicle receives the property 512 "Vehicle is leading vehicle" after the UIC inauguration.

Consequently it can be guaranteed that after a UIC inauguration at the most one vehicle can be the leading vehicle in the train[2]. It is the task of the application to assess possible conflict cases regarding the leading vehicle in the train and to display the corresponding error reports.

C.1.3.3 - Vehicle sequence and recognition of direction

C.1.3.3.1 - Algorithms of the UIC address allocation

The following events cause a UIC inauguration:

1. a TCN inauguration (the necessary exchange of inauguration telegrams for the UIC inauguration takes place during the TCN inauguration);

2. an inauguration request received from a node by means of E telegrams;

3. a request to reset the configuration received from a node by means of E telegrams;

4. a request to distribute a confirmation, a correction or an assignment of vehicle numbers for seat reservation received and converted from a node by means of multi-cast E telegram to all nodes in the train;

5. a vehicle gets or loses the property 511 "Vehicle wants to become leading vehicle" or changes the direction of the request (for trainsets).

---

1. Consequently, the inconsistent combination of the two properties 510 "Vehicle is the leading vehicle" but not 511 "Vehicle wants to become leading vehicle" should be ignored by the inauguration.
2. On the basis of these arrangements it is ensured that only one train bus node sends a type 1 telegram. For protection against extreme errors it should be ensured that should two type 1 telegrams be sent, these are not assessed by any vehicle.

Independent of the event which triggers a UIC inauguration, the allocation of the UIC addresses depends on the state in which the train configuration is transferred.

The vehicle with the UIC address 01 is determined for a UIC inauguration according to the following criteria:

If the result is that no confirmed train configuration is determined, then the rules a) to d) apply, otherwise the rules a) to c).

a) Leading vehicle at one end:

   This vehicle = vehicle 01

b) Leading vehicle within the train:

   Vehicle 01 is the end vehicle, which is nearer to the leading vehicle, when counting the number of intermediate vehicles.
   If there is an identical number of vehicles d) applies.

c) No leading vehicle within the train but a traction unit at one end of the train (at the other end, therefore, none, however possibly one or more within the train):

   This vehicle = vehicle 01

d) In all other cases:

   Vehicle 01 is at one of the two train ends and thus the TCN sequence tends to correspond to the UIC sequence [a]:

Example:	61	62	63	01	02	TCN
	01	02	03	04	05	UIC

a. That is to say the "reference direction train" is identical with the "reference direction TCN"

Regarding the differentiation of the rules it should be noted that for rules a) to c) the operating staff already have, when approaching the train, a certain expectation regarding the UIC address 01, because of the position of the leading vehicle or because of the position of the traction unit. With rule d) this is not the case. It should, therefore, be indifferent to the operating staff at which end of the train address 01 is formed. For orientation in the train, however, the operating equipments should indicate to the staff the direction in which the address 01 was given.

If the result of the UIC inauguration is a confirmed train configuration, then a check is made according to rules a) to c) to see whether the sequence must be inverted. At the same time only the two vehicles which are end vehicles according to the correction or confirmation action are considered as end vehicles. Their UIC address is, therefore, either 01 or identical with the number of the confirmed vehicles in the train. Furthermore vehicles with unknown UIC address 00[1] are not taken into consideration for the decision whether the order should be inverted (see point C.1.3.5.5 - page 52).

1. That is to say an undefined position in the train

C.1.3.3.2 - Direction determination forwards, backwards, left and right

A "reference direction train" is laid down for the complete train. This is calculated for each train bus node from the UIC order established with the UIC inauguration and stored in the NADI. No further data traffic beyond the UIC inauguration frame is necessary for this.

1) TCN inauguration

At the end of the TCN inauguration each train bus node sends its inauguration frame (see point C.3 - page 84) to all train bus nodes at the request of the train bus master. In this there is the orientation of the train bus node relative to the orientation of the train bus master. The latter defines the "reference direction TCN".

Serial No. from point E.2 - page 99	Purpose	Information route				Type of tele-gram	Information origin	Representation and processing				
		Source		Target				Meaning	Data type/ extent of the values	Octet	Bit	Code/ value
		Fct.	Veh.	Veh.	Fct.							
1a	2	13	3	9a	14	10	15	16	16a	17	18	19
505	Report: Orientation relative to "reference direction TCN"		all veh.	all veh.		Inaugu-ration frame	Train bus node	"Reference direction train bus node" is to "reference direction TCN" the same	Bitset8	3	7	1
								the inverse				0

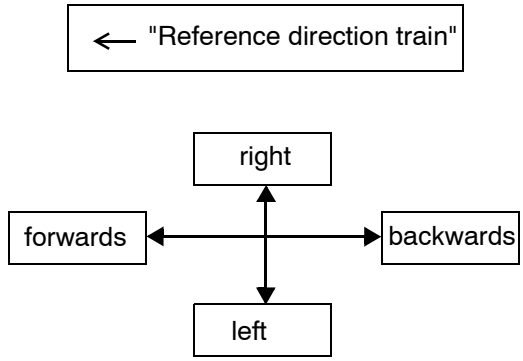
As a result of the TCN inauguration each train bus node recognises the reference direction of all other nodes relative to the "reference direction TCN".

2) UIC inauguration

Since the "reference direction train bus nodes" according to the installation specifications is identical with the reference direction of the trainset/vehicle (see Fig. 5 - page 58) each train bus node can calculate the UIC order and consequently the "reference direction train" according to the rules from point C.1.3.3.1 - page 40 using the following algorithm:

Vehicle X ≠ Vehicle 01 ⇒ "Reference direction train" = towards vehicle 01

Vehicle X = Vehicle 01 ⇒ "Reference direction train" = away from vehicle 02

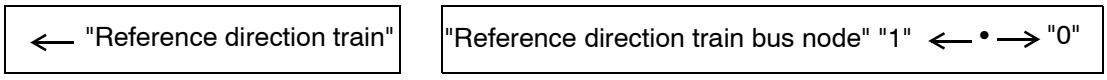


3) Assignment of the particular reference direction

Each vehicle determines its orientation relative to the "reference direction train" by the following rules:

1. If the "reference direction TCN" is identical with the "reference direction train" then the orientation to the "reference direction train" is **the same** as the orientation of the train bus node relative to the "reference direction TCN" (= property 505).

2. If the "reference direction TCN" is opposite to the "reference direction train" then the orientation to the "reference direction train" is **inverse** to the orientation of the train bus node relative to the "reference direction TCN" (= property 505).



The train bus nodes convert correspondingly all incoming and outgoing commands and reports, which contain the terms in Appendix A - page 35 "forwards", "front", "backwards", "back", "left" or "right" uniformly to the "reference direction train".

The further conversion of the "reference direction train bus nodes" into the "reference direction vehicle" depends on whether the train bus node controls an individual vehicle or a trainset.

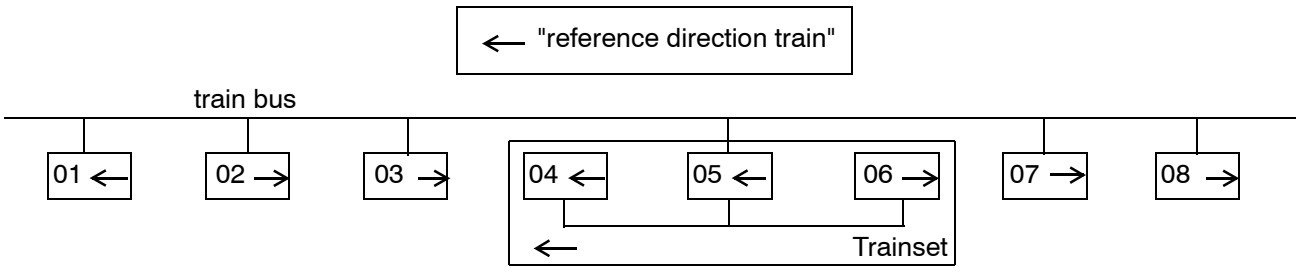
a) Train bus node controls an individual vehicle:
"Reference direction train bus node" = "Reference direction vehicle"

b) Train bus controls a trainset:
"Reference direction train bus node" = "Reference direction trainset"

"Reference direction train bus nodes", "reference direction trainset" and "reference direction vehicle" are laid down as specified in the regulations for the installation of the train bus nodes and, therefore, programmed in a fixed way. At the same time in a trainset the "reference direction trainset" and the reference directions of the individual vehicles do not necessarily have to correspond.

The further conversion of the "reference direction trainset" into the "reference direction vehicle" is no longer an item of the train bus functions, it is carried out by other applications, e.g. by an MVB according to TCN.

Example:



If there are several train bus nodes in one vehicle, then each of these nodes carries out the above mentioned calculations independently. In order to obtain the same results, it is important that the "reference direction train bus node" is given in each node of the vehicle in the same way.

C.1.3.4 - Data structure

C.1.3.4.1 - Principle of the construction of the inauguration frame

The detailed construction of the inauguration telegram is described in point C.3 - page 84.

Octet/bit No	*Name*	*Size (in bytes)*
	Information for train bus nodes (= trainsets):	
1	UIC identification: 0x1	1
2	UIC inauguration version number	1
3	Orientation and TCN address	1
4	Length of following net data in octets	1
5	UIC code of the operating RU	1
6	UIC code of the owning RU	1
7	National application identification	1
8	National telegram version	1
9	NN > 0: Number of vehicles controlled from this node (1…6) NN ≤ -2: Number of nodes in this vehicle	1
10	Status/control bits	1
11 [a]	UIC address of the vehicle in which the train bus node is located (property 141)	1
12	Confirmed number of vehicles	1
13...20	Confirmed position of the vehicles which are not reachable over the train bus	8
21	R telegram version number	1
22...35/5	Gateway (= trainset) specific properties	14/6
35/6...37	Reserve for gateway specific properties	2/2
38...41	CRC32 for safe data transmission of the inauguration frame	4
42...43	Reserve for national gateway specific properties	2
44	General reserve	1
	Information for max. 6 controlled vehicles: (i = 0...5)	
45+(i*14)... 49+(i*14)	UIC identification number (ID) [b]	5
50+(i*14)... 54+(i*14)	Vehicle specific properties	6
55+(i*14)...	Reserve for national vehicle specific properties	1
56+(i*14)...	General reserve	1
57+(i*14)... 58+(i*14)	Coach number for seat reservation [c]	2

a. The octets 11…20 in the inauguration frame of an unknown node are of no significance.
b. For the converting of the 11 significant digits of the UIC vehicle identification number (octet 45 + (i*14)…49 + (i*14)) into binary format the UIC vehicle identification number should be interpreted as an 11 character decimal figure (example: 008071-40492 gives when converted 00 30 1b fc 8c). The transmission of the binary format takes place first with the "most significant octet" (in the example "00").
c. The value 0 characterises that no coach number was assigned.

The first 4 bytes stand for particular net data in the ID traffic store; for a maximum of 6 controlled vehicles per trainset there is consequently a total length of the inauguration frame of:
44 + 14*6 = 128 bytes.

C.1.3.4.2 - Use of application specific TCN information

The first two bytes of the inauguration frame contain from the TCN view the node type and the node version. It is established that the node type is covered with the UIC identification and the node version with the UIC inauguration frame.

An inauguration is only then successful, if all vehicles report the UIC identification as node type. Otherwise no valid NADI can be computed (see point C.1.2 - page 37).

Regarding the TCN User Report it is determined, that from the point of view of the UIC application it is undefined and must not be changed.

C.1.3.4.3 - Local structures

The inauguration algorithm requires, beside the data in the inauguration frame, the following local data structures to control the various states during the inauguration:

- node status,
- confirmed UIC address,
- confirmed number of vehicles,
- confirmed position of the vehicles that cannot be reached over the train bus (gaps).

Moreover as a result of the TCN Topography_Request/Response phase, a table with the data of all nodes is available. It is designated as a network configuration and contains for each node the TCN address and a copy of the inauguration frames.

C.1.3.4.4 - Composition of the NADI

The NADI consists of a global part as well as a description for individual vehicles. In the global part there are the following elements:

1. UIC inauguration frame version number,
2. R data version number,
3. NADI status,
4. Topo counter,
5. Number of entries,
6. Position of the vehicles that cannot be reached,
7. Add on information.

The add on information is of the type BITSET8 and has the following meaning:

Bit 0: orientation of the UIC reference direction (direction of view of the vehicle with UIC address 01) relative to the TCN master, where: 0 = inverse, 1 = the same.

Bit 1: is set at 1, when at least one node was additionally active in a confirmed configuration.

Bit 2: is set at 1, when at least one node has failed in a confirmed configuration.

Bit 3-7: reserve.

The bits 1 and 2 are set to 0 at each changeover from a confirmed configuration to an actual configuration and vice versa.

Each line in the NADI contains the following elements:

1. TCN address,
2. Number of vehicles controlled,
3. UIC address,
4. Operating RU,
5. Owning RU,
6. National version,
7. Application identification,
8. Trainset properties,
9. UIC-ID,
10. Vehicle number for seat reservation,
11. Vehicle specific properties,
12. Vehicle add on information.

The entry vehicle add on information is of type BITSET8 and has the following significance:

Bit 0: orientation of the vehicle relative to the TCN master, where:
0 = inverse, 1 = same.

Bit 1: orientation of the vehicle relative to the UIC reference direction, where:
0 = inverse, 1 = same

Bit 2: property 512 "Vehicle is leading vehicle" where:
1 = leading, 0 = not leading

Bit 3: property 511 "Vehicle wants to be leading vehicle" was set in theo inauguration frame

Bit 4-7: reserve.

In the NADI various vehicles are represented as follows:

A Vehicles with one gateway
One line.

B Trainsets
One line for each controlled vehicle (own UIC address and own UIC ID); the entries in the following columns are identical for the controlled vehicles of a trainset:
- TCN address,
- owning RU,
- trainset specific properties,
- operating RU

C Vehicles with several gateways
A line for each gateway, the entries in the following columns are identical for vehicles with several gateways:
- owning RU,
- UIC address,
- UIC ID
- operating RU

D Vehicles which do not take part in the train bus traffic
A line just for the UIC address (such vehicles were inserted because of a correction).

The number of lines in the table is not restricted by the maximum number of vehicles, since several lines are necessary for vehicles with several gateways. The NADI contains an identification, which lays down whether an actual configuration was determined or whether an already confirmed configuration was found. The vehicles are shown in the NADI in rising order in the reference direction train.

C.1.3.5 - Description of the inauguration algorithm

C.1.3.5.1 - General

A train bus node recognises three states locally at the beginning of a UIC inauguration:

1. Configuration unknown;

2. UIC address set;

3. Configuration stored.

The state 2 is adopted only immediately after a correction/confirmation of the inauguration results. It means that the train bus node has already received its correction information.

Because of its state a train bus node announces various informations during the inauguration. In the state "configuration unknown" the values of the correction information are undefined. In the other states these values are valid.

The inauguration frames of all vehicles are distributed and entered in the network configuration according to the TCN address .

The inauguration algorithm is then carried out distributedly on each train bus node.

The UIC identification (Node_Type) is checked next, regardless of the state of a train bus node. If a vehicle has no UIC identification the inauguration is not successful.

The R telegram version number is then checked. Regarding the version numbering, it is assumed that only upward compatible extensions of a version of the leaflet are initially allowed. If different R telegram versions are reported, the minimum of all reported version numbers is automatically computed and used as the actual R telegram version number for the data interpretation (see octet 2 of the R telegram).

The further course of the algorithm depends on the state of **all** train bus nodes involved in the inauguration. The following state diagram gives an overview of the state change within a train bus node:

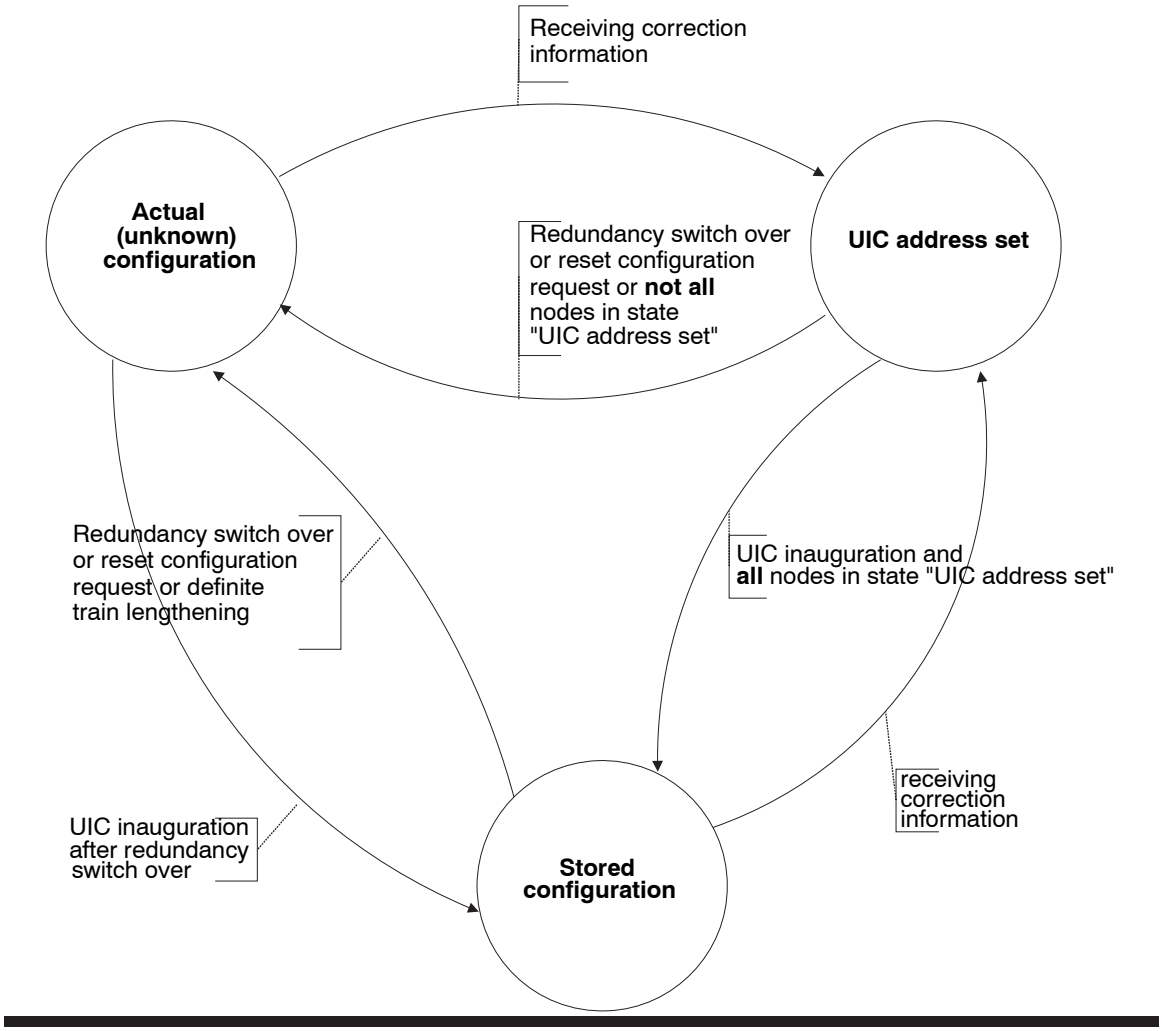


Fig. 2 - States of a train bus node and its state transitions

C.1.3.5.2 - Determining the type of train configuration

In general it must be possible to distinguish at the start of the UIC inauguration whether the result will be a confirmed train configuration or not. This is decided on the basis of the reset configuration request and the states of all the train bus nodes involved in the UIC inauguration.

The allocation of the UIC addresses is described in point C.1.3.3.1 - page 40. For details regarding the construction of the NADI see Appendix A - page 35, check telegram, telegram 0.01A.

If the control bit "reset confirmed configuration" is set by a train bus node then the NADI is calculated as the actual configuration. The train bus node changes to the state "configuration unknown".

If all train bus nodes report the control bit "configuration unknown", then the NADI is computed as the actual configuration.

If some but not all train bus nodes report the control bit "UIC address set", then the NADI is computed as the actual configuration. The train bus node changes into the state "configuration unknown".

If **all** train bus nodes report the control bit "UIC address set", a distinction must be made as to whether it is a correction or a confirmation. In both cases the train bus node changes into the state "configuration stored".

If all train bus nodes report the UIC address "00" in the inauguration frame, it is a confirmation. The computed configuration is then marked as a confirmed configuration. The calculated number of the vehicle and its own UIC address are stored in the inauguration frame.

Otherwise it is a correction, in which after the consistency check of each train bus node the correction information is accepted locally.

If the train bus nodes is in the state "configuration stored" and possibly one or more is in "configuration unknown", then after the consistency check the NADI is marked as "confirmed configuration".

If in the consistency check an error is found in the confirmed train configuration reported in the inauguration frame, then the NADI is computed as the actual configuration. The train bus node goes into the state "configuration unknown". This is the case if:

- *a definite train lengthening* (vehicles have been added) has taken place or
- the "reference direction train" can no longer be calculated or[1]
- the sequence of the UIC addresses reported in the inauguration frame is not monotonic.

A *definite train lengthening* is defined as follows:

In a UIC inauguration the number of the vehicles reported in the UIC inauguration is compared with the number of the confirmed vehicles.

1. The "reference direction train" can accurately be calculated if
   - at least two different stored UIC addresses are reported or
   - at least one train bus node is in the condition "configuration stored" or "UIC address set" and this node is a TCN end node, whose UIC address (octet 11 in the inauguration frame) is 01 or equal to the confirmed number of vehicles.

- If at the front of the train before the first vehicle or/and at the end of the train behind the last vehicle with "confirmed configuration" more new vehicles are reported, as the number of existing gaps, then the vehicles added cannot be arranged within the configuration. Then a definite train lengthening has taken place (coupling a confirmed train with individual vehicles or an unconfirmed train).

- Vehicles, which between the first vehicle with "confirmed configuration" and the last vehicle with "confirmed configuration" are recognised by a new UIC inauguration, do not lead to a "definite train lengthening" that is to say the train sequence remains confirmed.

- If from the vehicles with "confirmed configuration" different numbers of vehicles and/or different numbers or/and positions of confirmed gaps are reported, then a definite train lengthening has taken place (coupling of two confirmed trains).

- If the number of the vehicles reported in the UIC inauguration with "confirmed configuration" is greater than the reported number of vehicles, then a definite train lengthening has occurred (coupling of two confirmed trains).

C.1.3.5.3 - Integration of nodes in the state "configuration unknown" in a confirmed train configuration

If at least one unconfirmed node in the middle of the train (which was not an end node) was active during a UIC inauguration and confirmed nodes existed in the train, then by means of the following formula it is decided whether the unconfirmed vehicles can be included in the confirmed configuration:

$$N_{vehicle\_gaps}(V_1,V_2) = N_{unconfirmed\_vehicles}(V_1,V_2) + N_{confirmed\_vehicle\_gaps}(V_1,V_2)$$

in which:

- V_1,V_2 are the active vehicles with confirmed UIC address in front of and behind the vehicles to be classified,

- $N_{vehicle\_gaps}(V_1,V_2)$ are the number of vehicles between V_1 and V_2,

- $N_{confirmed\_vehicle\_gaps}(V_1,V_2)$ are the number of confirmed gaps between V_1 and V_2,

- $N_{unconfirmed\_vehicles}(V_1,V_2)$ are the number of active vehicles without confirmed UIC address between V_1 and V_2.

Example:

1. In a train with four vehicles a gap is corrected between the third and fourth vehicle, so that the train now has five confirmed addresses. If vehicle 03 now fails triggering an inauguration and becomes active again, then the following configuration is produced:

UIC address	01	02	unknownt	04	05
TCN address	0x01	0x02	0x03	confirmed gap (0x7F)	0x04

with the above formula we get 2 (02, 05) = 1 (02, 05) + 1 (02, 05), that is to say after the UIC inauguration the confirmation appears as follows:

UIC address	01	02	03	04	05
TCN address	0x01	0x02	0x03	confirmed gap (0x7F)	0x04

2. In a train with three vehicles, two gaps were corrected between the second and third vehicle so that the train now has five confirmed addresses. If a node becomes active with two vehicles at this location, then the following configuration is achieved:

UIC address	01	02	unknown	unknown	03	04	05
TCN address	0x01	0x02	0x03	0x03	confirmed gap (0x7F)	confirmed gap (0x7F)	0x04

with the above formula 2 (02, 05) ≠ 2 (02, 05) + 2 (02, 05), that is to say after the UIC inauguration the configuration will appear as follows:

UIC address	01	02	00	00	03	04	05
TCN address	0x01	0x02	0x03	0x03	confirmed gap (0x7F)	confirmed gap (0x7F)	0x04

3. A train with four vehicles is confirmed. If now vehicles 02 and 03 fail triggering an inauguration and if one of them again becomes active, then the following configuration is obtained:

UIC- address	01	unknown	02	03	04
TCN address	0x01	0x02	unconfirmed gap (0x7F)	unconfirmed gap (0x7F)	0x03

with the above formula 2 (01, 04) ≠ 1 (01, 04) + 0 (01, 04), that is to say after the UIC inauguration the configuration will appear as follows:

UIC address	01	00	02	03	04
TCN address	0x01	0x02	unconfirmed gap (0x7F)	unconfirmed gap (0x7F)	0x03

C.1.3.5.4 - Integration of nodes after redundancy switch over

A node can also be set into the state "configuration unknown" as a result of a redundancy switch over. It reports this in its inauguration frame by setting the redundancy bit.

The procedure regarding the redundancy bits is as follows:

1. When initialising the node, the redundancy bit is preset to 0.

2. The redundancy bit is set to 1 if the "redundancy switch over recognition" recognises a redundancy case.

3. The redundancy bit is then assessed by the inauguration algorithm.

4. The redundancy bit is reset to 0 after each UIC inauguration (for single nodes by a time out).

In addition to the integration conditions for nodes in the state "configuration unknown", a redundant node, which does not lie in the middle of the train between two nodes in the "configuration stored" state, is integrated then and only then in a confirmed train configuration, if

- the redundant bit is set to 1,

- the node is in the state "configuration unknown",

- the redundant node can be assigned in analogy to the above formula, that is to say a valid UIC address can be assigned to it.

C.1.3.5.5 - Display of unknown vehicles

A vehicle to which a valid UIC address cannot be assigned is written into the NADI with the UIC address 00 and its TCN address. They are arranged in the NADI bearing in mind the TCN order according to "reference direction train" as far as possible to the front. In the global part of the NADI, a bit is set which indicates that a vehicle was recognised, to which no UIC address could be assigned. If it is desired that the node again subscribes to the train bus traffic, then a correction must be carried out.

For a vehicle that was present in the confirmed configuration and is no longer active in the inauguration, the TCN address is set in the NADI to 0x7F, that is to say the node is marked as no longer accessible and its UIC address remains valid. Bit 2 is set in the add-on-information of the NADI.

In each case after the inauguration with the respective configuration, the UIC operation is started.

C.1.3.6 - Construction of the NADI from the network configuration

First the correct number of vehicles is determined (expanding of the trainsets, multi-mentions for vehicles with several gateways). This is important so that with the allocation of UIC addresses the rules regarding the position of the leading vehicle are correctly observed. Then the UIC addresses are assigned or accepted.

In the treatment of trainsets the following should be noted:

A trainset always gives the information regarding the checked vehicles in coach direction 1 as specified in the TCN standard (see Fig 5 - page 9). In the carrying out of the inauguration algorithm the orientation regarding the TCN masters is then known.

Because of the information regarding UIC address 1, the TCN addresses of all train bus nodes and the orientation relative to the bus master, the algorithm must now determine the correct arrangement of UIC addresses and UIC IDs (and consequently vehicle numbers for seat reservation) within trainsets and write them into the NADI. Moreover for vehicles with several gateways, all necessary entries are set up in the NADI.

C.1.3.7 - Confirmation/correction of an inauguration result

It is assumed that the correction activity is not time critical.

A correction/confirmation can be carried out by different authorities, e.g.:

- dialogue with the traction unit driver over a control device,

- automatic correction by the vehicle control under inclusion of additional information.

If the inauguration result is marked as "not confirmed", then a confirmation/correction can be carried out by the application. In this way new vehicles can be put into the train in any position. These can be cable vehicles or vehicles with defect gateways. The corrected configuration (the confirmation or correction telegram) for this is sent by multicast over the gateway of the TCN master[1].

1. This protects the data consistency with parallel correction action.

On receipt of a confirmation telegram, each node in the inauguration frame sets the control bit "UIC address set" and initialises in the inauguration frame its UIC address. After distribution of the telegram to all train bus nodes, the gateway of the TCN master initiates a UIC inauguration. The actual train configuration determined by the UIC inauguration is accepted by each gateway as the confirmed configuration.

On receipt of a correction telegram, each gateway calculates its own UIC address taking into consideration the vehicles inserted. Gaps at UIC address n are respectively coded by a 1 in bit position n-1. If all train bus nodes have received this correction E-telegram, the gateway of the TCN master initiates a UIC inauguration.

The confirming authority can check after the completion of this inauguration whether the confirmation or correction was carried out successfully. If the inauguration result is marked as "confirmed configuration", then the procedure was completed successfully. If the inauguration result is an actual configuration, the confirmation/correction could not be carried out and the procedure can be repeated.

C.1.3.8 - Versioning

It is not proposed that there should be a versioning of the inauguration result. Since the TCN topocounter is always changed when a UIC inauguration is triggered, the message data traffic to the interfaces to the train bus is correspondingly monitored and rejected if there are discrepancies in the topocounter.

C.1.3.9 - Interfaces to the application

Each application must be able to carry out the following actions:

- inform its own vehicle properties to the Mapping-Server,
- trigger a UIC inauguration,
- place a reset configuration request,
- send correction information to TCN master.

After the inauguration the application receives access to the following data within the node:

- NADI (e.g. property 512 "vehicle is leading vehicle" configuration result),
- Mapping Server status,
- train topography.

This interface is described in Appendix A - page 35 telegram groups 0 and 15.

After completion of the UIC inauguration the application gets access to the NADI. The access functions are described in point C.2 - page 55.

C.1.3.10 - Restrictions

Regarding the number of train bus nodes there are no restrictions on the basis of the inauguration algorithm, which are not stipulated by the TCN standard anyway.

On the basis of the design of the inauguration frame a gateway in a trainset can process information for a maximum of 6 controlled vehicles. The maximum number of vehicles in the train is 22.

Record of updates

Version	Date	Change	Reason for the change
002.01	01.11.2004	Change of the layout according to M1	New layout based on UIC guide M1
		Adoption of a history of updates	Increasing the usability
		Adoption of a revision number	Enhancement and redesign of the versioning
002.02	01.08.2005	Change in the layout	UIC requirement
		Renumbering of the Appendices	
002.03	01.03.2009	Error corrections and one new gateway specific vehicle property taken into account (see table in point C.1.3.4.1 - page 44).	New vehicle property corresponding to safe data transmission (Appendix L - page 224).

C.2 - UIC Mapping Server - Specification (Version 002.02, valid from 01.08.2005)

C.2.1 - Summary

The TCN gateway as it has been defined in the TCN standard has mainly to perform two tasks: first, to "marshall" process data from the MVB to the WTB and vice versa (Process Data Marshalling), and second, to route message data from and towards the WTB. All this is performed by the TCN protocol stack, extended by Process Data Marshalling module.

As these are basic network functions which allow full communication ability all over the network, the addressing capabilities are rudimentary (just speaking about WTB nodes and MVB functions) and the capabilities to have more information about the environment the TCN is embedded in are not subjected. Furthermore, the pure TCN gateway just provides information about the TCN network, but doesn´t consider parts of the environment that are not reachable via TCN. This gap is filled by *UIC Leaflet 556*, which defines a user profile of the TCN especially suited for UIC applications. This user profile defines all signals that are to be transmitted over the WTB (both process data signals and messages) and it defines an enhanced addressing scheme comprising addressing of physical vehicles and vehicle groups. It also provides additional information about the system environment, thus describing the whole train composition rather than only the communication network.

To cope with these additional, UIC specific, requirements a specialized software component, the **"UIC Mapping Server"** (see Bibliography - page 236), provides the user with all the information that is needed to use the enhanced addressing schemes as defined in *UIC Leaflet 556*. For this purpose, all necessary information like vehicle descriptions and train composition, is collected during the inauguration phase and stored in a database ("node address and attribute directory", NADI). Each WTB node holds an identical copy of that database. Information, which is not available at the time of inauguration like for instance group memberships, can be added at any time to that database (this can be done for instance by using a MMI display).

The UIC Mapping Server is a supplement of the TCN protocol stack for UIC applications. A TCN gateway enhanced by the capabilities of a UIC Mapping Server will be called **a UIC gateway**.

The present point describes the basic functions of the UIC Mapping Server in a more general way (like a tutorial) and describes the individual components the UIC Mapping Server consists of. In addition, it defines basic design principles of the UIC Mapping Server.

C.2.2 - Introduction

The relationship between the TCN standard on the one hand and the application of the TCN by UIC on the other hand is depicted in Fig. 3. *UIC Leaflet 556* describes all communication related issues based on the TCN services for process data and message data transfer. It is supplemented by a number of additional leaflets that detail the user profile of specific functions.

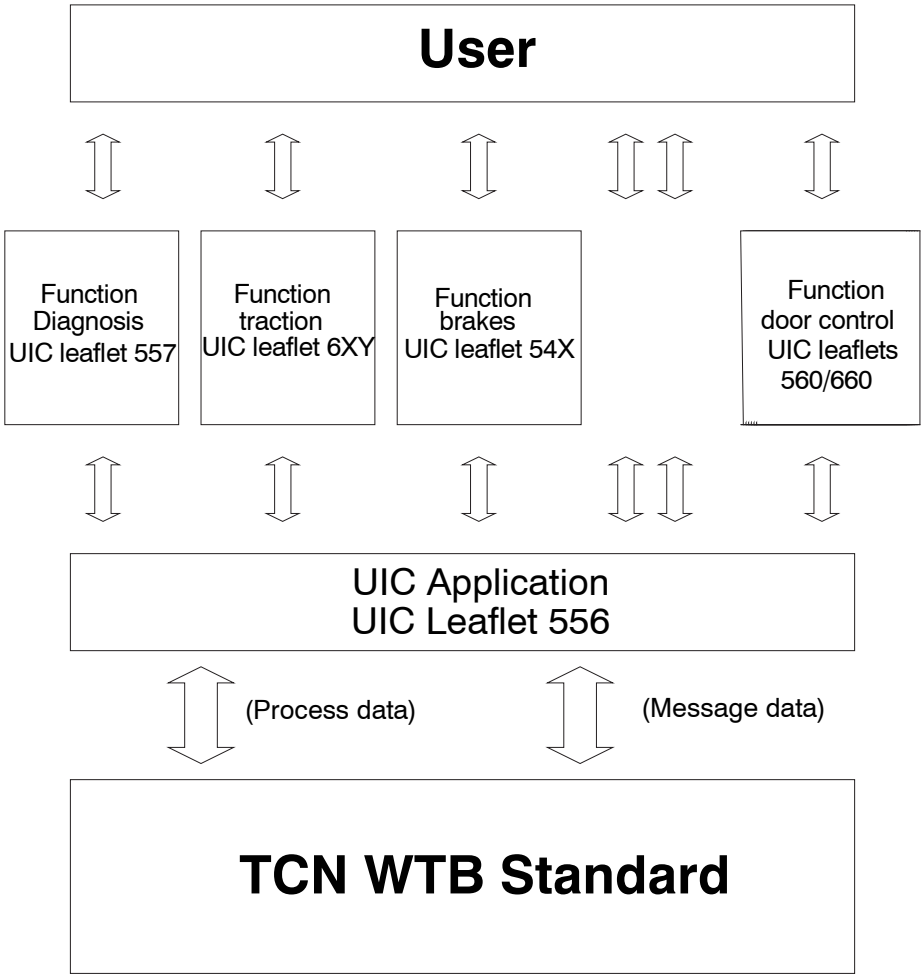


Fig. 3 - Relation between the TCN standard and its application by the UIC

As this UIC mapping server software will be an integral part of a UIC comformable TCN gateway ("UIC gateway") the development must take into consideration the whole gateway software structure and its interrelationships when specifying the UIC Mapping Server software.

The UIC Gateway is generally used for the connection of vehicle control systems, either distributed or not, to the WTB train bus. The gateway performs the task of data routing between the train bus and the subordinated vehicle bus. The UIC gateway can be seen as an ordinary TCN gateway customized for the special needs of UIC conformable applications. The compliancy with UIC requirements, that are defined in *UIC Leaflet 556*, is one of the main prerequisites the UIC gateway is faced with.

The terms "message" or "message data" and "E telegram" are used as synonyms in all of point C.2. "Message" is the standard expression used in the TCN Standard and is therefore used in texts relating more to the TCN/UIC relationship. "E telegram" is the corresponding term used in *UIC Leaflet 556*.

C.2.3 - UIC Mapping - Introduction

Figures 4 and 5 - page 58 illustrate the diversity an application programmer of typical train subsystem software components like door function, traction control etc. has to deal with.

Figure 4 shows the TCN view of a train, which shows only TCN network with its WTB nodes and MVB stations. The relation to the real train consisting of vehicles of different kinds is not obvious, thus the application programmer is "blind" when he is confronted with tasks that are vehicle related. Even more, the WTB addressing scheme is dynamic, it may change from inauguration to inauguration. Hence, addressing of vehicles may become a mess. If for instance a function in a remote vehicle shall be activated by messages, one has to ensure that between the time when the message was sent and the time when it was received no inauguration was made which altered the node addresses. Of course, there are mechanisms in the TCN protocol stack that prohibit malfunctions, like for instance the checking of the topography counter. But all this is not very practical.

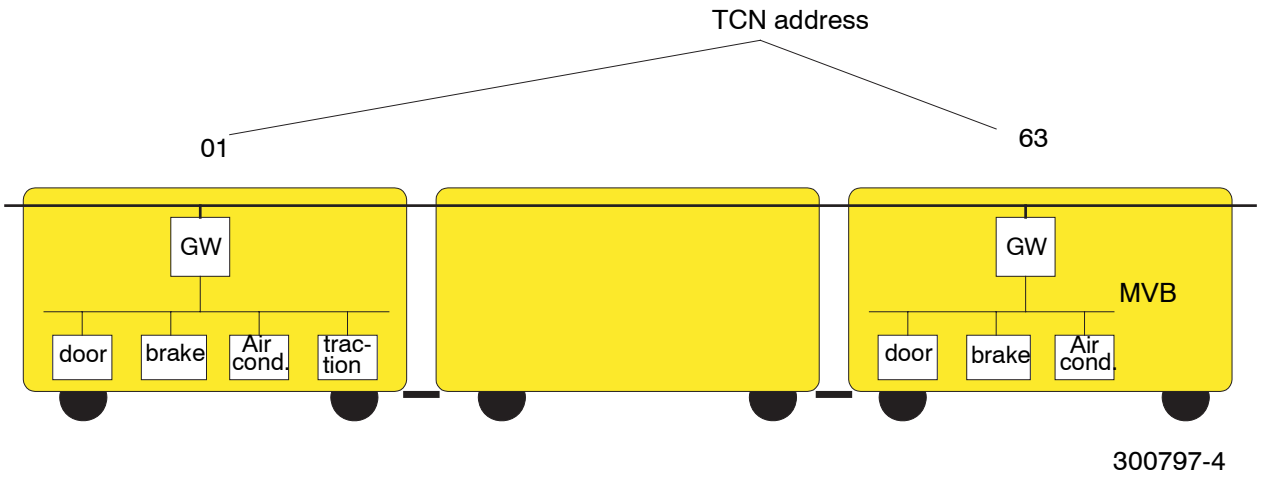


Fig. 4 - TCN view of a train

Since a "TCN view" is not sufficient, UIC defined in its leaflet 556 a "user view" of a train communication network. In this user view, only vehicles are considered and functions within that vehicles. The internal structure of a vehicle is invisible and, even more, irrelevant. For vehicle addressing four possibilities are defined:

1. range number ("UIC address"),

2. vehicle number for seat reservation,

3. vehicle identification number (unique number for UIC vehicles),

4. group (predefined and user defined).

Functions within the vehicles are simply identified through their function number.

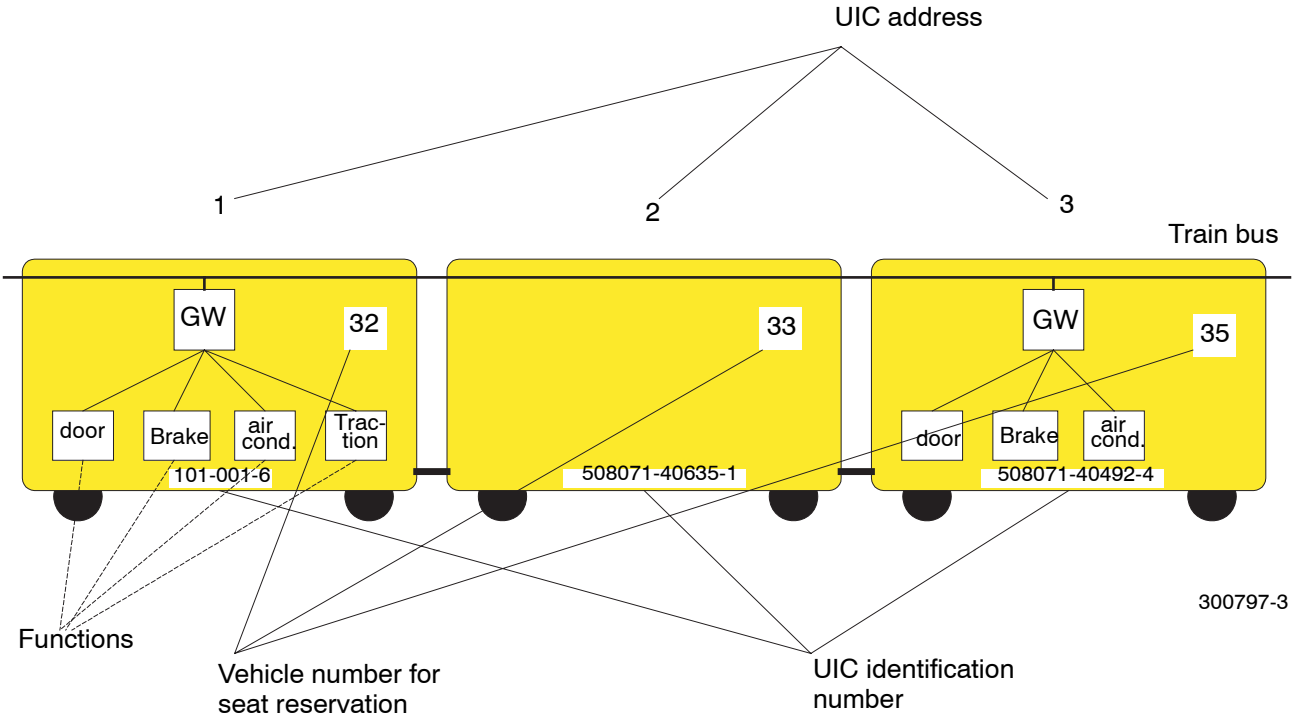


Fig. 5 - UIC view of a train

Concerning the ISO OSI 7-layer model, which defines the 7 layers of communication, the UIC Mapping Server must be part of layer 7 (application layer), although it is somehow between the TCN application interface (which is also defined in layer 7) and the application itself.

A TCN gateway enhanced by these capabilities of a UIC Mapping Server will be called a **UIC gateway**.

The services offered by the UIC gateway can be devided into two categories:

1. system services, which are autonomously performed and are mainly responsible for establishing a running network;

2. user services, which allow a remote or local user to control the communication.

These services are described in more details in point C.2.4 - page 59. The related user interface is defined in point C.2.5 - page 65.

The basic software structure of the UIC Mapping Server is described in point C.2.6 - page 67. Because the UIC Mapping Server cannot be considered independently from other parts of the UIC gateway, point C.2.7 - page 79 contains a brief description of those parts.

C.2.4 - UIC Mapping services

This point describes the services that are offered by a UIC gateway in general and the UIC Mapping Server in particular.

C.2.4.1 - Definitions

Two kinds of services exist within the UIC gateway. Some of these services are performed autonomously by the gateway once the gateway is started up. They have to be preconfigured on the gateway and are called "**gateway system services**".

Example

Once on, the gateway automatically starts as a weak master, performs TCN and UIC inauguration and marshals process data.

The gateway also acts as a service provider and offers "**gateway user services**" which can be used by remote users. These services are typically used to retrieve state information, topography information or to submit a multicast.

C.2.4.2 - Gateway system services

The following table lists all the services the UIC gateway has to perform autonomously. To do this, the gateway needs no interaction with the user, it must only be preconfigured accordingly.

Table 1 : System services

Object	*Service*	*Description*
TCN	tcn_inauguration	TCN inauguration procedure as defined in the TCN standard, part 4.
	process_data	Process data transmission over MVB and WTB as defined in the TCN standard, parts 2 and 4.
	message_data	Message data transmission including routing of message data between WTB, vehicle bus (e.g. MVB) and gateway local as defined in TCN standard, part 2.
	network_management	Agent for network management services as defined in the TCN standard, part 5.
PDM	process_data_marshalling	Marshalling of process data between WTB and vehicle bus (e.g. MVB) as defined in *UIC Leaflet 556*.
UMS	wtb_ configuration_and _control	Preconfiguring and starting the WTB link layer according to *UIC Leaflet 556* (weak master).
	uic_inauguration	Sending and receiving inauguration data, calculating the valid NADI by applying the algorithm defined in point C.1 - page 37 and storing the NADI in the NADI database.
	correction_train_inauguration	Reception of correction info and distribution to all other WTB nodes.
	nadi_control	Building up and maintaining the NADI.
	group_definition	Reception of new group list entries and storing it in the NADI database. Handling of groups.
	redundancy_handling	Handling of a redundant gateway.

C.2.4.3 - Gateway user services

In addition to the gateway system services the UIC gateway provides the following user services:

Table 2 : User services

Object	*Service*	*Access*	*Description*
UMS	change_omode	local	Change operational mode of the WTB node either to PASSIVE, SLAVE, WEAK or STRONG.
	inauguration_control	global	Allow or inhibit inauguration.
	sleep	global	Request or cancel the request to enter sleep mode.
	inauguration_enforce	global	Enforce a new UIC inauguration (WTB topography exchange).
	read_uwtm_state	global	Read status information from the WTB node.
	read_uwtm_topo	local	Read topography information from the WTB node.
	write_res_nr	global	Set the vehicle numbers for seat reservation.
	write_correction	global	Write correction information and update the NADI accordingly.
	leading	local	Request or reset the request of leadership.
	read_nadi	global	Read the actual valid NADI version.
	write_group	global	Store a group in the NADI.
	write_group_all	global	Store a new group list.
	read_group	global	Read the group members of a group.
	read_group_list	global	Read a list of all defined groups.
	delete_group	global	Delete a group.
	delete_group_all	global	Delete all groups.
	mutlicast_create	global	Request a multicast transmission.

global: access via WTB or MVB

local: access via MVB

C.2.4.4 - User service usage

User services are called up by message data, that means a service is requested by sending a call_message to the **UMS_Function** and the service result is delivered in the corresponding reply_message.

Hence the gateway operates as a server reacting only on requests from a remote user (Fig. 6)[1].

Example:

A MVB device wants to know about the newest version of the NADI. It sends a "read_nadi" service call message to the gateway and gets the new NADI in the reply.

It is invisible from the outside of the gateway which software instance of the gateway UMS finally does the job. The remote user sees only the "UMS_Function" he has to address.

By convention, the UMS_Function has the function number 15.

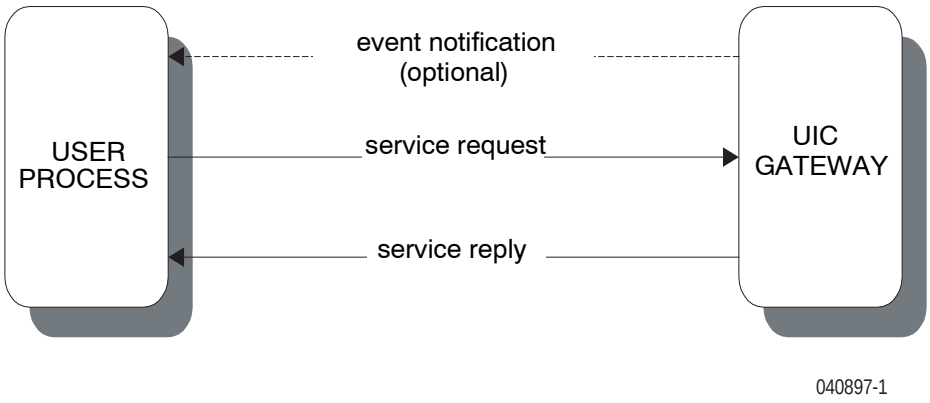


Fig. 6 - Gateway service access paradigm

C.2.4.5 - Process variables

This point describes an interface of the UIC Mapping Server to the vehicle bus. This is an example how it may be realized. However, its description is not normative as the *UIC Leaflet 556* makes no normative prescriptions about the vehicle bus.

The choice to run the UMS as a pure server for user services has one major impact on the system design: by default, there is no mechanism for event notification, that informs the user that something happened.

1. The restriction to run the gateway as a server only is motivated by the avoidance of complex and troublesome communication relations among the gateway itself and between gateway and user processes. Imagine for instance, the gateway distributes the new NADI version after an inauguration autonomously to its subordinated vehicle bus stations. Then it has to have the knowledge about all connected vehicle bus stations, which means to have a station list preconfigured. Each time there is a change in vehicle bus configuration this list has to be updated.

To avoid a burdensome polling of the UMS state to get informatons about new events, a few process variables are defined (see Tables 3 and 4 - page 64) which can be used to signal a change of some important status information on the vehicle bus. Each vehicle bus station interested in event notification has to cyclically poll these process variables. Signalling status changes over the WTB is not supported[1].

C.2.4.5.1 - Exported PVs

These PVs should be exported to the local vehicle bus in order to signal important events.

Table 3 : Process variables exported by the USM

PV name	*Type*	*Values*	*Remarks*
UIC_PV_CONFLICT	BOOLEAN1	UWTM_NOCONFLICT = 0 UWTM_CONFLICT = 1	Indicates a strong master conflict.
UIC_PV_OPMODE	ENUM4	UWTM_SLAVE = 0 UWTM_WEAK_MASTER = 1 UWTM_STRONG = 2 UWTM_PASSIVE = 3	Indicates current operational mode.
UIC_PV_STATE	ENUM4	UWTM_IDLE = 1 UWTM_PASSIVE = 4 UWTM_REGULAR = 2 UWTM_RESTRICTED = 3 UWTM_SINGLE = 5	Indicates the current UWTM state.
UIC_PV_LEADING	BOOLEAN1	UWTM_NOT_LEADING = 0 UWTM_LEADING = 1	Indicates whether vehicle is leading or not.
UIC_PV_TOPO_ COUNT	UNSIGNED8	0..63	Actual topo count (0 = invalid).
UIC_PV_NUMBER_ VEHICLES	UNSIGNED8	1..63	Number of vehicles.
UIC_PV_ ORIENTATION	ANTIVALENT2	UWTM_INVERSE = 0 UWTM_NOT_INVERSE = 1	Orientation with respect to the TCN.

1. But of course other mechanisms exist. If for instance a remote user is not informed about a NADI change and wants to send a message based on the old NADI address information, it will get an error indication from the local gateway which indicates a mismatch of the topo counter. Hence the user knows that something must be changed and can ask for new NADI address information (see also the description of the UIC Mapping Server Remote Interface).

C.2.4.5.2 - Imported PVs

The following PVs can be imported by the UMS allowing a rudimentary WTB control via process variables (This for instance can be important for vehicle bus devices that are not capable to send message data).

Table 4 : Process variables optionally imported by the UMS

PV name	*Type*	*Values*	*Remarks*
UIC_PV_OPMODE_SELECT	ENUM4	UWTM_PASSIVE = 3 UWTM_SLAVE = 0 UWTM_WEAK = 1 UWTM_STRONG = 2	Select operational mode.
UIC_PV_LEADING_REQUEST	BOOLEAN1	UWTM_NOT_LEADING = 0 UWTM_LEADING = 1	Request train leadership.
UIC_PV_LEADING_DIRECT	BOOLEAN1	UWTM_DIR1 = 0 UWTM_DIR2 = 1	Indicates direction of the leading vehicle in the case of a trainset with two driver´s cabs.
UIC_PV_SLEEP	BOOLEAN1	UWTM_NOSLEEP = 0 UWTM_SLEEP = 1	UWTM_NOSLEEP means no sleep request or, if already requested, cancelling the request.
UIC_PV_INAUG	BOOLEAN1	UWTM_ALLOWED = 0 UWTM_INHIBITED = 1	Allow/inhibit TCN inauguration.

Restriction:

The UMS supports only the PV import from one vehicle bus source

C.2.5 - UIC Mapping Server - User interface

As already mentioned in the previous section all user services are callable by the means of TCN messages. This chapter gives an overview of the messages that shall be used for this purpose.

C.2.5.1 - Message format

All user services are invoked by message data calls requesting a specific service, and the result is returned with the reply. The format of the telegrams transfered over the WTB follows the format specified in *UIC Leaflet 556*:

Octet		
0	Owner	UIC application code
1	Reserved	always 0
2	Destination vehicle	range of group number
3	Destination function	Function number according to UIC Leaflet 556
4	Source vehicle	range number
5	Source function	function number according to UIC Leaflet 556
6	Code (MSO)	message code (most significant octet)
7	Code (LSO)	message code (least significant octet)
8	Status (command)	Status or command (message type dependent)
9	Reserved	always 0
10	Message body	dependent on the message type

Fig. 7 - E telegram header

C.2.5.2 - UMS Messages - Summary

The following table lists all the user service telegrams that can be sent to the UIC gateway. From the gateway point of view it doesn´t matter whether these telegrams originate from the local vehicle (via MVB or local) or another vehicle (via WTB). But not all of the telegrams are allowed to be sent between vehicles (see below).

Table 5 : UIC Mapping Server Telegrams

Service message	*Code*	*Direction*	*Access rights*			*Description*
			local	test	regular	
UIC_FC_READ_NADI	0x0001	Aufruf		X		request to read the complete NADI
UIC_FR_READ_NADI	0x0A01	Antwort		X		copy of the currently valid NADI
UIC_FC_READ_UWTM_ STATE	0x0022	Aufruf		X		request for reading the UWTM state
UIC_FR_READ_UWTM_ STATE	0x0A22	Antwort		X		UWTM state
UIC_FC_CHANGE_OMODE	0x0003	Aufruf		X		command to change the operational mode of the gateway
UIC_FR_CHANGE_OMODE	0x0A03	Antwort		X		status
UIC_FC_READ_UWTM_ TOPOGRAPHY	0x0004	Aufruf		X		request to read the TCN topography
UIC_FR_READ_UWTM_ TOPOGRAPHY	0x0A04	Antwort		X		TCN topography
UIC_FC_READ_GROUP	0xB001	Aufruf			X	request to read the members of a group
UIC_FR_READ_GROUP	0xBA01	Antwort			X	members of a group
UIC_FC_READ_GROUP_ LIST	0xB002	Aufruf			X	request to read a list of all defined groups
UIC_FR_READ_GROUP_ LIST	0xBA02	Antwort			X	status
UIC_FC_WRITE_GROUP	0xB003	Aufruf			X	request to write the members of a group
UIC_FR_WRITE_GROUP	0xBA03	Antwort			X	status
UIC_FC_WRITE_GROUP_ ALL	0xB004	Aufruf			X	request to write a list of groups
UIC_FR_WRITE_GROUP_ ALL	0xBA04	Antwort			X	status
UIC_FC_DELETE_GROUP	0xB005	Aufruf			X	request to delete a group
UIC_FR_DELETE_GROUP	0xBA05	Antwort			X	status
UIC_FC_DELETE_GROUP _ALL	0xB006	Aufruf			X	request to delete all groups
UIC_FR_DELETE_GROUP _ALL	0xBA06	Antwort			X	status
UIC_FC_DEL_CONFIG	0xF001	Aufruf			X	request to delete the confirmed configuration
UIC_FR_DEL_CONFIG	0xFA01	Antwort			X	status
UIC_FC_WRITE_ CORRECTION	0xF002	Aufruf			X	request to write correction information (remark: the telegram causes implicitely a UIC inauguration)

Table 5 : UIC Mapping Server Telegrams

UIC_FR_WRITE_ CORRECTION	0xFA02	Antwort			X	status
UIC_FC_WRITE_RES_NR	0xF003	Aufruf			X	request to write the vehicle numbers for seat reservation (remark: the telegram causes implicitely a UIC inauguration)
UIC_FR_WRITE_RES_NR	0xFA03	Antwort			X	status
UIC_FC_SLEEP	0xF004	Aufruf			X	command to set or cancel sleep request
UIC_FR_SLEEP	0xFA04	Antwort			X	status
UIC_FC_INAUGURATION_ CONTROL	0xF005	Aufruf			X	command to inhibit or allow inauguration
UIC_FR_INAUGURATION_ CONTROL	0xFA05	Antwort			X	status
UIC_FC_INAUGURATION_ ENFORCE	0xF006	Aufruf			X	command to enforce a new inauguration
UIC_FR_INAUGURATION_ ENFORCE	0xFA06	Antwort			X	status
UIC_FC_MULTICAST_ CREATE	0xF007	Aufruf			X	request the exchange of a multicast message
UIC_FR_MULTICAST_ CREATE	0xFA07	Antwort			X	status
UIC_FC_LEADING	0x00F0	Aufruf	X			command to set or reset the leading request bit
UIC_FR_LEADING	0x0AF0	Antwort	X			status
UIC_FC_CONVERT_UIC_ TCN	0x00F1	Aufruf	X			request to convert a UIC address to TCN address
UIC_FC_CONVERT_UIC_ TCN	0x0AF1	Antwort	X			TCN address(es)

Access rights:

local: the telegram is not allowed to be sent on WTB.

test: the telegram can be sent on WTB only for test purpose (e.g. conformance test)

regular: the telegram can be sent on WTB

The specification of the regular and test telegrams is given in Apendix A - page 35.

Local telegrams are defined in point C.2.8 - page 82.

C.2.6 - Software

NB : The implementation of the UIC Mapping Server may deviate from the description given hereinafter in respect of the naming conventions of internal functions and the internal processing, like for instance the interfaces between the MS components.

C.2.6.1 - General UIC Gateway software structure

The UIC Mapping Server software can not be considered isolated from any other software component on the UIC gateway. Therefore it is necessary to mention all that components beside the UIC Mapping Server that build up a UIC gateway.

The UIC Gateway consists of the following major components:

- **TCN Protocol Stack** consisting of WTB Link Layer, MVB Link Layer (optionally, or another vehicle bus interface), Real Time Protocols (with modules "messenger" and "RTP PD interface") and Network Management Agent (NMA).

- **Process Data Marshalling** for process data import and export.

- **UIC Mapping Server** with the subcomponents UIC Train Bus Configurator, UIC NADI & Group Server, UIC Agent, UIC WTB Manager and UIC Intelligent Multicast Server.

- **Node Supervisor** with the Node Supervisor Data Base.

- **Realtime Operating System** with the PIL interface.

Besides the UIC Mapping Server which acts as the agent for incoming user service requests, there might be a corresponding managing process situated on MVB stations which allows to submit user service requests to the gateway. This managing process is called the **UIC Mapping Server Remote Interface** (UMSI).

The following points describe the individual components of the UIC mapping server.

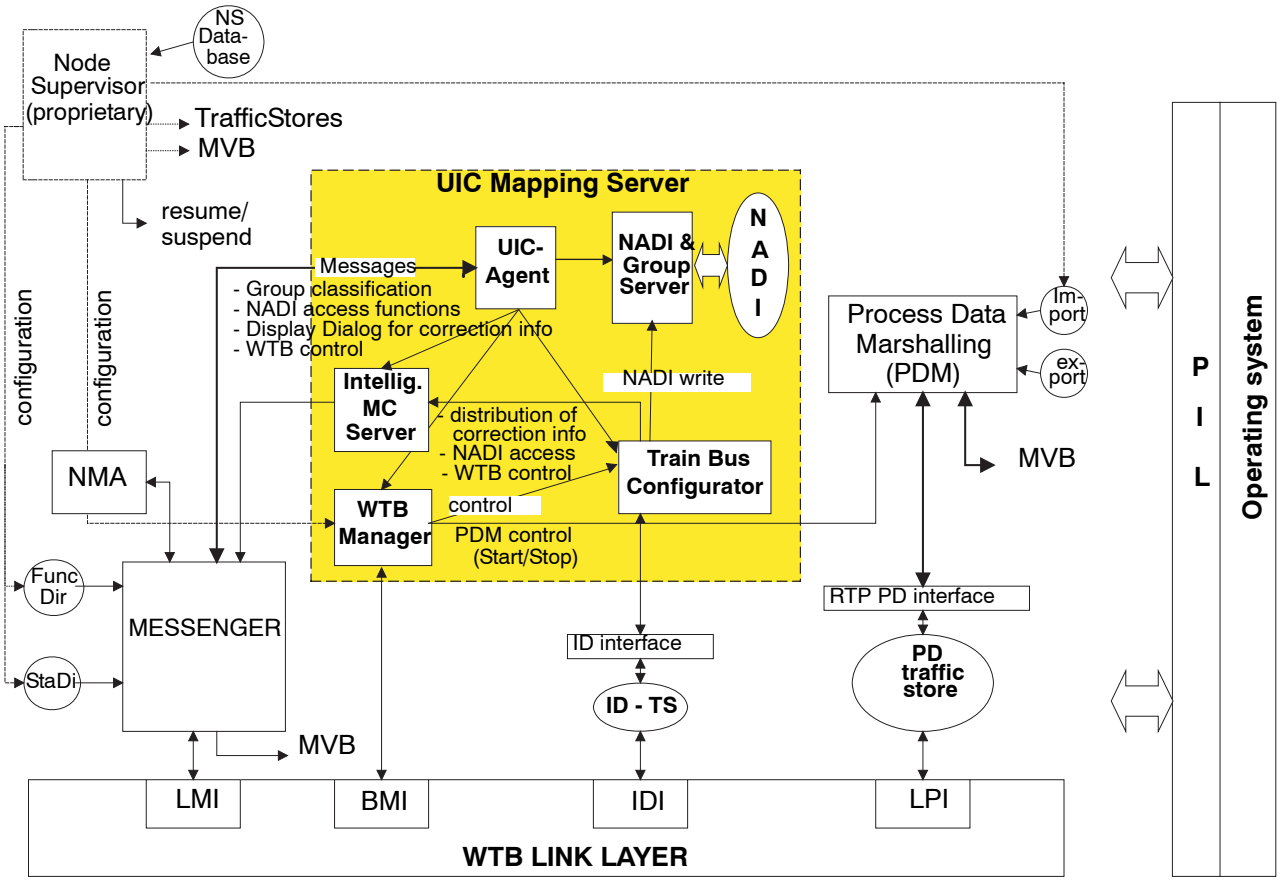


Fig. 8 - UIC gateway software structure (data flow)

C.2.6.2 - Basic Design Principles

C.2.6.2.1 - E-telegram handling

As already outlined, the UMS offers its user services by the means of E telegrams. A service is requested by sending the corresponding E telegram to the UMS, and the UMS returns the result in the related answer E telegram. For sending and receiving E telegrams the message call/reply mechanism of the TCN RTPs shall be used.

To manage messages one has to consider two aspects: first, the buffer management and second, the message conversion from a WTB byte (or bit-) stream to a high level language data structure suited to the processing of the content of the message. The conversion must take into consideration the data presentation of the processing unit (e.g. little endian or big endian format or alignment) compared to the presentation of a message on the WTB (big endian, byte stream). For message conversion, each function provider of the UMS incorporates a "Message Converter". Concerning call_messages, this converter inputs the message byte stream and outputs a corresponding data structure. For Reply_Messages, the procedure is reversed.

In detail, the handling of messages shall be performed as follows :

Upon reception of the first message data frame (which is the Call_Request frame that contains the overall length of the message), the RTPs will allocate the buffer for that Call_Message[1]. A pointer to the message buffer is passed to the UIC Agent ("am_receive_confirm"), which then calls the corresponding service provider ("<mod>_request")[2] for processing the Call_Message and generating the Reply_Message. The message converter of the service provider first allocates a buffer for the data structure, converts the message and passes a pointer to the data structure to the service provider function. This function returns a pointer to the reply message data structure. The message converter then allocates the buffer for the Reply_Message byte stream, converts the message and returns to the RTPs which thereafter start sending the Reply_Message.

The correct handling of message processing especially with respect to the buffer management and conversion are depicted in Fig. 9 - page 70.

1. The am_receive_request function of the RTPs message application interface has to be called with a NULL-pointer for the Call_Message so that the RTPs are responsible for buffer allocation
2. <mod> to be replaced by UWTM, UTBC, UNGS and UIMCS.

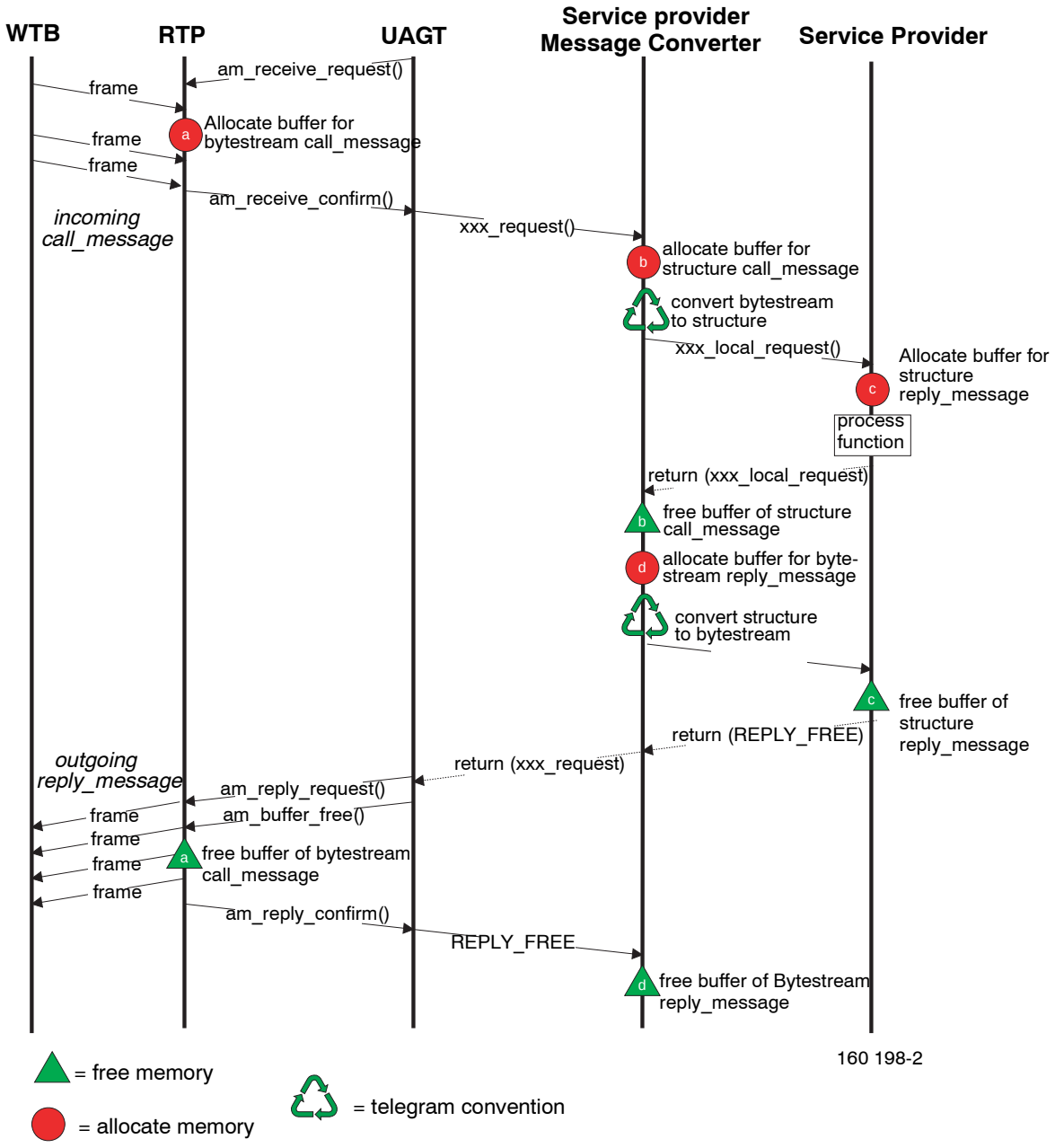


Fig. 9 - Message handling

The procedure of message conversion highly depends on the used hardware platform and therefore must be tailored while porting the software to a specific hardware.

In the minimal version (hardware supports big endian format and natural data aligment[1]) the message converter just outputs the input pointer without any processing (just type casting). In this case, there is no extra buffer allocation for the data structure necessary

1. Natural alignment of data types means, to align 16-bit data types to an address divisible by 2, 32-bit types to an address divisible by 4 and so on.

C.2.6.2.2 - Correction and reservation handling

The procedure for correcting the inauguration result and submitting seat reservation shall be handled as follows:

1. The display sends a multicast_create message enveloping the corresponding E-telegram (either UIC_FC_WRITE_CORRECTION or UIC_FC_WRITE_RES_NR) to the gateway with TCN address 01 (TCN master)[1].

2. The UIMCS of the TCN master then first distributes the correction or reservation telegrams to all other gateways. Thereafter, a new UIC inauguration will be triggered automatically, resulting in a new NADI version.

C.2.6.2.3 - Single nodes

Some particularities have to be observed with single nodes. A single node is a WTB node that is not connected to any other WTB node, thus there is no data transmission via WTB. But devices inside the vehicle, or vehicles within a trainset, need to have a valid NADI that holds the vehicle or trainset related information. Therefrom the following requirements result:

- even in the case of a single node there must be a valid NADI;

- from an application point of view, the single node gateway is in regular operation.

A gateway becomes a single node if it does not succeed in inaugurating other nodes or beeing inaugurated by another node within a time of 5 seconds.

C.2.6.2.4 - Group information handling

The UIC Mapping Server incorporates a group data base that is used to store user defined groups. The UIC Mapping Server provides services for reading, writing and deleting groups.

The handling of groups like the procedure of group definition or the adjustment in case of train coupling is not part of the UIC Mapping Server. This shall be done by a seperated Group Server.

C.2.6.3 - UIC Mapping Server Interfaces

The UMS is embedded in the UIC gateway firmware and such interfaces with several other components. There are mainly four external interfaces:

TCN Protocol stack	This interface is defined in the TCN standard and comprises the RTP interface (TCN part 2), the WTB interface (TCN part 4) and the MVB interface (TCN part 3).
Process Data Marshalling	This interface comprises functions for PDM control and PV read and write access. Has to be defined with the PDM.

1. The reason for this is to put concurrent requests from different displays into sequence.

Node Supervisor	The NSDB holds all the configuration data that are needed by the UMS. Thus the UMS defines several functions to be called by the node supervisor.
Operating System	Towards the OS an abstract interface has been defined which is also used by the TCN protocol stack. This interface is called the PIL (Processor Interface Library).

Apart from the external interfaces there are also internal interfaces between the different subcomponents of the UMS. These interfaces are defined with the subcomponent specifications. For conveniance, Fig. 10 gives an overview of these internal function interfaces.

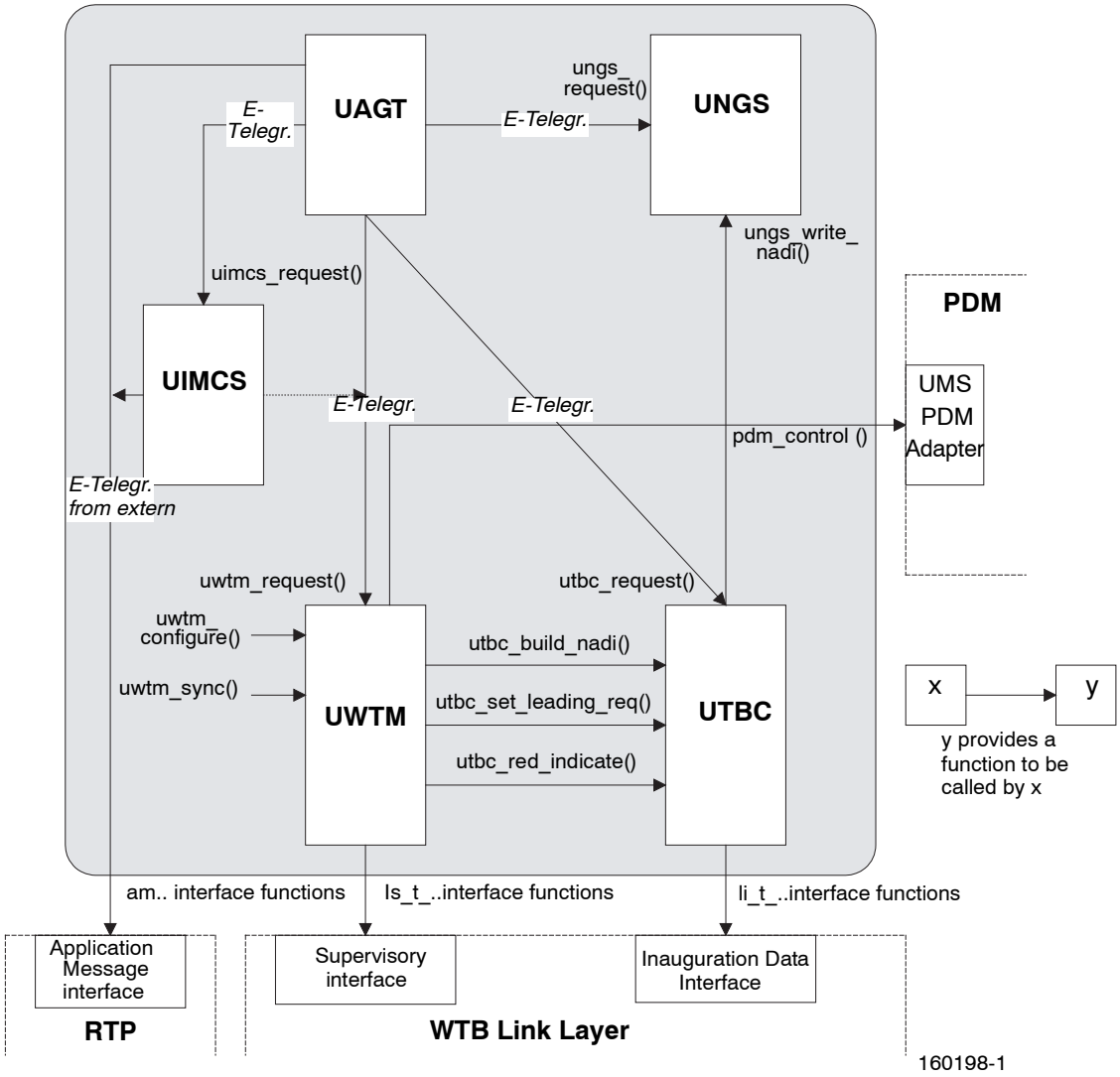


Fig. 10 - Functional interfaces

C.2.6.4 - Description of UIC Mapping Server components

This point gives a rough definition of the several components the UIC Mapping Server is composed of.

C.2.6.4.1 - UIC Agent

Function

The UIC Agent serves as the general UMS replier for all exchanged UIC_Messages which are directed to or from the UIC gateway mapping server function in order to manipulate any UIC Mapping Server object. For message exchange the Messages Services of the TCN will be used.

The UIC Agent can be seen as a message dispatcher in analogy to the NMA (NMA = Network Management Agent. See [TCN] for further information). Incoming Call_Messages are decoded, the corresponding object is accessed and a Reply_Message is sent back with the result of the service. Object manipulation is done via functions offered by the different UMS sub-components (so-called "function providers"). For our purpose the following instances act as corresponding function providers:

- the UIC NADI & Group Server
- the UIC Train Bus Configurator
- the WTB Manager
- the Intelligent MC Service.

Because many incoming calls can be in parallel, the UIC Agent has to synchronise such calls in a way that function calls to the function providers are put in sequence. Thus the different function providers need not to have to synchronize the function calls by themselves.

For details about the message handling see point C.2.6.2.1 - page 69.

Interface

The UIC Agent is directly connected to the TCN messenger as a replier function.

For generating the Reply_Messages it contacts the correspondent function providers by calling the <mod>_request[1] function. The function provider generates the reply and returns immediately, that means it is not allowed to go to a "wait" position (non-blocking function).

1. <mod> to be replaced by UWTM, UNGT, UTBC or UIMCS.

C.2.6.4.2 - Function Providers

Each of the function providers shall obey the following principles:

Message conversion and buffer management

The function provider has to convert the messages from bytestream to data structure and vice versa. The principle is described in Fig. 9 - page 70. Here, the procedure shall be described with the following pseudo-code example:

```
<mod>_request (..)
{
    allocate_buffer(B);          /* /* allocate buffer for the Call_Message data structure */
    NTOH_<telegram_type> ();     /* /* convert from bytestream to data structure */
    <telegram_type>_request();   /* /* call service provider function for processing the Call_Message and generating the
                                    Reply_Message */
    free_buffer(B);              /* /* free the buffer of the Call_Message data structure */
    allocate_buffer(D);          /* /* allocate buffer for the Reply_Message bytestream */
    NTOH_<telegram_type> ();     /* /* convert from structure to bytestream */
    (*free_replymsg)();          /* /* call call-back function to free the buffer of Reply_Message data structure*/
}
```

The reply message is generated by the telegram type related request function:

```
<telegram_type>_request (..)
{
    allocate_buffer(C);          /* /* allocate buffer for the Reply_Message data structure */
    process_call_message();
    generate_reply_message();    /* /* generate the Reply_Message data structure */
}
```

Function <mod>_request

The prototype of the <mod>_request function shall be as follows:

```
<mod>_RESULT <mod>_request                                  (
    void                    * p_uic_callmsg,                /* ptr to call msg */
    CARDINAL32                uic_callmsg_size,             /* call message size*/
    AM_ADDRESS              * caller,                       /* caller address */
    void                   ** pp_uic_replymsg,              /* ptr to reply msg */
    CARDINAL32              * p_uic_replymsg_size,          /* size of reply msg*/
    REPLY_FREE              * p_reply_free_fct              /* buffer free fct. */
);
```

Typ REPLY_FREE

This function shall be used to free previously allocated buffer.

```
typedef void (*REPLY_FREE) (void *);
```

Function <telegramm_type>_request

The prototype of the <mod>_local_request function shall be as follows:

```
<mod>_RESULT <telegram_type>_request  (
   UIC_T_<telegram_type>                  *p_uic_callmsg,
   UIC_T_<telegram_type>                  **pp_uic_replymsg,
   REPLY_FREE                             *p_reply_free_fct
);
```

Functions for conversion

For each telegram type, there must be two conversion functions:

Function

```
<mod>_RESULT NTOH_<telegram_type>     (
   UN-SIGNED8                          *p_bytestream_in,
   CARDINAL32                           bytestream_length,
   UIC_T_<telegram_type>               *p_struct_out
);
```

for conversion from bytestream to structure

and

Function

```
<mod>_RESULT HTON_<telegram_type>     (
   UIC_T_<telegram_type>               *p_struct_in,
   UN-SIGNED8                          *p_bytestream_out,
   CARDINAL32                           bytestream_length
);
```

for conversion from structure to bytestream.

Rules for message reply

With the exception of the UIMC and the UWTM, all other service providers must not wait for the result for generating the reply, because their services are defined in a way that the result can be immediately achieved. Normally its just a copying of a buffer region (e.g. a NADI copy).

For the UIMC and the UWTM, however, things are not so easy because most of the services these two are confronted with are complex and require a multi-step handling. In these cases, only the *acceptance* of the service will be replied. If a caller wants to know about the success of a call, it has to issue a seperate status poll.

Table 6 : Error handling

Type of error	***Responsible***	***Handling***
call_message shorter than 10 octets	UAGT	The UAGT shall free the buffer of the call_message and cancel the connection
invalid call_message code	UAGT	return reply_message (header only) with error indication in the status octet
invalid source/destination address, invalid status	Function provider	return reply_message (header only) with error indication in the status octet
error during telegram processing	Function provider	return reply_message (header only) with error indication in the status octet
error with buffer allocation	Function provider	error return in the <mod>_request function. The UAGT shall free the buffer of the call_message and cancel the connection.
unability to reply even a header indicating an error (static memory error)	Function provider UAGT	error return in the <mod>_request function. The UAGT shall free the buffer of the call_message and cancel the connection.

C.2.6.4.3 - UIC WTB Bus Manager (UWTM)

Function

The UIC WTB bus manager is responsible for the configuration, start-up (e.g. STRONG/WEAK/SLAVE/PASSIVE mode) and control (e.g. inhibit/allow) of the WTB link layer. For this reason it shall implement a state machine which follows the states of the WTB Link Layer BMI. To properly start-up the WTB, it shall receive its configuration data from the Node Supervisor, and it shall work in a way that it performs WTB control completely autonomously thereafter.

Interfaces

1. Exported functions

uwtm_configure ()	used for UWTM configuration. To be called by the Node Supervisor.
uwtm_sync ()	lifesign indication for redundancy control
uwtm_request()	called by the UAGT for E-telegram processing (user services)

2. Supported user services

   The UIC WTB Manager shall provide uwtm_request() functions to the UIC Agent. The UWTM shall handle the following user services:

   - change_omode
   - inauguration_control
   - sleep
   - inauguration_enforce

- read_uwtm_state
- read_uwtm_topo
- leading

3. Controlling with process variables

   In parallel to the user services the WTB manager provides process variables for easy remote control and state reading (see Table 2 - page 61 and 3 - page 63).

C.2.6.4.4 - UIC Intelligent Multicast Server (UIMCS)

Function

The UIMCS serves as a central instance for multicasting messages using the TCN single-cast protocol. Messages are multicasted to all gateways connected to the WTB.

For the UIC Mapping Server a multicast service is required for:

- distribution of correction info to all other vehicles;

- distribution of group information to all other vehicles.

Interfaces

1. Exported functions
   uimcs_request () called by the UAGT for E-telegram processing (user services)
2. Supported user services
   multicast_create

C.2.6.4.5 - UIC Train Bus Configurator

Function

The UIC Train Bus Configurator is responsible for the UIC train inauguration, that means to distribute and collect vehicle descriptors and to assign the correct UIC address to nodes and vehicles. The basic concept is given in point C.1 - page 37. The whole functionality of the UTBC is devided into two tasks:

- After TCN inauguration, the UTBC calculates the new NADI (algorithm see point C.1) and stores it in the NADI database directly. This algorithm is implemented within one function that will be called by the UWTM each time a new topography has been distributed.

- Correction information is passed to the UTBC by the UIC Agent, e.g. from any display. This correction information is processed by a function that is called by the UIC Agent with the UTBC as the corresponding function provider.

Processing of inauguration data and processing of correction information are completely different tasks with disjunctive functions and operating on disjunctive data, therefore there is no special synchronisation needed. None of the tasks needs to incorporate a state machine, that is why they can be implemented as non-blocking functions.

Interfaces

1. Exported functions

utbc_request()	called by the UAGT for E-telegram processing (user services)
utbc_set_leading_req()	to be called by the UWTM in the case leading is requested or a leading request shall be canceled
utbc_build_nadi()	to be called by the UWTM each time a new topography has been distributed
utbc_delete_config ()	to be called by the UWTM after reception of an order to delete the confirmed configuration
utbc_red_indicate()	to be called by the UWTM if a redundancy switch-over occurs

2. Supported user services

- write_res_nr
- write_correction

C.2.6.4.6 - NADI and NADI & Group Server

Function

The NADI is the database for all train-describing data. Thus it contains information about the train topography, the individual vehicles within the train ("vehicle descriptors") and the group memberships of vehicles. The NADI is accessable by functions that are supplied by the "UIC NADI & Group Server". Thus the NADI is completely encapsulated in an "object oriented" way and the data structure itself is hidden to the user.

Each time a new inauguration takes place the UTBC generates a new version of the NADI. This version can either be confirmed by the application or not. This new version is stored in the NADI database. It shall be possible to read from and write to the database in parallel in a consistent and non-blocking way.

Concerning group handling the NADI server stores new groups in the NADI database. These groups are delivered by the seperated Group Server[1]. After storage of the groups, the group membership information can be read at any time from local instances.

Interface

1. Exported function

ungs_write_nadi()	called by the UTBC each time a new NADI version has been calculated
ungs_request()	called by the UAGT for E-telegram processing (user services)

1. It is assumed that group memberships are inserted either by the train attendants or by any other mechanism. The handling of groups is on behalf of the Group Server, that is not a part of the UIC Mapping Server.

2. Supported user services

- read_nadi
- write_group
- write_group_all
- read_group
- read_group_list
- delete_group
- delete_group_all

C.2.7 - Other components related to the UIC Mapping Server

This point presents a brief informal description of the UIC gateway software components which are not part of the UIC mapping server.

C.2.7.1 - UIC Mapping Server Remote Interface (UMSI)

The UIC Mapping Server standardizes the remote user access with TCN messages. It is not specified herein, nor it is requested, how a remote user (e.g. an MVB station) may initiate such message exchange with the UIC Mapping Server. But it would be convenient to provide a unique and simple to use functional interface to remote user processes that allows to access the UIC Mapping Server services in a general way. This functional interface will be called the **UIC Mapping Server remote interface**. This interface consists of a set of C-functions (assembled in a library) that can be called by user processes and which handle all the message data exchange with the UIC Mapping Server, thus discharging the user from assembling messages and managing message transfer by himself.

Besides the necessary message data exchange with the UIC Mapping Server for UMS service calls, the UMSI shall also be reponsible for the mapping of UIC addresses to TCN addresses when user processes apply message calls to other users somewhere in the TCN. Thus it should provide functions for message data sending similar to that defined by the TCN RTPs, but with the distinction that UIC addresses are used rather than TCN ones.

Example:
The user function "Door" in vehicle 8 wants to send a diagnostic message to the user function "diagnosis" in the leading vehicle. For this, it calls a function:

umsi_call_request (destination_vehicle = leading vehicle,
destination_function = Diagnosis
....

The UMSI converts the UIC addresses to the TCN addresses (e.g. leading vehicle = WTB node 62, diagnosis = TCN function "34") and calls the corresponding TCN function for message transmission. In the other direction, the same is done with the reply.

Hence the user only has to think in UIC addresses and needs not to care about TCN addressing, making user programs more easy to understand.

The UMSI also takes care of holding the newest NADI version in order to prevent missadressing of messages.

C.2.7.2 - Other Gateway software components

This point describes the software components which are needed besides the UIC Mapping Server to build a UIC conformable gateway. They are included herein for information, but they do not belong to the UIC Mapping Server software package.

C.2.7.2.1 - WTB Link Layer

The WTB Link Layer is responsible for the WTB inauguration and the sending and receiving of PD and MD frames. The WTB Link Layer shall be in accordance to the *IEC 61375, part 4*. To adapt to a certain hardware plattform all HW related functions are seperated and encapsulated into a driver software, thus that the WTB link layer is completely HW independent.

The WTB link layer provides the following interfaces:

- Bus Management Interface BMI,
- PD Traffic Store,
- Inauguration Data Interface,
- Driver interface,
- PIL interface.

The inauguration data interface provides functions for reading and writing inauguration data datasets. These datasets are used by the UTBC to calculate a valid NADI.

C.2.7.2.2 - Messenger and RTP

The Messenger software module takes care of routing and transporting TCN message data within the TCN. It shall be in accordance to the *IEC 61375, part 2*.

C.2.7.2.3 - Process Data Marshalling

The process data marshalling is responsible for data marshalling between MVB (or any other vehicle bus) and the WTB. Marshalling means one of the following:

- copy selected process variables from WTB to MVB and vice versa;
- apply functions to one or a set of process variables and sending the function result on either WTB (exporting) or vehicle bus (inporting). Allowed functions are defined in *UIC Leaflet 556*.

The PDM shall support import and export of process data frame types 0 (invalid), 1 (leading vehicle), 2 (traction vehicle) and 3 (other).

Towards the UMS, the PDM shall provide a function pdm_control that will be used by the UWTM to start and stop the PDM and to select the marshalling mode.

C.2.7.2.4 - Node Supervisor and NSDB

The node supervisor is a "virtual" instance because its implementation greatly depends on the used hardware plattform and therefore can not be a common component of the gateway. It mainly has the task to start the gateway in a proper way ("main task"), that means both initialization and configuration. Configuration data are held in the node supervisor database (NSDB). The NSDB contains configuration data for the following components:

- Messenger,
- PDM and its database,
- Mapping Server,
- WTB configuration,
- MVB device address list,
- WTB traffic store,
- MVB traffic store.

C.2.7.2.5 - Operating System and PIL

Because selecting an operating system is a choice of the hardware deliverer there shall be a standardized processor interface library (PIL) available which provides a set of hardware-independent operating system services like:

- task lock/unlock,
- copy functions,
- interrupt functions,
- semaphores,
- timers,
- queues.

The PIL shall be in accordance to the JDP specification. All generic software components of the UIC Mapping Server will use the PIL.

C.2.8 - Specification of the local Mapping Server E Telegrams

The local E telegrams are restricted to the vehicle internal application and shall not be sent on the WTB.

N°	Purpose	Information path				Telegram type	Representation and processing								
		Source		Destination			Origin of information	Meaning	Data type/ values	Octet	Bit	Code/ value	PDM	Time out [a] [sec.]	Substitute value
		fct	veh	veh	fct										
1	2	13	3	9a	14	10	15	16	16a	17	18	19	20	21	22
	UIC Mapping Server - Local telegrams														
1.1	set/reset attribute "leading vehicle"	NN	1..63	1..63	15	E		Code	Unsigned 16	7+8		0x00F0			
								State	Unsigned 8	9		0: reset 1: set			
								Direction (for trainsets with 2 driver cabs)	Unsigned 8	11		0: direction 1 1: direction 2			
1.1A	acknowledge	15	1..63	1..63	NN	E		Code	Unsigned 16	7+8		0x0AF0			
								State	Unsigned 8	9		0: accepted >200: error			
1.2	request UIC to TCN conversion	NN	1..63	1..63	15	E		Code	Unsigned 16	7+8		0x00F1			
								UIC Address	Unsigned 8	11		0..63			
1.2A	acknowledge	15	1..63	1..63	NN	E		Code	Unsigned 16	7+8		0x0AF1			
								State	Unsigned 8	9		0: accepted >200: error			
								topo_count	Unsigned 8	11					
								UIC Address	Unsigned 8	12		0..63			
								number n of the returned TCN addresses	Unsigned 8	13		0: no found 1..63			
								TCN Address 1 TCN Address n	Unsigned 8	14 . 14+(n-1)		1..63			

a. i = Ini-value

Record of updates:

Version	Date	Change	Reason for the change
002.01	01.11.2004	Change of the layout according to M1	New layout based on UIC guide M1
		Adoption of a history of updates	Increasing the usability
		Adoption of a revision number	Enhancement and redesign of the versioning
002.02	01.08.2005	Change of the layout	UIC requirement
		Renumbering of the Appendices	

C.3 - Construction of the inauguration frame (Version 002.03, valid from 01.03.2009)

Octet	Bit	Data type / Scope of values	Code	Meaning	Source Gateway (=Trainset)	Source vehicle	Serial No from Appendix E
1	0-3	ENUM4	1	Type of application UIC[a]	X		
	4-7		0	Reserve			
2		ENUM8	NN	UIC inauguration frame - version number	X		133
3	0-6	"WORD7"	NN	TCN inauguration address	X		514
	7	BITSET1	0	Orientation relative to the TCN master = reference direction train bus node is opposite to the reference direction TCN	X		505
			1	= reference direction train bus node is the same as the reference direction TCN			
4		UNSIGNED8	124	Length of the application data in the inauguration frame	X		
5		ENUM8	NN	UIC Code of the operating RU	X		135
6		ENUM8	NN	UIC Code of the owning RU	X		136
7		ENUM8	NN	National application identification	X		138
8		ENUM8	NN	National telegram version	X		137
9		INTEGER8	NN	NN > 0: Number of vehicles monitored from this node (1…6) NN ≤ -2: Number of nodes in this vehicle is |NN|	X		139
10	0	BITSET8	1	Delete confirmed configuration	X		501
	1		1	Configuration unknown	X		502
	2		1	UIC address placed	X		503
	3		1	Configuration stored	X		504
	4		1	Vehicle was leading vehicle	X		510
	5		1	Vehicle wants to be leading vehicle	X		511
	6			if bit 10/5 = 0: meaningless if octet 9 < 2: meaningless	X		515
			0	In the trainset, vehicle with cab 1 wants to be leading. Note: Cab 1 is the front cab in the reference direction of the trainset			
			1	In the trainset, vehicle with cab 2 wants to be leading			
	7		1	Inauguration triggered by redundancy change over	X		516
11		UNSIGNED8	NN	Confirmed UIC address of the vehicle which is the location of the train bus node (property 141)	X		507
			0	No confirmed UIC address known			
12		UNSIGNED8	NN	Confirmed number of vehicles = NN	X		508
			0	No confirmed number of vehicles known			

a. A vehicle which does not contain the value 0x1 in the lower four bits is not a UIC vehicle. That means that for this train no valid NADI can be produced. The upper four bits are reserve.

Octet	Bit	Data type / Scope of values	Code	Meaning	Source Gateway (=Trainset)	Source vehicle	Serial No from Appendix E
13		Array BITSET8		Confirmed position of the vehicles which cannot be reached over the train bus:		X	509
14							
15			0. Bit	= Vehicle position 1			
16			1. Bit	= Vehicle position 2			
17							
18			n. Bit	= Vehicle position n+1			
19							
20			63. Bit	= Vehicle position 64			
21		ENUM8	NN	R telegram version number	X		134
22	0	BITSET8	1	Vehicle has sealed toilets	X		17
	1		1	Vehicle is pressure tight	X		19
	2		1	Vehicle has side sensitive door locking over the train bus	X		21
	3		1	Vehicle has side sensitive door not locking over the train bus	X		22
	4		1	Vehicle supports "door close"	X		23
	5		1	Vehicle supports door close monitoring	X		24
	6		1	Vehicle supports prevent/allow WC use	X		29
	7		1	Vehicle supports lighting control over train bus	X		30
23	0	BITSET8	1	Vehicle supports internal loudspeaker (choice receipt)	X		31
	1		1	Vehicle supports internal loudspeaker (obligation receipt)	X		32
	2		1	Vehicle supports speech connection with leading vehicle	X		33
	3		1	Vehicle supports speech connection between leading vehicle and trailing tractive unit	X		34
	4		1	Vehicle has external loudspeaker	X		35
	5		1	Vehicle supports external loudspeaker control	X		36
	6		1	Vehicle supports public address in individual vehicles/vehicle groups	X		37
	7		1	Vehicle supports warm chain	X		142
24	0	BITSET8	1	Tractive unit with electric drive	X		38
	1		1	Tractive unit with diesel engine drive	X		39
	2		1	Vehicle has ≤ 2 pantographs	X		40
	3		1	Vehicle has > 2 pantographs independent of each other	X		41
	4		1	Vehicle can remotely control the drive of other (electric) tractive units with control system 1e over the train bus	X		44
	5		1	Vehicle can remotely control the drive of other (diesel) tractive units with control system 1d over the train bus	X		45
	6		1	Vehicle can remotely control the drive of other (electric) tractive units with control system 2e over the train bus	X		46
	7		1	Vehicle can remotely control the drive of other (diesel) tractive units with control system 2d over the train bus	X		47

Octet	Bit	Data type / Scope of values	Code	Meaning	Source Gateway (=Trainset)	Source vehicle	Serial No from Appendix E
25	0	BITSET8	1	Vehicle can remotely control the drive of other tractive units with control system 3 over the train bus	X		48
	1		1	Drive of the electric tractive unit with control system 1e can be remote-controlled over the train bus	X		49
	2		1	Drive of the diesel tractive unit can be remote-controlled with control system 1d over the train bus	X		50
	3		1	Drive of the electric tractive unit can be remote-controlled with control system 2e over the train bus	X		51
	4		1	Drive of the diesel tractive unit can be remote-controlled with control system 2d over the train bus	X		52
	5		1	Drive of the tractive unit can be remote-controlled with control system 3 over the train bus	X		53
	6		1	Vehicle can remotely control the drive of other (tractive) vehicles but not over the train bus	X		54
	7		1	Drive of the (tractive) vehicle can be remote-controlled but not over the train bus	X		55
26	0	BITSET8	1	Vehicle has a speed controller for traction	X		56
	1		1	Vehicle has a train line	X		57
	2		1	Vehicle can globally remotely control the train line over the train bus	X		58
	3		1	Vehicle can remotely control the train line over the train bus, selectively	X		59
	4		1	Vehicle can be remote-controlled by the train line over the train bus, selectively	X		60
	5		1	Vehicle supports report "train line in"	X		61
	6		1	Vehicle supports report "train line earthed"	X		62
	7		1	Vehicle supports report "train line externally supplied"	X		63
27	0	BITSET8	1	Vehicle supports remote control of fan over the train bus	X		64
	1		1	Vehicle supports compressor remote control over the train bus	X		65
	2		1	Vehicle supports input of target speed	X		66
	3		1	Vehicle supports fault reset	X		67
	4		1	Vehicle supports "establish traction readiness"	X		68
	5		1	Vehicle supports "sand"!	X		69
	6		1	Vehicle supports "drive or prepare to drive...!"	X		70
	7		1	Vehicle supports "drive through tunnel"	X		71
28	0	BITSET8	1	Vehicle supports over high current limitation	X		72
	1		1	Vehicle supports "drive through a neutral section"	X		73
	2		1	Vehicle supports start train power supply or switch on/switch off or cut out	X		74
	3		1	Vehicle supports preheat coolant	X		75
	4		1	Vehicle supports transmission gear change	X		76
	5		1	Vehicle supports fast brake command	X		77
	6		1	Vehicle supports control of the Mg-brake	X		78
	7		1	Vehicle supports release of the WB-brake	X		79

Octet	Bit	Data type / Scope of values	Code	Meaning	Source Gateway (=Trainset)	Source vehicle	Serial No from Appendix E
29	0	BITSET8	1	Vehicle supports control of the WB-brake	X		80
	1		1	Vehicle supports tilting technology	X		81
	2		1	Vehicle supports report of high current	X		82
	3		1	Vehicle supports report of the overhead line voltage	X		83
	4		1	Vehicle supports report of the train power supply	X		84
	5		1	Vehicle supports report of preheating operating	X		85
	6		1	Vehicle supports report of transmission gear change	X		86
	7		1	Vehicle supports report of diesel engine speed	X		87
30	0	BITSET8	1	Vehicle supports auxilliary control of compressor	X		88
	1		1	Vehicle supports report of max. possible tractive effort value	X		89
	2		1	Vehicle supports report of max. possible brake force value	X		90
	3		1	Vehicle supports report of actual traction value	X		91
	4		1	Vehicle supports ep-brake over the train bus with control system1	X		92
	5		1	Vehicle supports ep-brake over the train bus with control system 2	X		93
	6		1	Vehicle has ep-brakes but they are not controlled over the train bus	X		94
	7		1	Vehicle supports emergency brake shorting over the train bus	X		95
31	0	BITSET8	1	Vehicle has emergency brake shorting but not over the train bus	X		96
	1		1	Vehicle has magnetic rail brakes	X		97
	2		1	Vehicle supports magnetic rail brakes over the train bus	X		98
	3		1	Vehicle has motor brakes	X		99
	4		1	Vehicle supports motor brakes over the train bus	X		100
	5		1	Vehicle has eddy current brakes	X		101
	6		1	Vehicle supports eddy current brakes over the train bus	X		102
	7		1	Vehicle supports automatic brake test	X		103
32	0	BITSET8	1	Tractive unit reports: motor brake is in service	X		104
	1		1	Vehicle supports tail light operation	X		106
	2		1	Vehicle supports tail light monitoring	X		107
	3		1	Vehicle supports check of automatic coupling engagement	X		110
	4		1	Vehicle supports control of air conditioning system(s)	X		111
	5		1	Vehicle supports diagnostics: flashing defect indicator light and acknowledgement	X		112
	6		1	Vehicle supports diagnostics: transmission of diagnostic results	X		113
	7		1	Vehicle supports diagnostics: all fault reports to the leading vehicle	X		114

Octet	Bit	Data type / Scope of values	Code	Meaning	Source Gateway (=Trainset)	Source vehicle	Serial No from Appendix E
33	0	BITSET8	1	Vehicle supports diagnostics: individual fault reports to the leading vehicle	X		115
	1		1	Vehicle supports diagnostics: sum of fault reports	X		116
	2		1	Vehicle supports electronic train running indicator	X		117
	3		1	Vehicle supports "next station"	X		119
	4		1	Vehicle supports "train connection at next station"	X		120
	5		1	Vehicle supports transmission of advertising	X		121
	6		1	Vehicle supports "stop requested"	X		123
	7		1	Vehicle supports data refreshing in ticket canceller	X		124
34	0	BITSET8	1	Vehicle supports "energy saving" (generic order)	X		126
	1		1	Vehicle supports "data channel"	X		128
	2		1	Vehicle supports report date and time	X		131
	3		1	Vehicle has radio clock	X		132
	4		1	Vehicle supports "run through washer"	X		143
	5		1	Vehicle can remotely control the drive of other tractive units with control system 4 over the train bus.	X		144
	6		1	Drive of the tractive vehicle with control system type 4 can be remotely controlled over the train bus	X		145
	7		1	Vehicle supports report of traction resources	X		146
35	0	BITSET8	1	Vehicle supports report of additional informations	X		147
	1		1	Vehicle supports parking mode	X		148
	2		1	Vehicle supports diagnostics: report of detailed faults to the leading vehicle	X		149
	3		1	Vehicle supports diagnostics: fault correction procedure	X		150
	4		1	Vehicle supports diagnostics: tests	X		151
	5		1	Vehicle supports safe data transmission	X		152
	6-7		0	Reserve for gateway specific properties	X		
36 to 37		BITSET8	0	Reserve for gateway specific properties	X		
38 to 41		UNSIGNED32	0	32-bit CRC Polynomial: G32(x) = x32 + x31 + x30 + x29 + x28 + x26 + x23 + x21 + x19 + x18 + x15 + x14 + x13 + x12+ x11 + x9 + x8 + x4 + x1 + 1 = 1F4ACFB13 Start Value = 0xFFFFFFFF Fill Value = 0xFFFFFFFF	X		
42 to 43		BITSET8	0	Reserve for national gateway specific properties	X		
44			0	General reserve	X		

Octet	Bit	Data type / Scope of values	Code	Meaning	Source Gateway (=Trainset)	Source vehicle	Serial No from Appendix E
Vehicle based properties – 1st octet series = 1st vehicle in trainset (in reference direction front of trainset)							
45		Array of UNSIGNED8	binary	UIC Identification number of the vehicle		X	
46							
47							
48							
49							
50	0	BITSET8	1	Vehicle has 1st class seats		X	1
	1		1	Vehicle has 2nd class seats		X	2
	2		1	Vehicle has seats for smokers		X	3
	3		1	Vehicle has seats for non-smokers		X	4
	4		1	Vehicle has equipment for disabled people		X	5
	5		1	Vehicle has a compartment for mother and child		X	6
	6		1	Vehicle has a conference compartment		X	7
	7		1	Vehicle has a guard's compartment		X	8
51	0	BITSET8	1	Vehicle is restaurant car or has seats for meals		X	9
	1		1	Vehicle has a support point for a minibar		X	10
	2		1	Vehicle has a support point for catering		X	11
	3		1	Vehicle is a couchette coach or has couchette places		X	12
	4		1	Vehicle is a sleeping car		X	13
	5		1	Vehicle is a special coach (e.g. company coach)		X	14
	6		1	Vehicle is a baggage van or has space for carrying baggage		X	15
	7		1	Vehicle is a postal vehicle or has space for carrying mail		X	16
52	0	BITSET8	1	Vehicle has a telephone for passenger use		X	18
	1		1	Vehicle is a freight wagon		X	20
	2		1	Vehicle has moveable footsteps		X	25
	3		1	Vehicle supports release of the footsteps		X	26
	4		1	Vehicle supports locking of the doors of sleeping cars		X	27
	5		1	Vehicle supports common operation of the corridor connection doors to the neighbouring vehicle		X	28
	6		1	Vehicle has a cab for one direction of travel		X	42
	7		1	Vehicle has two cabs for both directions of travel		X	43
53	0	BITSET8	1	Vehicle supports completeness of the train		X	105
	1		1	Vehicle has an automatic coupling on vehicle No. 1 end		X	108
	2		1	Vehicle has an automatic coupling on vehicle No. 2 end		X	109
	3		1	Vehicle supports electronic seat reservation		X	118
	4		1	Vehicle supports train crew call		X	122
	5		1	Vehicle supports FIS-exchange		X	125
	6		1	Vehicle supports "energy saving" (vehicle selective)		X	127
	7		1	Vehicle supports group addressing		X	129

Octet	Bit	Data type / Scope of values	Code	Meaning	Source Gateway (=Trainset)	Source vehicle	Serial No from Appendix E
54	0	BITSET8	1	Vehicle supports report of actual speed value		X	130
	1		1	Vehicle has a train bus node		X	141
	2-7		0	Reserve for special vehicle properties		X	
55		BITSET8	0	Reserve for national vehicle specific properties		X	
56			0	General vehicle specific reserve		X	
57 - 58		UNSIGNED16	NNN 0	Seat reservation number of the vehicle No seat reservation number given		X	513
Vehicle based properties – 2nd octet series = 2nd vehicle in trainset (in reference direction front of trainset)							
59 bis 72				As octet 45 - 58		x	
Vehicle based properties – 3rd octet series = 3rd vehicle in trainset (in reference direction front of trainset))							
73 bis 86				As octet 45 - 58		x	
Vehicle based properties – 4th octet series = 4th vehicle in trainset (in reference direction front of trainset))							
87 bis 100				As octet 45 - 58		x	
Vehicle based properties – 5th octet series = 5th vehicle in trainset (in reference direction front of trainset))							
101 bis 114				As octet 45 - 58		x	
Vehicle based properties – 6th octet series = 6th vehicle in trainset (in reference direction front of trainset))							
115 bis 128				As octet 45 - 58		x	

Record of updates:

Version	Date	Change	Reason for the change
002.01	01.11.2004	Admission of 9 new gateway specific static vehicle properties in the inauguration frame (properties 143-151)	Inclusion of the new functions concerning *UIC Leaflet 647*
		Change of the layout according to M1	New layout based on the UIC guide M1
		Adoption of a history of updates	Increasing the usability
		Adoption of a revision number	Enhancement and redesign of the versioning
002.02	01.08.2005	Change of the layout	UIC requirement
		Modification of the numbering of the Appendices	
002.03	01.03.2009	Insertion of the CRC32 and the new vehicle property according to appendix L - page 224	New appendix L

Appendix D - Process Data Marshalling (PDM) - Specification (Version 002.02, valid from 01.08.2005)

D.1 - Introduction

The International Electrotechnical Commission, Technical Committee 9, Working Group 22 (*IEC TC9 WG22*), in collaboration with the International Union of Railways, Pilot Group 5R (UIC 5R), has developed a set of standards defining a Train Communication Network (TCN).

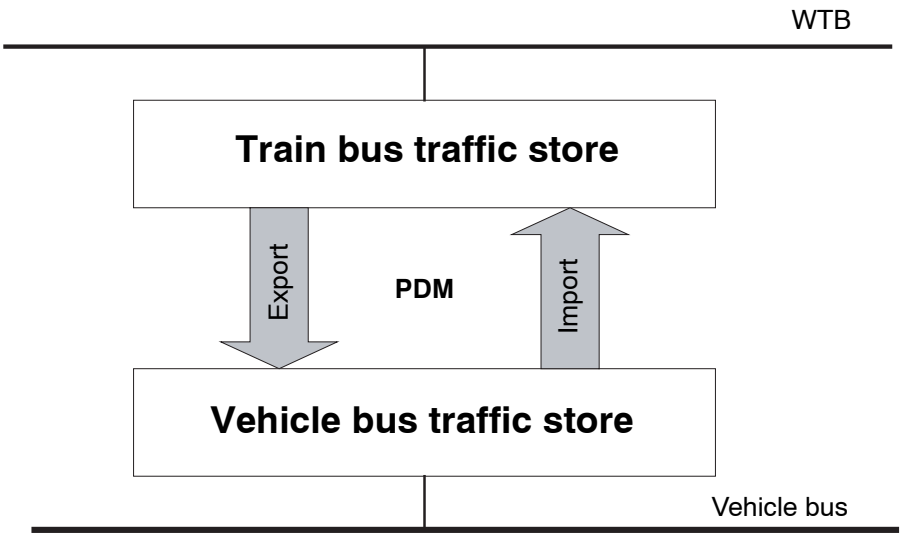
This network consists of a train bus which interconnects the different vehicles of a train and of a vehicle bus which interconnects the equipment within a vehicle.

This document describes the functionality of Process Data Marshalling PDM. In principle, PDM transports process variables from TCN Wire Train Bus WTB to a vehicle bus (e.g. TCN Multifunction Vehicle Bus MVB) and vice versa. PDM handles periodic data only, message data are not handled.

The Process Data Marshalling PDM is the application task to transfer process variables over bus boundaries. The PDM can be customized to a specific application by lists. These lists define where process variables come from and where they go to. PDM Function can do some processing on the process variables.

D.2 - Marshalling types

PDM copies process variables from one Traffic Store to another Traffic Store:



Two types of marshalling are defined:

- Export Marshalling
- Import Marshalling

D.2.1 - Export Marshalling

Export Marshalling means, to copy variables from one or more Vehicle Bus Traffic Stores to the WTB Traffic Store source port. The entire WTB port is written, so unused space in the port has to be filled with default values.

The Export Marshalling can do some processing on the process variables. The Export Marshalling determines the length of the exported frame depending on the frame type.

D.2.2 - Import Marshalling

Import Marshalling means, to copy variables from the WTB Traffic Store to the statically configured Vehicle Bus Traffic Store(s). The Import Marshalling can do some processing on the process variables.

D.3 - Marshalling Modes

A vehicle may have different dynamic operation modes. According to these vehicle operation modes, PDM offers marshalling modes. Each marshalling mode may have a different configuration.

Example:
A traction vehicle like a locomotive may be the traction leader of a train set, traction follower or support no traction at all. In each case different data will be imported and exported.

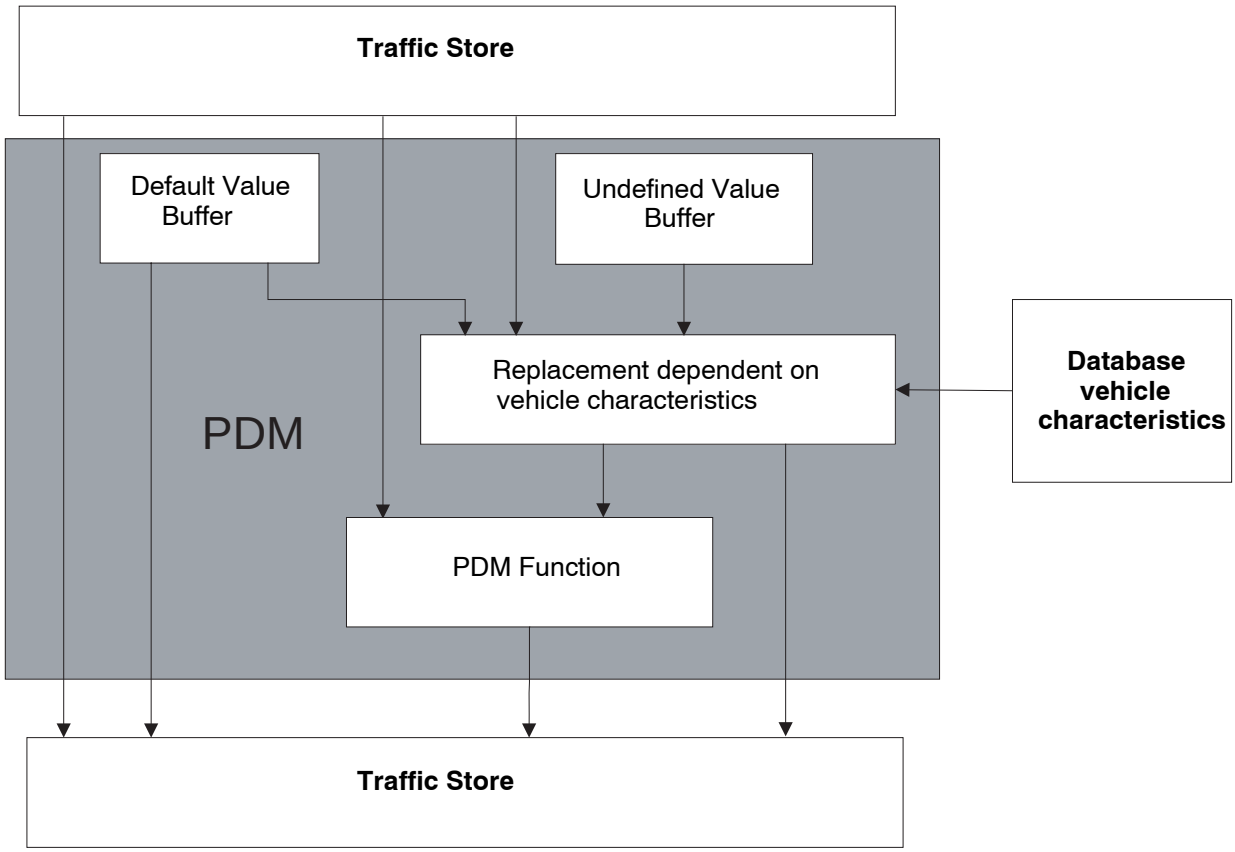
A default mode should always be present, if no specific mode is used for PDM.

If the vehicle operation mode changes, PDM can be reconfigured to accept the new marshalling mode.

D.4 - Data paths in PDM

This point is for informational purposes only and not normative as each gateway supplier may implement their gateways differently.

PDM marshalls the process data from a source to a destination. The destination is always a Traffic Store. The source is a Traffic Store, a Default Value Buffer or an Undefined Value Buffer.



The following data paths can be configured:

- from Traffic Store to Traffic Store, direct or via a PDM Function.
  This is the basic task of PDM. The process variable can be processed by a function before it is written to the destination Traffic Store.

- from Default Value Buffer to Traffic Store.
  Default values are used when a value should be written to a port, but no process variable can deliver this value. Default values are not intended to substitute invalid or too old (not fresh) process variables.

- Vehicle feature dependent process variable from Traffic Store to Traffic Store, direct or via a PDM Function.
  Process variables may be combined with static and dynamic vehicle features. Marshalling will only be done, when the vehicle feature is present. Otherwise PDM substitutes the process variable by an undefined or application defined default value of the apropriate data type. An undefined value is represented by a check variable set to 11_b or the variable value set to all "1". This conforms to the *TCN-Norm IEC 61375*.

Three cases should be considered for vehicle dependent process variables:

- One variable with one check variable: PDM can substitute an undefined value by setting the check variable to 11_b.
- Several variables with one check variable: If all variables are not supported, PDM can substitute undefined values by setting the check variable to 11_b.
- Several variables with one check variable: If only some variables are not supported, PDM should substitute application defined default values.

The process variable can be processed by a function before it is written to the destination Traffic Store. If the process variable is substituted by an undefined value, it can be ignored as function argument.

D.5 - PDM Operation

PDM is activated by a configurable timer or an event (e.g. WTB data receive). After activation, PDM copies all variables configured for the activated marshalling type.

The copy process has four steps:

1. All variables are read, one set of data after the other.
2. For each variable with configured freshness supervision the freshness is checked. Too old variables are invalidated (see below).
3. Then all configured functions (see point D.6 - page 96) are applied to the variables.
4. At last PDM copies the variables, function results and default values to the destination ports and Traffic Stores.

for all datasets from which Variables come				
	read all variables of dataset			
	if (Import Marshalling *and* Frame Type OK) *or no* Import Marshalling			
	then			*else*
	for all variables of dataset			set all variables of dataset invalid
		if check freshness, data too old		
		then set variable invalid	*else*	
if Import Marshalling *and* any frame type field not OK				
then set all variables of WTB Traffic Store invalid				*else*
for all PDM Functions				
	execute PDM Function			
for all datasets to which variables go to				
	write all variables of dataset (PD Variables, Function results, Default Values)			

A variable or a function is invalidated by the following algorithm:

if Variable or function result has check	
then set check to 00b	*else* set variable value to all "0"

If the variable or function result has a check variable, only the check variable is set to 00_b . The variable value is not set to all "0" since the variable can not be very invalid.

If the variable or function result has no check variable the value will be set to all "0". This conforms to the *TCN Standard IEC 61375*.

NB : Since overwriting the value of a Process_Variable with all "0" or "1" may yield a legal value, a Check_Variable of the same dataset is used as a validity indicator where this could be a problem (*IEC 61375, ch. 2.2.1.2.4 Error handling*).

D.6 - PDM Function

D.6.1 - General

Additionally to the pure copying of variables, PDM supports the processing of process variables by functions. The function processing is supported for all Marshalling types.

Example:
Suppose, an application needs to know whether all doors of a train are closed. Since a train may consist of 1 to 20 vehicles with doors, the application must be able to process a wide range of input data. Using a function of the PDM which reads all door states and delivers one variable saying "all doors closed" makes application programming much easier.

The functions offered by PDM have generally the form:

$y = f(x_1, x_2, \ldots x_n); x_i, i = 1 .. n$, input argument, y function result

There may be any number (greater than zero) of arguments and one result. Arguments of a function may come from different ports and Traffic Stores. Arguments and the result are described by PV_Names.

The PDM offers the following standard processing functions:

- logical functions: AND, AND_IGNORE_INVALID
  OR, OR_IGNORE_INVALID
  XOR, XOR_IGNORE_INVALID
- numeric functions: MIN, MIN_IGNORE_INVALID
  MAX, MAX_IGNORE_INVALID
  SUM, SUM_IGNORE_INVALID

D.6.2 - Function processing

All arguments of a function are checked for validity. Too old variables are already invalidated by the second step of the copy process. Arguments can have a check variable or not. Both variants can be mixed if the function has more than one argument.

If an argument is invalid or undefined, it can be ignored (functions XXX_IGNORE_INVALID). If an invalid or undefined argument is not ignored, it sets the function result to invalid.

Only valid arguments are processed. If all arguments are invalid or undefined, the result is set to invalid.

If necessary, the type of the arguments is converted to a type suitable for processing. The processing type is configurable.

If an error occurs during the evaluation of the function the result is set to invalid.

After all arguments are processed, the computed result is converted to the desired function result described by a PV_Name.

The function result can have a check variable or not, independent of the arguments.

for all arguments of function			
	if variable is valid		
	then cast arguments type to computing type	*else* *if* IGNORE_INVALID	
	apply function to argument and compute function result	*then* ignore argument	else set function result invalid *return* function result
if all variables are invalid			
then set function result invalid		*else*	
return function result			
cast computing type to result type			
if function result has check			
then *set* check to 01b			*else*
return function result			

The validity of a variable is checked by the following algorithm:

if Variable has check			
then		*else*	
if check = 10b *or* check = 01b		*if* Variable value has all "0" or "1"	
then Variable is valid	*else* Variable is invalid	*then* Variable is invalid	*else* Variable is valid

D.6.3 - Logical functions: AND, OR and XOR

For these functions the arguments are of BOOLEAN type, ANTIVALENT type or of BITSET type. If the argument is of BITSET type, it is necessary to specify also the bit position of the bit within the bit-set. The types of arguments may be mixed within one function call.

The result value of this functions is of BOOLEAN or ANTIVALENT type.

Additionally, it is possible to specify whether the variable is used directly or whether the variable is negated before use.

D.6.4 - Numeric functions: MIN, MAX, SUM

For these functions the arguments are of INTEGER, UNSIGNED, REAL and FRACTIONAL type with a maximal size of 32 bits. INTEGER and UNSIGNED with different sizes may be mixed within one function call. After processing the type of the result is converted (by a cast) to the type of the destination variable. There is no range checking, be careful with overflows.

Record of updates:

Version	Date	Change	Reason for the change
002.01	01.11.2004	Change in the layout according to M1	New layout in accordance with UIC guide M1
		Adoption of a record of updates	Increasing the usability
		Adoption of a revision number	Enhancement and redesign of the versioning
002.02	01.08.2005	Change in the layout	UIC requirement
		Renumbering of the Appendices	

Appendix E - Vehicle properties and collective addresses

E.1 - List of static vehicle properties (Version 001.03, valid from 01.03.2009)

The list of static vehicle properties is to be found on the UIC-Website (http://www.uic.org) Technology and Research/Products/UIC leaflets/Appendices/556.

E.2 - List of dynamic vehicle properties (Version 001.02, valid from 01.08.2005)

The list of dynamic vehicle properties is to be found on the UIC-Website (http://www.uic.org) Technology and Research/Products/UIC leaflets/Appendices/556.

E.3 - List of collective addresses (Version 001.02, valid from 01.08.2005)

The list of collective addresses is to be found on the UIC-Website (http://www.uic.org) Technology and Research/Products/UIC leaflets/Appendices/556.

Appendix F - TCN Data Types (Version 001.02, valid from 01.08.2005)

F.1 - Presentation and encoding of transmitted data

The Appendix F specifies data types and encoding rules to specify how these types are encoded for transmission. It serves as an informal, first reference; more detailed information can be found in the TCN standard.

F.2 - Data ordering

F.2.1 - Transmission format

The TCN standard prescribes the order in which bits and words are transmitted over the Train Communication Network. To this effect, it defines a number of primitive types and structured types.

F.2.2 - General rules

- Information about the data type shall not be sent with the data. Types are expected to be defined and agreed beforehand between the users of the TCN in a specific application.
- All data shall be transmitted most significant octet first (Big-Endian). On the WTB, each octet is transmitted with the least significant bit first.
- The elements of a structured type (e.g. Record) shall be transmitted in the order in which they are declared.
- Arrays shall be transmitted in order of increasing index. Multi-dimensional arrays are transmitted in the order their indices are listed (e.g. ARRAY OF [row, column] is transmitted row by row).
- To ease implementation, a variable shall be transmitted at an offset address which is a multiple of its size (natural alignment).
- Variable length data (open arrays, records, sets...) shall not be used as Process_Variables.
- When the network is unable to supply data, for instance when a transmission error occurs or when the Traffic_Store is not initialised, any instance which detects it shall overwrite the unsafe data with "0".

This notation uses compact encoding rules:

- It assumes that all user-defined types are recognised by the destination.
- It uses primitive types of fixed size.
- It includes the size of the elements explicitly in a dedicated field where needed.
- It is not aligned, although natural alignment should be used.

F.3 - Notation for the primitive types

F.3.1 - Notation for the Boolean type

Definition

A primitive type with two distinguished values, TRUE and FALSE.

NB : This type is used to represent binary inputs and outputs (relay, led, micro switch, etc.).

Encoding

A variable of Boolean type shall be encoded as one bit:

1. Bit	Interpretation
$2^{\wedge}0$	
0	FALSE
1	TRUE

F.3.2 - Notation for the Antivalent type

Definition

A primitive type with four distinguished values.

NB : Variables of this type are used as check variable for other variables or for critical Booleans.

Encoding

A variable of Antivalent type shall be transmitted as 2 bits, the first one corresponds to the Boolean meaning of the variable and the second to its inverse. It may take one of four states, as follows:

1. Bit	2. Bit	Interpretation
$2^{\wedge}1$	$2^{\wedge}0$	
0	0	ERROR
0	1	FALSE
1	0	TRUE
1	1	UNDEFINED

NB : the ERROR and UNDEFINED states may be interpreted as legal states by an application.

F.3.3 - Notation for the unsigned integer types (UNSIGNED#)

Definition

A primitive type with distinguished values which are positive whole numbers, including zero (as a single value), having a fixed size in bits defined by the suffix #.

Encoding

UNSIGNED8 Type

Offset	0	1	2	3	4	5	6	7
	$2^{\wedge}7$	$2^{\wedge}6$	$2^{\wedge}5$	$2^{\wedge}4$	$2^{\wedge}3$	$2^{\wedge}2$	$2^{\wedge}1$	$2^{\wedge}0$

Range: $0..2^{\wedge}8-1$

UNSIGNED16 Type

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	$2^{\wedge}15$	$2^{\wedge}14$	$2^{\wedge}13$	$2^{\wedge}12$	$2^{\wedge}11$	$2^{\wedge}10$	$2^{\wedge}9$	$2^{\wedge}8$	$2^{\wedge}7$	$2^{\wedge}6$	$2^{\wedge}5$	$2^{\wedge}4$	$2^{\wedge}3$	$2^{\wedge}2$	$2^{\wedge}1$	$2^{\wedge}0$

Range: $0..2^{\wedge}16-1$

UNSIGNED32 Type

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	$2^{\wedge}31$	$2^{\wedge}30$	$2^{\wedge}29$	$2^{\wedge}28$	$2^{\wedge}27$	$2^{\wedge}26$	$2^{\wedge}25$	$2^{\wedge}24$	$2^{\wedge}23$	$2^{\wedge}22$	$2^{\wedge}21$	$2^{\wedge}20$	$2^{\wedge}19$	$2^{\wedge}18$	$2^{\wedge}17$	$2^{\wedge}16$
	$2^{\wedge}15$	$2^{\wedge}14$	$2^{\wedge}13$	$2^{\wedge}12$	$2^{\wedge}11$	$2^{\wedge}10$	$2^{\wedge}9$	$2^{\wedge}8$	$2^{\wedge}7$	$2^{\wedge}6$	$2^{\wedge}5$	$2^{\wedge}4$	$2^{\wedge}3$	$2^{\wedge}2$	$2^{\wedge}1$	$2^{\wedge}0$
Offset	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Range: $0..2^{\wedge}32-1$

F.3.4 - Notation for the integer type (INTEGER#)

Definition

A primitive type with distinguished values which are positive and negative whole numbers, including zero (as a single value), having a fixed size in bits defined by the suffix #.

Encoding

The value shall be represented in binary 2's complement.

When the carried value has a smaller size than the INTEGER# type, it shall be right-justified and sign-extended to the left (if it is negative, with "1", otherwise, with "0").

INTEGER8 encoding

Offset	0	1	2	3	4	5	6	7
	sign	$2^{\wedge 6}$	$2^{\wedge 5}$	$2^{\wedge 4}$	$2^{\wedge 3}$	$2^{\wedge 2}$	$2^{\wedge 1}$	$2^{\wedge 0}$

Range: $-2^{\wedge 7}..2^{\wedge 7}-1$

Example:

'1111 1110'B = -2

INTEGER16 encoding

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	sign	$2^{\wedge 14}$	$2^{\wedge 13}$	$2^{\wedge 12}$	$2^{\wedge 11}$	$2^{\wedge 10}$	$2^{\wedge 9}$	$2^{\wedge 8}$	$2^{\wedge 7}$	$2^{\wedge 6}$	$2^{\wedge 5}$	$2^{\wedge 4}$	$2^{\wedge 3}$	$2^{\wedge 2}$	$2^{\wedge 1}$	$2^{\wedge 0}$

Range: $-2^{\wedge 15}..2^{\wedge 15}-1$

INTEGER32 encoding

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	sign	$2^{\wedge 30}$	$2^{\wedge 29}$	$2^{\wedge 28}$	$2^{\wedge 27}$	$2^{\wedge 26}$	$2^{\wedge 25}$	$2^{\wedge 24}$	$2^{\wedge 23}$	$2^{\wedge 22}$	$2^{\wedge 21}$	$2^{\wedge 20}$	$2^{\wedge 19}$	$2^{\wedge 18}$	$2^{\wedge 17}$	$2^{\wedge 16}$
	$2^{\wedge 15}$	$2^{\wedge 14}$	$2^{\wedge 13}$	$2^{\wedge 12}$	$2^{\wedge 11}$	$2^{\wedge 10}$	$2^{\wedge 9}$	$2^{\wedge 8}$	$2^{\wedge 7}$	$2^{\wedge 6}$	$2^{\wedge 5}$	$2^{\wedge 4}$	$2^{\wedge 3}$	$2^{\wedge 2}$	$2^{\wedge 1}$	$2^{\wedge 0}$
Offset	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Range: $-2^{\wedge 31}..+2^{\wedge 31}-1$

F.3.5 - Notation for the enumerated type (ENUM#)

Definition

A primitive type whose values are given distinct identifiers as part of the type notation. It has a fixed size in bits defined by the suffix #.

Example:

```
Day_Of_WeeK_Typ::= ENUM4
        {
        MONDAY                  (1),
        TUESDAY                 (2),
        WEDNESDAY               (3),
        THURSDAY                (4),
        FRIDAY                  (5),
        SATURDAY                (6),
        SUNDAY                  (7),
        UNDEFINED               (0)
        }
```

Value "2" means "TUESDAY".

Encoding

Values of ENUM# shall be represented by an unsigned integer occupying the same place.

ENUM4 encoding

Offset	0	1	2	3
	$2^{\wedge 3}$	$2^{\wedge 2}$	$2^{\wedge 1}$	$2^{\wedge 0}$

Range: 0..15

Example:

'0001'B means "MONDAY" in the above example.

ENUM8 encoding

Offset	0	1	2	3	4	5	6	7
	$2^{\wedge 7}$	$2^{\wedge 6}$	$2^{\wedge 5}$	$2^{\wedge 4}$	$2^{\wedge 3}$	$2^{\wedge 2}$	$2^{\wedge 1}$	$2^{\wedge 0}$

Range: 0..255

Example:

'0000 0001'B means "MONDAY" in the above example (considering it is ENUM8 rather than ENUM4 .

F.3.6 - Notation for the unipolar types

Definition

Primitive types with distinguished values which are non-negative, whole numbers divided by a fixed power of two, expressing a value in percent of a span.

NB:
1 - The number before the comma gives the number of power of 2 forming the integer part.
2 - The epsilon factor is equal to the value of the smallest power of two in the word.

UNIPOLAR2.16 encoding

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	$2^{\wedge 1}$	$2^{\wedge 0}$	$2^{\wedge -1}$	$2^{\wedge -2}$	$2^{\wedge -3}$	$2^{\wedge -4}$	$2^{\wedge -5}$	$2^{\wedge -6}$	$2^{\wedge -7}$	$2^{\wedge -8}$	$2^{\wedge -9}$	$2^{\wedge -10}$	$2^{\wedge -11}$	$2^{\wedge -12}$	$2^{\wedge -13}$	$2^{\wedge -14}$
	integrer part		fractional part													

Span: 0..400% - Epsilon

F.3.7 - Notation for the bipolar types

Definition

Primitive types with distinguished values which are positive or negative, whole numbers (including zero) divided by a fixed power of two, expressing a value in percent of a span.

NB: 1 - The number before the comma gives the span in multiple of 100%.
2 - The epsilon factor is equal to the value of the smallest power of two in the word.

Encoding

BIPOLAR2.16 Encoding

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	sign	$2^{\wedge 0}$	$2^{\wedge -1}$	$2_{\wedge -2}$	$2^{\wedge -3}$	$2^{\wedge -4}$	$2^{\wedge -5}$	$2^{\wedge -6}$	$2^{\wedge -7}$	$2^{\wedge -8}$	$2^{\wedge -9}$	$2^{\wedge -10}$	$2^{\wedge -11}$	$2^{\wedge -12}$	$2^{\wedge -13}$	$2^{\wedge -14}$
	integrer part	fractional part														

Span: -200%..+200% - Epsilon

BIPOLAR4.16 Encoding

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	sign	$2^{\wedge 2}$	$2^{\wedge 1}$	$2^{\wedge 0}$	$2^{\wedge -1}$	$2^{\wedge -2}$	$2^{\wedge -3}$	$2^{\wedge -4}$	$2^{\wedge -5}$	$2^{\wedge -6}$	$2^{\wedge -7}$	$2^{\wedge -8}$	$2^{\wedge -9}$	$2^{\wedge -10}$	$2^{\wedge -11}$	$2^{\wedge -12}$
	integrer part				fractional part											

Span: -800%..+800% - Epsilon

F.3.8 - Notation for the character type

Definition

A primitive type whose distinguished values are members of the set of characters defined in *ISO/IEC 8859-1* (see Bibliography - page 236).

Encoding

Visible characters shall be transmitted in one octet, without parity bit.

Offset	0	1	2	3	4	5	6	7
	$2^{\wedge 7}$	$2^{\wedge 6}$	$2^{\wedge 5}$	$2^{\wedge 4}$	$2^{\wedge 3}$	$2^{\wedge 2}$	$2^{\wedge 1}$	$2^{\wedge 0}$

Example:

'01100001'B = character "a" according to *ISO/IEC 8859-1*.

F.3.9 - Notation for the Unicode character type

Definition

A primitive type whose distinguished values are members of the set of characters defined in *IEC 10646* (see Bibliography - page 236).

Encoding

Visible characters shall be transmitted in two octets.

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}

F.3.10 - Notation for the uncommitted types

Definition

An uncommitted type of undefined contents, but of fixed size.

Encoding

A variable of uncommitted type has no prescribed encoding.

Bits shall be named according to the power of two of a variable of UNSIGNED# type which would occupy that place.

NB : this naming is not identical to the offset.

WORD8 encoding

Offset	0	1	2	3	4	5	6	7
	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}

WORD16 encoding

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}

F.4 - Structured types

Five different structured types are defined:

RECORD (variable length),
ARRAY (fixed length or variable length),
BITSET# (fixed length),

F.4.1 - Notation for the record types

Definition

A structured type defined by referencing a fixed, ordered list of types; each value of the new type is an ordered list of values, one from each of the component types.

NB : It is recommended to observe natural alignment when defining a RECORD, i.e. all elements should be located at an offset with respect to the beginning of the record which is a multiple of their size.

Encoding

Elements of a RECORD shall be transmitted in the order of their declaration.

Example:

A value of type TimeOfDay32 is represented as follows:

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Year															
	Dummy				Month				Day							

```
TimeOfDay32::= RECORD
        {
        year                    INTEGER16,
        dummy                   WORD4,
        month                   UNSIGNED4,
        day                     UNSIGNED8
        }
```

"Dummy" was introduced to align the variable "day" on a 8-Bit word boundary.

F.4.2 - Notation for the bitset types

Definition

an ARRAY [#] of BOOLEAN1, occupying a fixed size of #.

Encoding

BITSET8 encoding

Offset	0	1	2	3	4	5	6	7
	1st Bit							8th Bit

Range: 8-Bit set of Boolean

Example:

```
AccessType8::= BITSET8
        {
        system          (0),        -- this is the first bit of the set (MSB)
        owner,          (1),
        group,          (2),
        world,          (3),
        reserved4       (4),
        reserved5       (5),
        reserved6       (6),
        reserved7       (7)         -- 8th or last bit of the set (LSB)
        }
```

An UNSIGNED8 occupying that space with a value of '80'H means that "system" is the only member of the set.

BITSET16 encoding

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	1st Bit															last Bit

Example:

AccessType ::= BITSET16 { system (0), owner (1), group (2), world (3)}.

Value '0110 0000 0000 0000'B means that "owner" and "group" are members of the set.

BITSET32 encoding

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Byte 1	1st Bit															
Byte 2																last Bit
Offset	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

BITSET64 encoding

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Byte 0	1st Bit															
Byte 2																
Byte 4																
Byte 6																last Bit

F.4.3 - Notation for the array type

Definition

A structured type, defined by referencing a single existing type; each value of the new type is an ordered list of zero, one or more values of the existing type. The position of each value is identified by an index. The number of values is indicated by either a constant or a field of the embedding structure.

The number or identifier specifies the size of the array in number of elements. Its type shall be UNSIGNED Type. If an UNSIGNED type without a defined identifier is indicated, the array count is located before the array elements begin.

Encoding

Arrays shall be transmitted in order of increasing index.

Multi-dimensional arrays shall be transmitted in the order their indices are listed.

NB : ARRAY OF [row, column] is transmitted row by row.

Arrays of octets (uncommitted contents, e.g. memory dump) shall be transmitted in increasing memory address (or index) of the Application Memory.

All elements of the array shall be transmitted, even those which are not significant.

Example:

transmission of an octet memory dump.

DumpOctetType ::= ARRAY [array_count UNSIGNED16] OF WORD8.

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1st	array_count (before the array structure)															
...	Octet at address M								Octet at address M+1							
..	Octet															
Last	Octet at address (M + array_count-2)								Octet at address (M + array_count-1)							

If the type were:

DumpOctetType ::= ARRAY [count] OF WORD8

the number of elements would be given by the field "count" located in the nesting data structure.

Example:

Same as above, but as a WORD16 memory dump. The array_elements are now half the value than in the preceding case.

DumpWordType ::= ARRAY [array_count UNSIGNED16] OF WORD16.

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
first	array_count (before the array structure)															
...	Word at address M															
..																
Last	Word at address (M + array_count x 2 - 1)															

F.5 - Natural alignment

Natural alignment is an attribute of a data item that refers to its placement:

- a primitive type is "naturally aligned at any position" that is a multiple of its length in octets
- a structure is "naturally aligned" when all members of the structure are naturally aligned.

To achieve this, reserved fields may have to be defined. Such fields have to be initialized with "0".

F.6 - Notation for the TIMEDATE48 type

Definition

A structured type expressing the absolute time in number of seconds since Universal Co-ordinated Time (UTC), 00:00:00, 1st January 1970 (Unix and ANSI-C format).

NB : This type is used for distribution of the actual time, event tagging and synchronisation.

Encoding

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
First	seconds (most significant)															
	seconds (least significant)															
Last	ticks = 1/65536 second															

Time can be represented to a granularity of 15,3 μs (=1/65536 s)

The precision of the fractional part shall be at least 10 bits.

Unused low order bits shall be set to zero.

Record of updates:

Version	Date	Change	Reason for the change
001.01	01.11.2004	Change in the layout according to M1	New layout in accordance with UIC guide M1
		Adoption of a record of updates	Increasing the usability
		Adoption of a revision number	Enhancement and redesign of the versioning
001.02	01.08.2005	Change in the layout	UIC requirement
		Renumbering of the Appendices	

Appendix G - Homologation Procedure for Train Bus Nodes (Version 001.02, valid from 01.03.2009)

G.1 - Foreword

This Appendix describes the formal homologation procedure that shall be applied to homologate train bus nodes whose conformity to this Leaflet has been assessed according to the conformity test described in Appendix H - page 123 .

Practical aspects and responsibilities are taken into account to clarify, simplify and accelerate the homologation process.

G.2 - General prescriptions

The entities involved in the homologation procedure are:

1. UIC System Department;
2. UIC Steering Group Train Bus;
3. Homologated Test Laboratory;
4. Node Manufacturer.

All documents and letters exchanged among the previous entities shall be written in the English language: the electronic transmission is allowed but the documents and letters must also be sent by post.

At the end of this procedure, the complete set of issued documents and the exchanged correspondence shall be provided, in electronic form, on a CD-rom to the UIC System Department and to the UIC Steering Group Train Bus. The files shall be in Acrobat PDF format or others of similar popularity.

G.3 - Procedure

The procedure for the homologation of a train bus node consists of the phases listed hereafter.

1. The manufacturer requests a conformance test at the Test Laboratory of his choice.
2. The conformity test is executed at the test laboratory optionally attended by Subgroup 1b.
3. The Test Laboratory proposes a homologation recommendation to the UIC Steering Group Train Bus after successful execution of the conformance test steps.
4. The Steering Group decides on the homologation certification, writes out the conformance certificate for the manufacturer and forwards the relevant documentation to the UIC System Department.

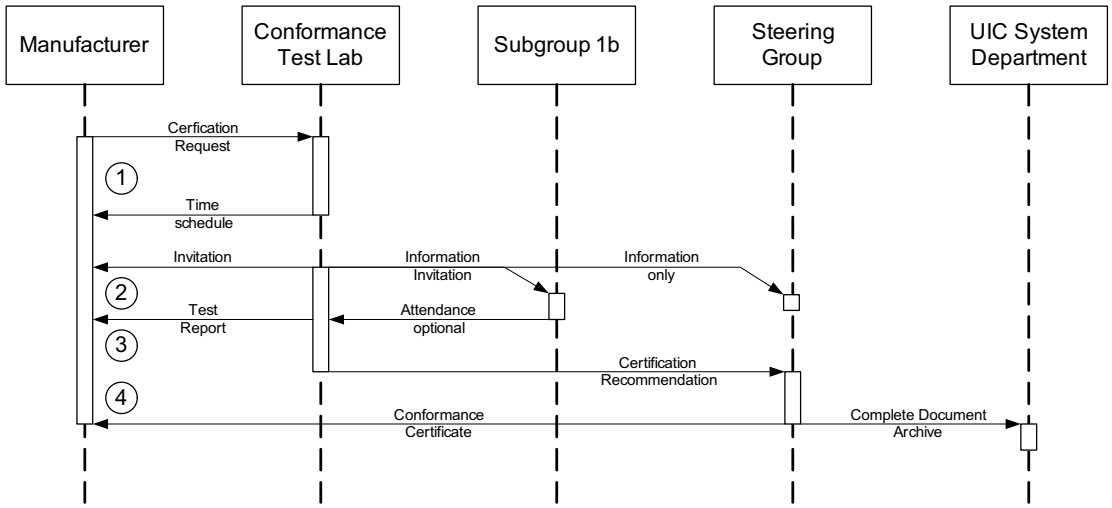


Fig. 11 - Conformance Test Procedure

G.3.1 - Homologation request application to the certified laboratory

The node manufacturer sends to the chosen homologated and certified Test Laboratory all the preparatory documents which are:

1. Details about the company, including a brief presentation of its activities in the railway communication market.
2. A detailed description of the train bus node to homologate.
3. The test reports of the Manufacturer Configuration Conformance Testing according to point H.4.1 - page 130.
4. A document regarding the FIT-values (failure in time) of the train bus node to homologate.
5. Test reports of all other tests carried out on the train bus node by the manufacturer.
6. Documentation regarding the testing of similar products already homologated, if applicable.

The manufacturer has to clarify with the Test Laboratory all issues concerning the infrastructure, logistics and the time schedule including all documentation as requested.

G.3.2 - Conformity test TTC1/TTC2/TTC3

The test laboratory informs and invites the subgroup 1b to the conformance test. The attendance of members of subgroup 1b is not mandatory but optional.

In accordance to point H.4.2 - page 130, personnel of the Test Laboratory shall be present during the tests.

The personnel of the Test Laboratory approves the required documentation and is responsible for the test execution. External WTB experts are not necessary.

The Steering Group shall be informed by the Test Laboratory, but there is no action by the Steering Group.

If the Test Laboratory is located at the manufacturer's site, it is recommended that at least one independent member of Sg1b should attend this step.

G.3.3 - Test report and homologation recommendation

In case of a positive test result, the Test Laboratory forwards a comprehensive test report to the UIC System Department and to the UIC Steering Group Train Bus.

This report shall include detailed information on:

1. the setting up of the laboratory,
2. the results of the tests, including all the detailed outputs of the test bed according to point H.6 - page 132,
3. the minutes of the test session, signed by the participants.

All the documentation produced and the correspondence exchanged must be archived by the Test Laboratory, by the UIC System Department and by the UIC Steering Group Train Bus.

G.3.4 - Homologation

Once the conformity test is passed, the procedure is concluded and the UIC Steering Group Train Bus ratifies the certification of homologation of the node given in point G.3.3 - page 114 and a new edition of Appendix I - page 221 including that node is issued.

G.4 - Test Laboratory Certification

A test laboratory has to install a clear process for the *UIC Leaflet 556* conformance test which is prerequisite for the certification audit.

This chapter gives some guidelines and a proposal for such a process of a test laboratory.

The process should meet some criteria to allow a certification.

- what will be tested (Input)?
- what Test Laboratory equipment is needed (Physical Resources)?
- who will execute the tests (Personnel Resources)?
- how will the tests be executed (Procedure)?
- which test results are expected (Indicators)?
- what is the result of the homologation process (Output)?

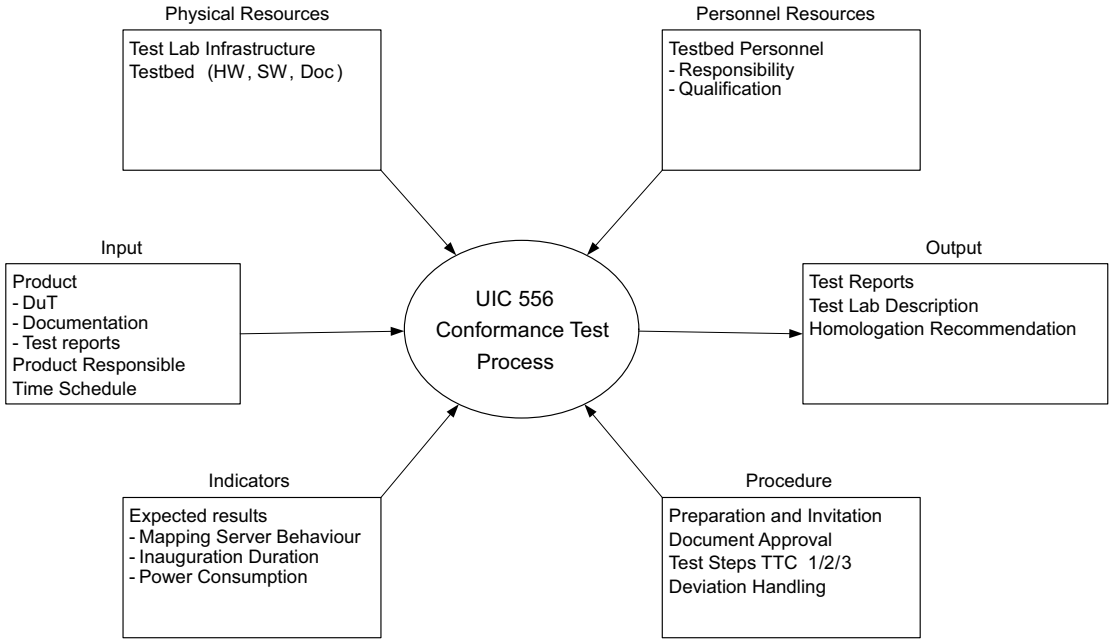


Fig. 12 - Conformance Test Process Definition

G.4.1 - Input

The input of the homologation process should meet the following conditions:

- The Device under Test (DuT), the documentation and the product responsible as required in this Appendix to *UIC Leaflet 556* should be available.
- The DuT product responsible should be available for the conformance test,
- A time schedule has to be agreed between manufacturer and the Test Laboratory,
- The DuT manufacturer should have unrestricted access to Test Laboratory,
- The conditions and charges for the conformance test should be defined,
- The prerequisite normative requirements for the conformance test are described in *UIC Leaflet 556*.

G.4.2 - Physical Resources

The description of the physical resources of the homologation process should meet the following conditions:

- The infrastructure of the Test Laboratory should be described:
  - Name of company or operator of the test lab,
  - Record in trade register (if applicable),
  - Field of activity,
  - Address of Test Laboratory (as part of company or as part of a organisation),
  - Services sector of test lab,
  - Applicability of the conformance test,
  - Independence of test lab.
- The equipment of the Test Laboratory should be described:
  - Hardware (Reference gateways, test bed platform, auxiliary components),
  - Software (Automated execution of tests, evaluation of test results),
  - Configurations (TTC1/2/3).

G.4.3 - Personnel Resources

The description of the personnel resources of the homologation process should meet the following conditions:

- Who is responsible for the test execution and how is this organized?
- Who is responsible to propose the homologation recommendation to the UIC Steering Group Train Bus?
- Who will monitor the conformance test procedure?
- Which competence and qualification has the Test Laboratory personnel?
- How is the independence of the Test Laboratory guaranteed?
- How is the know how preserved?
- Which are the financial resources of the Test Laboratory and who monitors them?
- Who and how to deal with conflicts between manufacturer and Test Laboratory ?
- What relations exist to subgroup 1b.

G.4.4 - Procedure

The description of the procedure of the homologation process should meet the following conditions:

- The preparation and the invitation (whom and when) should be defined.
- The criteria for the document approval should be defined (e.g. check list).
- The execution of the test steps TTC1/2/3 should be defined.
- The handling of deviation to the expected results (e.g. sporadic failures) should be defined.
- The Test Laboratory should have a working Quality Management System which defines:
  - the Quality Management Officer,
  - the rules for the conformance test,
  - the traceability of the conformance test,
  - internal audits.

G.4.5 - Indicators

The description of the indicators of the homologation process should meet the following conditions:

- There should be a reference to Appendix H - page 123 of *UIC Leaflet 556*, which describes the test steps and expected results in detail.

G.4.6 - Output

The description of the output of the homologation process should meet the following conditions:

- The deployed test bed and the test procedure should be documented.
- The test results should be documented.
- A homologation recommendation should be sent to the UIC Steering Group Train Bus.
- The documentation should be managed by the Test Laboratory to cover issues of:
  - availability,
  - actuality (list of documents),
  - history of changes,
  - traceability of the procedure,
  - distribution,
  - confidentiality,
  - archive.

G.4.7 - Process Proposal

The process is described as a series of steps. A *pre-condition* has to be met, before a step starts. The *participants* accomplish the *actions* usíng the *input* to build the *output*. A *post-condition* holds after the step.

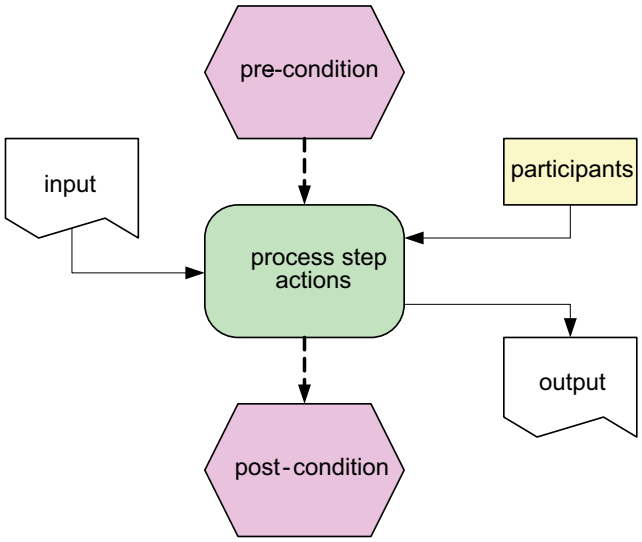


Fig. 13 - Description of a process step

Name	Function
TLP	Test Laboratory personnel operates the Test Laboratory according to *UIC Leaflet 556*
M	Manufacturer of the Device under Test DuT
Sg1b	Subgroup 1b, Conformance test
SGTB	Steering Group Train Bus

The following figure shows an overview of all proposed steps of the conformance test process.

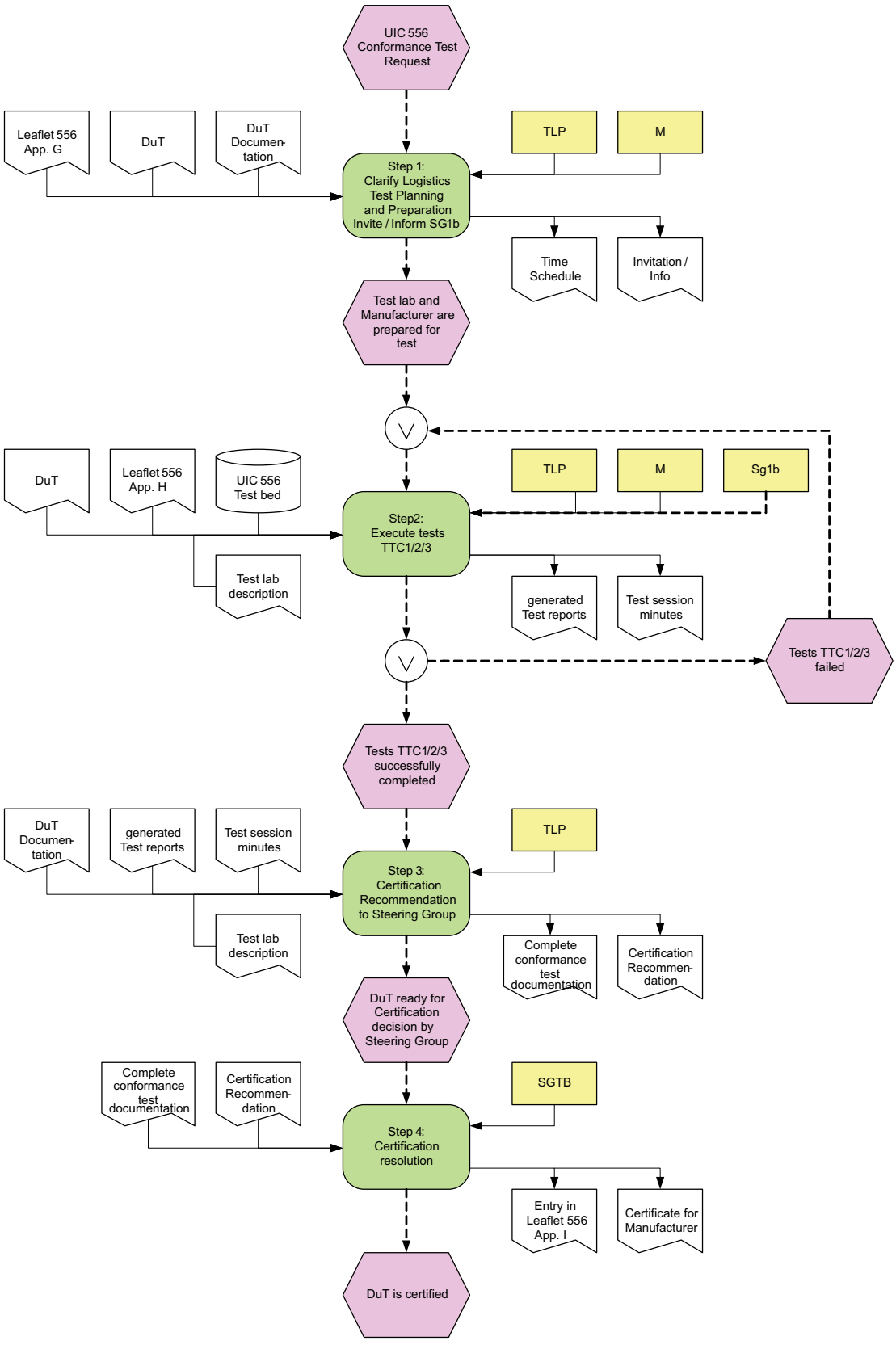


Fig. 14 - Conformance Test Process Steps

The process starts with a request of the manufacturer of the DuT to the Test Laboratory of the manufacturer's choice.

Step	Actions
1	The Test Laboratory personnel (TLP) contacts the manufacturer (M) to clarify the logistics of the conformance test as described in this Appendix to *UIC Leaflet 556*. The TLP makes an offer including the charges for the conformance test to the M. The M has to provide the DuT, documentation about the manufacturers company and the DuT, DuT test reports (according to Appendix H - page 123 and other tests), documentation of the fit values of the DuT and documentation regarding the testing of similar products already homologated, if applicable. The TLP approves the documentation. The TLP and the M agree about the time schedule to execute the conformance tests as described in Appendix H. The TLP and the M prepare the conformance test with the repeated execution of test cases TTC1/TTC2/TTC3 to get enough confidence in conformity. The TLP informs (invites) the Subgroup 1b (Sg1b) and the Steering Group Train Bus (SGTB) about the intended conformance test. The indication contains information about the manufacturer the Test Laboratory and the time schedule. The information (invitation) is sent out at least two weeks before step 2.
2	The TLP prepares the conformance test bed (hardware and software). The M has to provide sufficient number of DuTs for test configurations TTC1/TTC2/TTC3. The TLP and the M build up the test configurations TTC1/TTC2/TTC3 as described in Appendix H. TTC3 may be executed at M's site. The TLP runs the conformance test control program which implements the test cases TTC1/TTC2/TTC3 as described in Appendix H. The TLP saves the generated test reports. The test cases should complete without errors. If an error occurs in a test case, the TLP decides to repeat the failed test case one or more times. If the retries succeed, the TLP decides about the overall success of the test case. In case of permanent failure, the whole phase of step 2 shall be repeated. Unresolved conflicts between TLP and M are escalated to the Sg1b. The TLP logs the execution and results of all test cases in the test session minutes. The session minutes are signed by all participants of this step. One or more members of Sg1b may optionally attend the execution of the test cases. If the Test Laboratory is located at the manufacturer site, it is recommended that at least one independent member of Sg1b should attend this step.
3	The TLP combines the DuT documentation, the Test Laboratory description, the test session minutes and the test reports to a complete conformance test documentation and sends it with a certification recommendation to the STGB. The TLP archives the complete conformance test documentation in a version control system.

Step	Actions
4	The SGTB decides about the certification of the DuT based on the certification recommendation. The SGTB adds an entry in Appendix I - page 221. The SGTB archives and forwards the complete conformance test documentation to the UIC System department. The SGTB writes out the certificate of conformity for the M.

G.4.8 - Certification and Audit of the Test Lab

The certification of the Test Laboratory is done by Subgroup 1b.

The first certification should be done by watching the full conformance test process.

The certification of the Test Laboratory should be refreshed by an audit by Subgroup 1b after 3 years.

G.5 - Contact details

Addresses of the UIC entities involved in the homologation procedure:

G.5.1 - UIC System Department, UIC Steering Group Train Bus

UIC - International Union of Railways System Department
16 rue Jean Rey
F-75015 Paris
Tel. +33 1 44 49 20 20
Fax +33 1 44 49 20 29
Internet: http://www.uic.org

UIC - International Union of Railways Secretary of Steering Group Train Bus
16 rue Jean Rey
F-75015 Paris
Tel. +33 1 44 49 20 20
Fax +33 1 44 49 20 29
Internet: http://www.uic.org

G.5.2 - Homologated and certified Laboratories

The list of the homologated and certified laboratories is to be found on the UIC-Website (http://www.uic.org) Technology and Research/Products/UIC leaflets/Appendices/556.

Record of updates

Version	Date	Change	Reason for the change
001.01	01.08.2005	Change in the layout	UIC requirement
		Renumbering of the Appendices	
001.02	01.03.2009	Major rework for test laboratory certification	define a test process to reduce effort in homologation procedure

Appendix H - Conformance testing (Version 001.02, valid from 01.08.2005)

H.1 - Definitions

The following terms are used in this Appendix with a special meaning which is given hereinafter.

UIC 556 Node

This is, at least one WTB device compliant to *IEC Standard 61375-1.* The vehicle bus interface and the gateway functionality are not mandatory, nevertheless, when existing, they may be helpful to attach the tester device. Note that this definition is quite different from the definition given by *IEC 61375-1* that defines a node as a gateway between WTB and MVB.

Key position

This is a position of the node within the test train composition that is believed to be highly significant for the test coverage.

Reference Node

The Reference Node (ref) is a node that is already homologated according to this Appendix.

Device under Test (DuT)

This is a node that is under test with the aim to be homologated according to this Appendix. The Device under Test (DuT) shall have already passed the Manufacturer Configuration Conformance Testing as stated by the Certification Statement given by the manufacturer.

Ref Position

This is a position of the node within the test train composition that has to be hold by a reference node during the execution of the Reference Configuration Conformance Testing.

DuT Position

This is a position of the node within the test train composition that has to be hold by a node under test during the execution of the Reference Configuration Conformance Testing.

Qualitative Result

The qualitative result of a test case is PASSED or NOT PASSED respectively if the checks do or do not match the contents of such checks reported in the table Test Sequence of the test case.

Quantitative Result

The quantitative result of a test case are values measured according to the request of the test sequence of that test case.

TTC (Testing Train Configuration)

The Testing Train Configuration is an important issue in order to reach the required coverage of the test nad to run the test in a condition that is believed to be close to the condition of the actual exploitation. Three different Testing Train Configuration are used called TTC1, TTC2 TTC3.

Standard Addresses

The result of the computation of an actual full configuration with unknown master node occurs frequently and is therefore called "standard addresses", this sentence may appear in the column called "expected configuration" in the table reporting the test sequence.

H.2 - General

H.2.1 - Scope

A UIC 556 node is an equipment designed according to *IEC Standard 61375-1* providing at least the WTB interface and the ability to run the relevant protocols and the application profile specified by *UIC Leaflet 556;* the MVB interface is not strictly required, nevertheless the term UIC 556 node may indicate also a TCN gateway or any other gateway between WTB and the relevant vehicle bus.

The scope of this document is to provide a UIC Conformance Testing procedure that defines the test conditions and the test steps and gives the requirements of the test bed.

UIC 556 conformance testing provides the means of:

- assuring at a convenient coverage that a UIC node complies with the UIC 556 specification;
- facilitating interoperability between different UIC 556 nodes and between vehicles in which such equipment is installed.

The interoperability issues which are covered include interfaces between vehicles relevant to the train bus and do not include the interfaces between equipment relevant to the vehicle bus.

H.2.2 - Levels of Conformity

Two different levels of conformance testing are distinguished:

- **Manufacturer Configuration Conformance Testing:** it is requested when vehicles in the train are equipped with UIC 556 nodes of the same manufacturer. This level of conformance is a prerequisite to the Reference Configuration Conformance. For this first level, it is the responsibility of UIC 556 node manufacturers or system integrators to actually implement the conformance tests and to develop suitable testers where necessary.
- **Reference Configuration Conformance Testing:** it is requested when trains are equipped with UIC 556 nodes from different manufacturers and must be interoperable in their normal operation. For this level of conformance, it is necessary to qualify a UIC 556 node as a UIC approved node and it is the responsibility of UIC to actually implement the conformance tests and to develop suitable testers.

Complete interoperability requires not only that vehicles or equipment can correctly communicate between each other, but also that they are able to sustain the *UIC Leaflet 556* defined traffic with adequate performance; for this reason both levels of Conformance Testing include some test cases that have the aim to check and validate:

- some UIC 556 special functionalities (like sleep mode)

- some UIC 556 performance characteristics (like inauguration time and node redundancy switch-over).

H.2.3 - Entities

The entities involved in order to run the UIC Conformance Testing are the following:

- **UIC Node:** it may be a node under test or a reference node, both are equipped with the software that implements the TCN Stack and the UIC Stack.

- **TCN Stack Software:** this is the software that implements the protocols according to *IEC Standard 61375-1.*

- **UIC Stack Software:** this is the software that implements the protocols according to *UIC Leaflet 556*.

- **UIC tester:** this is an equipment to be attached to the UIC node in order to stimulate the execution of the test cases and/or to record the test cases results. To perform such a task the UIC tester is equipped with a UIC simple testing application.

- **Simple testing application:** this is the software that implements the protocols according to *IEC Standard 61375-1*.

H.2.4 - Coverage Boundaries and Limitations

The boundaries of the UIC Conformance Testing are the following:

- **Lower Boundary:** the coverage of the UIC Conformance Testing is limited by the coverage of the TCN Conformance Testing; consequently the TCN conformance is not a UIC Conformance Testing issue but it is a pre-requisite which has to be already achieved before the UIC conformance is issued.

- **Upper Boundary:** the coverage of the UIC Conformance Testing is limited to a simple testing application that is provided by the UIC 556 node manufacturer for the purpose of running the UIC Conformance Testing. Such a simple testing application shall be compliant to the test cases as they are defined and described in this document. The conformity of any UIC 556 application is beyond the scope of the UIC 556 Conformance Testing.

H.3 - Train configurations

Different Train Configurations are used to run the UIC Conformance Testing as reported by Table 7; each configuration is dedicated to a set of test cases. The column NRN reports the number of non redundant nodes.

The most used configuration is the TTC1 which, for this reason, is called Reference Configuration.

Table 7 : Testing Train Configurations

Short Form ID	Test Cases	number of vehicles	number of non redundant nodes	number of redundant nodes
TTC1	♦ Confirmed compositions test suite • Confirmation, correction and reset of train compositions • Loss and integration of intermediate nodes • Loss and integration of end nodes for confirmed compositions • Lengthening of confirmed compositions • Coupling of two confirmed compositions • Setting the vehicle number for seat reservation ♦ UIC redundancy test suite • Redundancy switch-over for unconfirmed compositions • Redundancy switch-over for confirmed compositions ♦ Unconfirmed compositions test suite; ♦ Leading vehicle test suite; ♦ Inhibit functionality test suite; ♦ Group telegrams test suite; ♦ E telegram error handling test suite; ♦ Sleep mode test suite	7	6	3
TTC2	♦ Unconfirmed compositions test suite (applied to trainsets of all possible lengths)	22	7	
TTC3	♦ UIC redundancy test suite • Duration of redundancy switch-over ♦ Performance test suite • Duration of UIC Inauguration	22	22	1 [a]
Single DuT	♦ Performance test suite • Power consumption	**not applicable**	1	

a. one node shall be redundant in order to measure the redundancy switch-over time.

H.3.1 - Testing Train Configurations

Fig. 15 shows the Testing Train Configuration TTC1 where the following vehicles are represented:

- vehicle A is a locomotive equipped with a redundant node;
- vehicle B is a coach equipped with a single node;
- vehicle C is a steering coach;
- vehicle D is part of a trainset and it is equipped with a redundant node;
- vehicle E is part of a trainset, no node;
- vehicle F is part of a trainset, no node;
- vehicle G is a steering coach equipped with two nodes, one single node and one redundant node.

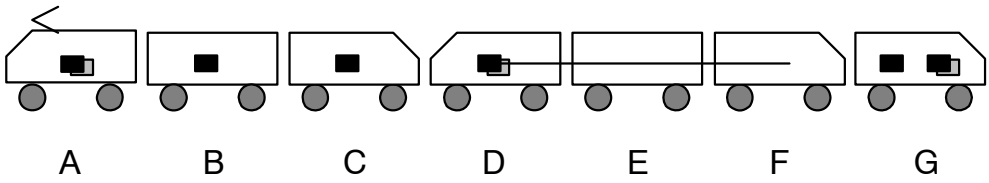


Fig. 15 - Testing Train Configuration TTC1

Table 8 : TTC1 - Identification of key positions, ref positions, DuT positions and properties

Vehicle position	A	B	C	D	E	F	G_1	G_2
Key positions	K		K	K			K	K
Regular test cases								
1st test run	Ref	DuT	Ref	Ref			DuT	DuT
2nd test run	DuT	Ref	DuT	DuT			Ref	Ref
All redundancy tests	K			K			K	K
1st test run	Ref	DuT	Ref	Ref			DuT	DuT
2nd test run	DuT	Ref	DuT	DuT			Ref	Ref
Orientation	←	→	→	←			→	→
Trainset properties	38, 44, 45, 49	---	44, 45	39, 44 45, 50	%	%	---	44, 45
Vehicle properties	42, 141	1, 141	2, 42, 141	1, 42, 141	2	2, 42	2, 141	42, 141

H.3.2 - Testing Train Configuration 2

Fig. 16 shows the Testing Train Configuration TTC2 where the following vehicles are represented:

- vehicle A to E is a trainset equipped with a single node (group of vehicles);
- vehicle F to H is a trainset equipped with a single node;
- vehicle I is a locomotive equipped with a single node;
- vehicle J to K is a trainset equipped with a single node;
- vehicle L to M is a trainset equipped with a single node;
- vehicle P to U is a trainset equipped with a single node;
- vehicle V is a locomotive equipped with a single node;

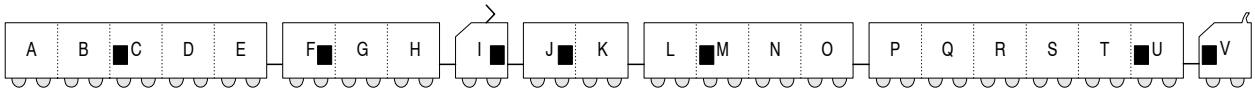


Fig. 16 - Testing Train Configuration TTC2

Table 9 : TTC2 - Identification of key positions, ref positions, DuT positions and properties

Vehicle position	Trainset A/E	Trainset F/H	Vehicle I	Trainset J/K	Trainset L/O	Trainset P/U	Vehicle V
All test cases							
1st test run	DuT	Ref	DuT	Ref	DuT	Ref	DuT
2nd test run	Ref	DuT	Ref	DuT	Ref	DuT	Ref
Vehicle properties	C: 141	F: 141	141	J: 141	L: 42, M: 141, O: 42	U: 42, 141	141
Orientation			←			→	→
Trainset properties			38, 44, 45, 49,				39, 44, 45, 50,

H.3.3 - Testing Train Configuration 3

This Testing Train Configuration is dedicated to the performance measurement.

Although the performance, like inauguration time, is a characteristic that shall be assured in the maximum configuration (it means 22 vehicles with 32 nodes), practical reasons lead to use a reduced configuration that seems, nevertheless, to be significant.

Such configuration contains 22 vehicles and 22 single node.

The configuration shall be implemented in such a way that it is possible to measure the relevant characteristics with shortened configuration of 10 and 16 nodes in order to list a table of the characteristics themselves.

As an example if the inauguration time is measured, the result will be three inauguration times relevant to 10, 16 and 22 nodes.

The value relevant to 22 nodes shall be less than 1.4 second while the values relevant to 10 and 16 nodes are to be considered indicative of the inauguration time versus the number of the nodes in the composition.

Figure 17 shows the Testing Train Configuration TTC3 where the following vehicles are represented:

- vehicle A is a locomotive equipped with a redundant node;
- vehicles B to U are coaches equipped with a single node;

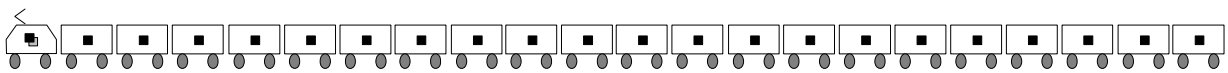


Fig. 17 - Testing Train Configuration TTC3

Table 10 : TTC3 - Identification of key positions, ref positions, DuT Positions and properties

Vehicle position	A	B	C-U
Key positions	K	K	K
Regular test cases	DuT	Ref	DuT
All redundancy tests	DuT	Ref	DuT
Orientation	←	←	←
Trainset properties	38, 44, 45,49		
Vehicle properties	43, 141	141	141

H.4 - Guidelines to conformance testing execution

H.4.1 - Manufacturer Configuration Conformance Testing

The Manufacturer Configuration Conformance Testing is executed by the manufacturer in his own laboratory.

The testing train compositions to be used are TTC1, TTC2 and TTC3.

All test cases shall be executed using DuT produced by the manufacturer. The key positions are intrinsically covered.

The manufacturer shall compile a test report that identifies:

- the nodes that were submitted to the testing;
- the qualitative results and the quantitative results, if any, of each test step.

The manufacturer shall keep the DuT nodes and eventually use them for the Reference Configuration Conformance Testing.

H.4.2 - Reference Configuration Conformance Testing

The Reference Configuration Conformance Testing is executed in a laboratory that is homologated according to the rules specified in point 8.3 - page 32 .

There are two types of nodes which may be homologated

- non-redundant UIC 556 nodes,
- redundant UIC 556 nodes.

In order to run the test, the manufacturer shall provide:

	TTC1	TTC2	TTC3
non-redundant UIC 556 nodes	3	4	21
redundant UIC 556 nodes	1 + 2	-	20 + 1

The nodes shall have been submitted by the manufacturer himself to the homogeneous configuration conformance test and a certification statement, declaring that the test was passed, shall be provided.

The Reference Configuration Conformance Testing is performed using the TTC1, TTC2, and TTC3 test composition, as well as the devices under Test (DUT) and Reference nodes that shall be already homologated according to point 8 - page 32.

The relevant test cases with TTC1 and TTC2 shall be executed twice, each time is called a run.

The first run is executed with the Ref nodes and the DuT nodes in the positions specified by the Tables 8 - page 127 and 9 - page 128 then, during the second run, the positions hold by the Ref nodes are hold by the DuT nodes and vice versa.

For practical reasons, the relevant test cases with TTC3 are executed at the Manufacturer's location with in presence of the personnel of the homologated laboratory.

H.5 - Test bed

The test bed is implemented by the relevant Testing Train Configuration that shall support the following manipulations in the Testing Train Configuration itself:

- coupling and uncoupling of nodes;
- turning on and off of nodes;
- redundancy switching over.

Furthermore it is necessary to attach the UIC Tester to the relevant node that has to perform the action as it is specified in the test sequence described in point H.6.1 - page 132.

The UIC Tester and a Line Monitor, when necessary, are needed in order to log the results of the action and check them for the qualitative and quantitative aspects.

It is not required that the DuT and the Ref node have exposed interfaces dedicated to the purpose of the test, the existing WTB and MVB interfaces are sufficient and the UIC Tester is believed to be normally attached to the MVB interface of the relevant node that causes the action or, when the MVB bus is not used by the node, to the application interface of the node itself.

The monitoring and logging of the results is performed by instrumentation devices attached to the WTB and/or MVB interface (this may be the UIC Tester itself if it provides this possibility).

Nevertheless, if the node provides a service interface, this interface may be used to provide the logging data to a PC unit.

In order to reduce the testing cost and the total duration of the testing, a full automation of the testing sequences is suggested but not required.

Full automation also has the advantage of reducing the number of human errors and obtaining an automatic print-out of the results and checks.

The main areas to be automated are:

- the coupling and uncoupling of the WTB connection between the nodes of the Testing Train Configuration in order to control by means of the UIC Tester the shortening and the lengthening of the configuration;
- the switching on and off of the nodes in order to control by means of the UIC Tester the simulation of the node failure and node re-insertion;
- the switching-over of the redundant node in order to control in a precise and simultaneous way the taking over of the redundant node.

H.6 - Test suites

H.6.1 - Guidelines to test case interpretation and execution

The test cases are grouped and each group is reported in one paragraph of this document; the grouping criteria are to test homogeneous features specified by the *UIC Leaflet 556* such homogeneous features are listed in the paragraph title of the test group itself.

Each paragraph is structured in the following topics:

Purpose:

This is a synopsis highlighting the main aims of the test cases group.

Reference to UIC 556:

The references to the relevant points of *UIC Leaflet 556* are listed, where the characteristics that are tested by the test cases reported in the **test sequence** are specified.

Train composition:

Here is specified which testing train compositions are used to run the test cases.

NB : This topic reports, if any, all the specific notes that may be useful to better understand the test cases.

Test sequence:

The test sequence is given in a table where the test cases are listed in the order in which they shall be executed.

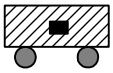
The table has five columns: step, action, expected results, checks, expected configuration.

The meaning of each column is listed hereinafter:

- the first column is the test step number;
- the second column describes which action shall be done on the testing configuration in order to generate stimuli;
- the third column describes the qualitative and quantitative results which are the consequences of the action and shall be observed and logged;
- the forth column describes which parameters, data on the bus and/or events shall be checked;
- the fifth column describes, possibly graphically, the train configuration that is expected to be verified in the NADI with particular reference to the TCN and UIC numbering.

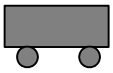
Graphic Symbols:

Hereinafter graphic symbols and their meanings are shown as they appear in the fifth column of the test sequence table.



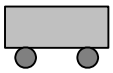
B
00

Fig. 18 - Vehicles with unknown UIC adress 00
(a vehicle which cannot be integrated into a confirmed composition)



gap
02

Fig. 19 - confirmed missing vehicle with UIC adress 02
(a vehicle which has not been present during correction/confirmation)



missing
04

Fig. 20 - unconfirmed missing vehicle with UIC adress 04
(a vehicle which was present during correction/confirmation)

The following explanations are relevant to the operation mode of nodes:

- TCN adresses are preceded by T, UIC adresses are preceded by U.

H.6.2 - List of Test Cases

The following is a list of test cases which should be covered by both the Manufacturer Conformance Test and the Reference Conformance Test.

1. Confirmed compositions test suites

   - Confirmation, correction and reset of train compositions
   - Loss and integration of intermediate nodes
   - Loss and integration of end nodes for confirmed compositions
   - Lengthening of confirmed compositions
   - Coupling of two confirmed compositions
   - Setting the vehicle number for seat reservation

2. UIC redundancy test suites

   - Redundancy switch-over for unconfirmed compositions
   - Redundancy switch-over for confirmed compositions
   - Duration of redundancy switch-over

3. Performance test suite

   - Duration of UIC inauguration
   - Power consumption

4. Unconfirmed compositions test suites

   - applied to TTC1
   - applied to trainsets of all possible lengths

5. Leading vehicle test suites
6. Inhibit functionality test suites
7. Group handling test suites (B001…B006)
8. E telegram error handling test suites
9. Sleep mode test suites

H.6.3 - Description of test suites

H.6.3.1 - Confirmed compositions test suites

Purpose:

Test the features for confirmed compositions, especially

- confirmation, correction and reset of train compositions;
- loss and integration of intermediate nodes;
- loss and integration of end nodes;
- lengthening of confirmed compositions;
- coupling of two confirmed compositions;
- setting the vehicle number for seat reservation.

Additionally, test the following additional features of leading requests for a confirmed composition

- ignore a leading request from a vehicle with unknown UIC address

NB : For computing the composition, the following rules are applied to the first 4 test cases:

Test case	**0**	**0**	**0**	**0**
rule "a"	3, 6	8, 9, 10, 11, 19	2, 3, 6, 11	
rule "b"		12, 26		
rule "c"	2, 4, 5, 7, 8, 9	2, 4, 6, 7, 15, 17, 18, 22, 24, 25	4, 5, 7, 12, 13, 14, 5	
no inversion			8, 9, 10	3, 4

H.6.3.1.1 - Confirmation, correction and reset of train compositions.

Purpose:

Test the correct handling of correction and confirmation E telegrams:

	test steps
• NADI confirmation	2
• NADI correction	5
• NADI invalidation	8
• incomplete distribution of correction information	9
• change of the leading vehicle	3, 4, 6, 7

Reference to UIC 556:

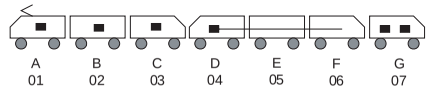
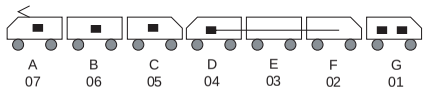
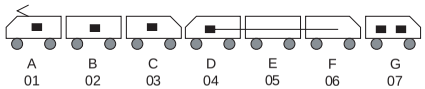
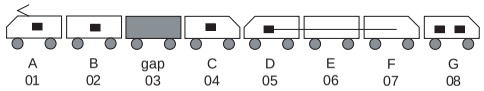
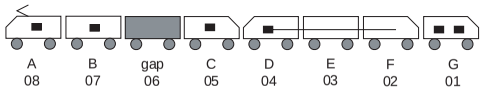
Point 5.1 - page 10	operational mode: INAUGURATION/REGULAR	
Point C - page 37	description of algorithm	
Point C.2.5 - page 65	UMS user services / E telegrams:	
Appendix A - page 35, No 15.02	write_correction:	UIC_FC_CORRECTION
Appendix A - page 35, No 15.01		UIC_FC_DELETE_CONFIG
Appendix A - page 35, No 15.07	multicast_create:	UIC_FC_MULTICAST_CREATE
Point C.2.8 - page 82	Leading:	(local)
Point C.2.6.2.2 - page 71	correction handling	

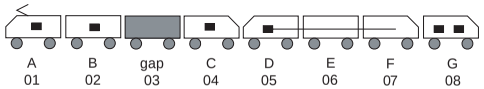
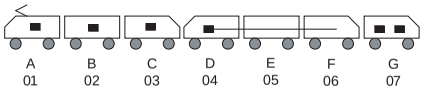
Train composition:

TTC1 defined in Point H.3.1 - page 127. The key positions are the train bus nodes of vehicles A, C, D/E/F and G.

NB : Nothing to note.

Test sequence:

Step	Action	Expected result	Checks	Expected configuration
1	turn A - G on	TCN/UIC inauguration with A - G: ordering according to rule "c"	• unconf. NADI (8 entries) • no leading vehicle	
2	confirm ordering	UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - G: "UIC adress set") • conf. NADI (8 entries) • no leading vehicle	see above
3	set leading request on G_2 in direction 1	UIC inauguration with A - G: ordering according to rule "a"	• inauguration data (A - G: "conf. configuration", info on conf. configuration; G_2: requests leading in direction 1) • conf. NADI (8 entries) • leading vehicle = 01	
4	reset leading request on G_2	UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - G: "conf. configuration", info on conf. configuration) • conf. NADI (8 entries) • no leading vehicle	
5	correct ordering with A, B, gap, C, D, E, F, G	UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - G: "UIC address set", info on conf. configuration) • conf. NADI (9 entries) • no leading vehicle	
6	set leading request on G2 in direction 1	UIC inauguration with A - G: ordering according to rule "a"	• inauguration data (A - G: "conf. configuration", info on conf. configuration; G2: requests leading in direction 1) • conf. NADI (9 entries) • leading vehicle G_2 = 01	

Step	Action	Expected result	checks	Expected configuration
7	reset leading request on G_2	UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - G: "conf. configuration", info on conf. configuration); • conf. NADI (9 entries) • no leading vehicle	
8	invalidate ordering on vehicle G_2	UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - G_1: "conf. configuration", info on conf. configuration; G_2: "delete conf. configuration") • unconf. NADI (8 entries) • no leading vehicle	
9	confirm ordering on A only and enforce a UIC inauguration	UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A: "UIC address set") • unconf. NADI (8 entries) • no leading vehicle	see above

H.6.3.1.2 - Loss and integration of intermediate nodes

Purpose:

Test the correct handling of correction and confirmation for intermediate nodes with:

	Test steps
• node failure	3, 5, 16, 23
• successful node reinsertion	4, 8, 17, 18, 24, 25, 26
• failed node reinsertion	6, 11
• change of the leading vehicle	8, 19, 26

and the correct handling of leading requests for confirmed compositions with

• setting the leading property for an intermediate vehicle with unknown UIC address	7, 8
• leading vehicle conflict resolution (case one vehicle was and requests leading and another one requests leading)	9, 10, 11

Reference to UIC 556:

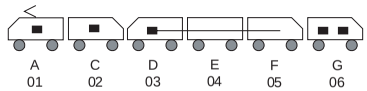
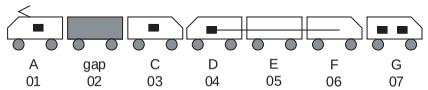
Point 5.1 - page 10	operational mode: INAUGURATION/REGULAR	
Point 6.2 - page 27	Failure tolerance	
Point C.1.3.5 - page 47	Description of algorithm: node, failure, node insertion, unknown nodes	
Point C.2.5 - page 65	UMS user services / E-telegrams:	
Appendix A - page 35, No 15.02	write_correction:	UIC_FC_CORRECTION
Appendix A - page 35, No 15.07	multicast_create:	UIC_FC_MULTICAST_CREATE
Point C.2.8 - page 82	leading:	(local)
Point C.2.6.2.2 - page 71	correction handling	

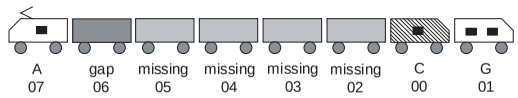
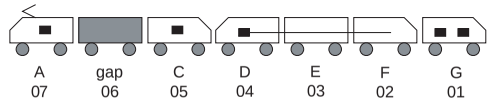
Train composition:

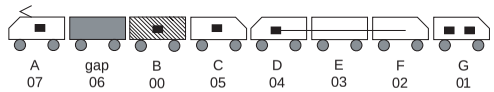
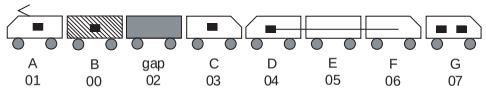
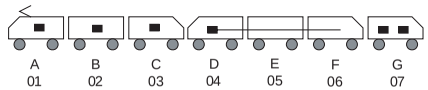
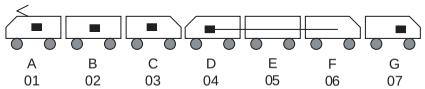
TTC1 defined in point H.3.1 - page 127. The key positions are the train bus nodes of vehicles C, D/E/F and G (1st node).

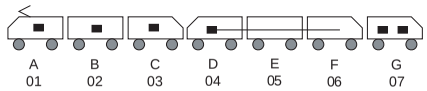
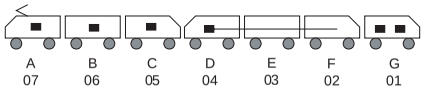
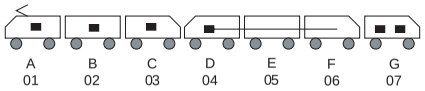
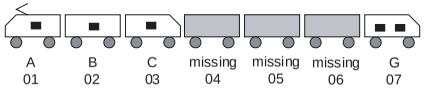
NB : Nothing to note.

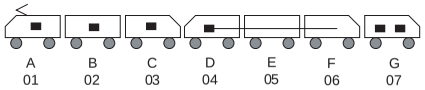
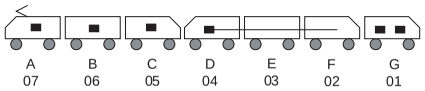
Test sequence

Step	Action	expected result	Checks	Expected configuration
1	turn A, C - G on with master A	TCN/UIC inauguration with A, C - G: ordering according to rule "c"	• unconf. NADI (7 entries) • no leading vehicle	
2	correct ordering with A, gap, C, D, E, F, G	UIC inauguration with A, C - G: ordering according to rule "c"	• inauguration data (A, C - G: "UIC address set"; info on conf. configuration) • conf. NADI (8 entries) • no leading vehicle	
3	turn C off	no UIC inauguration (TCN master at A)	• Topography (counter not changed) • NADI unchanged	see above
4	turn C on	UIC inauguration with A, C - G: ordering according to rule "c" (Position of C can be determined)	• inauguration data (A, D - G: "conf. configuration", info on conf. configuration; C: no info on conf. configuration) • conf. NADI (8 entries) • no leading vehicle	see above
5	turn C and D/E/F off	no UIC inauguration (TCN master at A)	• Topography (counter not changed) • NADI unchanged	see above
6	turn C on	TCN/UIC inauguration with A, C, G: ordering according to rule "c" (position of C cannot be determined)	• inauguration data (A, G: "conf. configuration", info on conf. configuration; C: no info on conf. configuration) • conf. NADI (9 entries; missing + unknown vehicle) • no leading vehicle	

Step	Action	Expected result	checks	Expected configuration
7	set leading request on C in direction 1	UIC inauguration with A, C - G: ordering according to rule "c" (position of C cannot be determined) (C has unknown UIC address)	• inauguration data (A, G: "conf. configuration", info on conf. configuration; C: no info on conf. configuration, requests leading in direction 1) • conf. NADI (9 entries; missing + unknown veh.) • no leading vehicle	see above
8	set leading request on G_2 in direction 1	UIC inauguration with A, C - G: ordering according to rule "a" (position of C cannot be determined) (C has unknown UIC address)	• inauguration data (A, G: "conf. configuration", info on conf. configuration; C: no info on conf. configuration, requests leading in direction 1; G_2: requests leading in direction 1) • conf. NADI (9 entries; missing + unknown veh.) • leading vehicle G_2 = 01	
9	turn D/E/F on	TCN/UIC inauguration with A, C - G: ordering according to rule "a" (position of C, D/E/F can be determined)	• inauguration data (A, G: "conf. configuration", info on conf. configuration; C - F: no info on conf. configuration; C: requests leading in direction 1; G_2: was and requests leading in direction 1) • conf. NADI (8 entries) • leading vehicle G_2 = 01	
10	correct ordering with A, gap, C, D, E, F, G	UIC inauguration with A , C - G: ordering according to rule "a"	• inauguration data (A, C - G: "UIC adress set", info on conf. configuration; C: requests leading; G_2: was and requests leading in direction 1) • conf. NADI (8 entries) • leading vehicle G_2 = 01	see above

Step	Action	Expected result	Checks	Expected configuration
11	turn B on	TCN/UIC inauguration with A - G ordering according to rule "a" (position of B cannot be determined)	• inauguration data (A, C - G: "conf. configuration", info on conf. configuration; B: no info on conf. configuration; C: requests leading in direction 1; G_2: was and requests leading in direction 1) • conf. NADI (9 entries) • leading vehicle G_2 = 01	
12	reset leading request on G_2	TCN/UIC inauguration with A - G ordering according to rule "b" (position of B cannot be determined)	• inauguration data (A, C - G: " conf. configuration", info on conf. configuration; B: no info on conf. configuration; C: requests leading in direction 1) • conf. NADI (9 entries) • leading vehicle C = 03	
13	turn off			
14	switch A - G on with master at A	TCN/UIC inauguration with A - G: ordering according to rule "c"	• unconf. NADI (8 entries) • no leading vehicle	
15	Confirm ordering	UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - G: "UIC adress set") • conf. NADI (8 entries) • no leading vehicle	see above
16	turn D/E/F and G_1 off	no UIC inauguration (TCN master at A)	• Topography (counter not changed) • NADI unchanged	see above
17	turn D/E/F on	TCN/UIC inauguration A - F, G_2: ordering according to rule "c" (position of D/E/Fcan be determined)	• inauguration data (A - C, G_2: "conf. configuration", info on conf. configuration; D/E/F: no info on conf. configuration) • conf. NADI (7 entries) • no leading vehicle	

Step	Action	Expected result	Checks	Expected configuration
18	turn G_1 on	TCN/UIC inauguration A - G: oredering according to rule "c" (position of G1 can be determined)	• inauguration data (A - F, G_2: "conf. configuration", info on conf. configuration; G_1: no info on conf. configuration) • conf. NADI (8 entries) • no leading vehicle	
19	set leading request on G_2 in direction 1	UIC inauguration with A - G: ordering according to rule "a"	• inauguration data (A - G: "conf. configuration", info on conf. configuration; G_2 requests leading in direction 1) • conf. NADI (8 entries) • leading vehicle G2 = 01	
20	turn off			
21	turn A - G on with master at A	TCN/UIC inauguration with A - G: ordering according to rule "c"	• unconf. NADI (8 entries) • no leading vehicle	
22	Confirm ordering	UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - G: "UIC adress set") • conf. NADI (8 entries) • no leading vehicle	see above
23	turn D/E/F and G_1off	no UIC inauguration (TCN master at A)	• Topography (counter not changed) • NADI unchanged	see above
24	turn G_1 on	TCN/UIC inauguration A - C, G: ordering according to rule "c" (position of G_1 can be determined)	• inauguration data (A - C, G_2: "conf. configuration", info on conf. configuration; G_1: no info on conf. configuration) • conf. NADI (8 entries; missing veh.) • no leading vehicle	

Step	Action	Expected result	Checks	Expected configuration
25	turn D/E/F on	TCN/UIC inauguration A - G: ordering according to rule "c" (position of D/E/F can be determined)	• inauguration data (A - C, G: "conf. configuration", info on conf. configuration; D/E/F: no info on conf. configuration) • conf. NADI (8 entries) • no leading vehicle	
26	set leading request on F (direction 2)	UIC inauguration A - G: ordering according to rule "b"	• inauguration data (A - G: "conf. configuration", info on conf. configuration; D/E/F: requests leading in direction 1) • conf. NADI (8 entries) • leading vehicle F = 02	

H.6.3.1.3 - Loss and integration of end nodes for confirmed compositions

Purpose:

Test the correct handling of confirmed compositions for end nodes with:

	Test steps
• node failure	4, 8
• successful node reinsertion	5
• failed node reinsertion	9
• change of the leading vehicle	6, 7, 11

and the correct handling of leading requests for confirmed compositions with:

• setting the leading property for an end vehicle with unknown UIC address	10, 15

Reference to UIC 556:

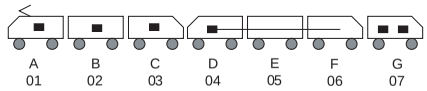
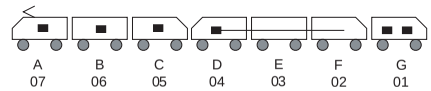
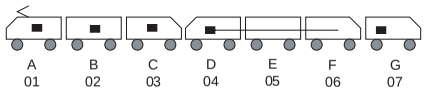
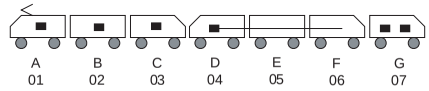
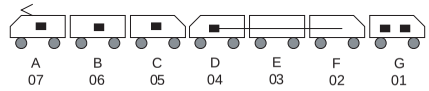
Point 5.1 - page 10	operational modes: INAUGURATION/REGULAR	
Point 6.2 - page 27	failure tolerance	
Point C.1.3.5 - page 47	description of algorithm	
Point C.2.5 - page 65	UMS user services / E-Telegrams:	
Appendix A - page 35, No 15.02	write_correction:	UIC_FC_CORRECTION
Appendix A - page 35, No 15.07	multicast_create:	UIC_FC_MULTICAST_CREATE
Point C.2.8 - page 82	leading:	(local)
Point C.2.6.2.2 - page 71	correction handling	

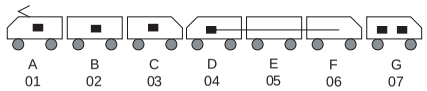
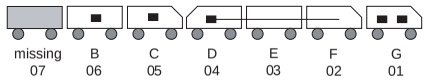
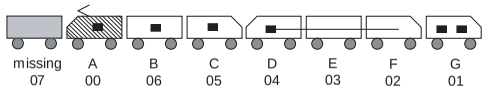
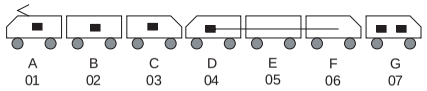
Train composition:

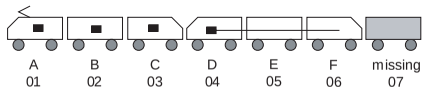
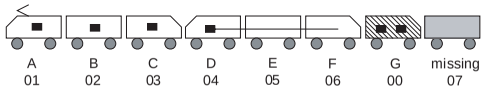
TTC1 as defined in point H.3.1 - page 127. The key positions are the train bus nodes of vehicles A and G_2.

NB : Nothing to note.

Test sequence

Step	Action	Expected result	Checks	Expected configuration
1	turn A - G on with master at A	TCN/UIC inauguration with A - G: ordering according to rule "c"	• unconf. NADI (8 entries) • no leading vehicle	
2	set leading request on G_2 in direction 1	UIC inauguration with A - G: ordering according to rule "a"	• inauguration data (G_2: requests leading in direction 1)) • unconf. NADI (8 entries) • leading vehicle G_2 = 01	
3	Confirm ordering	UIC inauguration with A - G: ordering according to rule "a"	• inauguration data (A - G: "UIC address set"; G_2: was and requests leading in direction 1) • conf. NADI (8 entries) • leading vehicle G_2 = 01	see above
4	turn G_2 off	TCN/UIC inauguration with A - G_1: ordering according to rule "c"	• inauguration data (A - G_1: "conf. configuration", info on conf. configuration) • conf.NADI (7 entries) • no leading vehicle	
5	turn vehicle G_2 on	TCN/UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - G_1: "conf. configuration", info on conf. configuration; G_2: no info on conf. configuration) • conf. NADI (8 entries) • no leading vehicle	
6	set leading request on G_2 in direction 1	UIC inauguration with A - G: ordering according to rule "a"	• inauguration data (A - G: "conf. configuration", Info on conf. configuration; G_2: requests leading in direction 1) • conf. NADI (8 entries) • leading vehicle G_2 = 01	

Step	Action	Eexpected result	Checks	Expected configuration
7	reset leading request on G_2	UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - G: "conf. configuration", info on conf. configuration) • conf. NADI (8 entries) • no leading vehicle	
8	turn A off	TCN/UIC inauguration with B - G: no inversion of ordering (rules a - c cannot be applied	• inauguration data (B - G: "conf. configuration", info on conf. configuration) • conf. NADI (8 entries; missing veh.) • no leading vehicle	
9	turn A on	TCN/UIC inauguration with A - G: no inversion of ordering (rules a - c cannot be applied)	• inauguration data (B - G: "conf. configuration", info on conf. configuration; A: no info on conf. configuration) • conf. NADI (9 entries; missing + unknown veh.) • no leading vehicle	
10	set leading request on A in direction 1	UIC inauguration with A - G: no inversion of ordering (A has unknown UIC address; rules a - c cannot be applied)	• inaugration data (B - G: "conf. configuration", info on conf. configuration; A: no info on conf. configuration, requests leading in direction 1) • conf. NADI (9 entries; missing + unknown veh.) • no leading vehicle	see above
11	confirm ordering	UIC inauguration with A - G: ordering according to rule "a"	• inauguration data (A - G: "UIC address set"; A: requests leading in direction 1) • conf. NADI (8 entries) • leading vehicle A = 01	

Step	Action	Expected result	Checks	Expected configuration
12	reset leading request on A	UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - G_2: "conf. configuration", info on conf. configuration) • conf. NADI (8 entries) • no leading vehicle	see above
13	turn G_1, G_2 off	TCN/UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - G_2: "conf. configuration", info on conf. configuration) • conf. NADI (7 entries; missing nodes) • no leading vehicle	
14	turn G_1, G_2 on	TCN/UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - F: "conf. configuration", info on conf. configuration; G: no info on conf. configuration) • conf. NADI (9 entries; missing + unknown nodes) • no leading vehicle	
15	Set leading request on G_2	UIC inauguration with A - G: ordering according to rule "c" (G_2 has unknown UIC address)	• inauguration data (A - F: "conf. configuration", info on conf. configuration; no info on conf. configuration; G_2: requests leading in direction 1) • conf. NADI (8 entries) • no leading vehicle	see above

H.6.3.1.4 - Lengthening of confirmed compositions

Purpose:

Test the correct handling of confirmed compositions:

	Test steps
• lengthening of confirmed compositions	3, 5

and the correct handling of leading requests for confirmed compositions with:

• setting the leading property for an end vehicle with unknown UIC address	4

Reference to UIC 556:

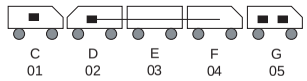
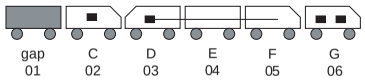
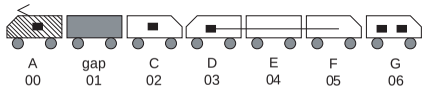
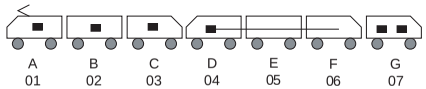
Point 5.1 - page 10	operational modes: INAUGURATION/REGULAR	
Point C.1.3.5 - page 47	description of algorithm	
Point C.2.5 - page 65	UMS user services / E-Telegrams:	
Appendix A - page 35, No 15.02	write_correction:	UIC_FC_CORRECTION
Appendix A - page 35, No 15.07	multicast_create:	UIC_FC_MULTICAST_CREATE
Point C.2.8 - page 82	leading:	(local)
Point C.2.6.2.2 - page 71	correction handling	

Train composition:

TTC1 as defined in point H.3.1 - page 127. The key positions are the train bus nodes of vehicles A, C, D/E/F and G.

NB : Nothing to note.

Test sequence:

Step	Action	Expected result	Checks	Expected configuration
1	turn C - G on with master at D	TCN/UIC inauguration with C - G: ordering according to rule "d"	• unconf. NADI (6 entries) • no leading vehicle	
2	correct ordering with gap C, D, E, F, G	UIC inauguration with C - G: ordering according to rule "d"	• inauguration data (C - G: "UIC address set", info on conf. configuration) • conf. NADI (7 entries) • no leading vehicle	
3	turn A on	TCN/UIC inauguration with A, C - G: no inversion of ordering (no lengthening)	• inauguration data (C - G: "conf. configuration", info on conf. configuration; A: no info on conf. configuration) • conf. NADI (8 entries; unknown veh.) • no leading vehicle	
4	Set leading request on A	UIC inauguration with A, C - G: no inversion of ordering (UIC address of A is unknown)	• inauguration data (C - G: "conf. configuration", info on conf. configuration; A: no info on conf. configuration, requests leading in direction 1) • conf. NADI (8 entries; unknown veh.) • no leading vehicle	see above
5	turn B on	TCN/UIC inauguration with A - G: ordering according to rule "a" (lengthening)	• inauguration data (C - G: "conf. configuration", Info on conf. configuration; A, B: no info on conf. configuration; A: requests leading in direction 1) • unconf. NADI (8 entries) • leading vehicle A = 01	

H.6.3.1.5 - Coupling of two confirmed compositions.

Purpose:

Test the correct handling of confirmed compositions:

coupling of a confirmed composition with another confirmed composition	*Test steps*
• with a different number of confirmed vehicles	6
• with a different list of confirmed missing vehicles	12
• resulting in a number of confirmed vehicles larger than the confirmed number of vehicles	18

Reference to UIC 556:

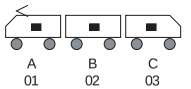
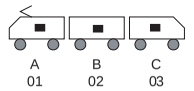
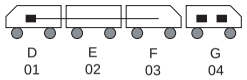
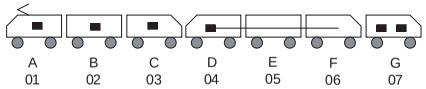
Point 5.1 - page 10	operational modes: INAUGURATION/REGULAR	
Point 6.2 - page 27	failure tolerance	
Point C.1.3.5 - page 47	description of algorithm	
Point C.2.5 - page 65	UMS user services / E-Telegrams:	
Appendix A - page 35, No 15.02	write_correction:	UIC_FC_CORRECTION
Appendix A - page 35, No 15.07	multicast_create:	UIC_FC_MULTICAST_CREATE
Appendix A - page 35, No 15.06	inauguration_control:	UIC_FC_INAUGURATION_CONTROL
Point C.2.6.2.2 - page 71	correction handling	

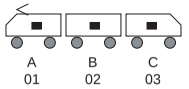
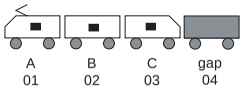
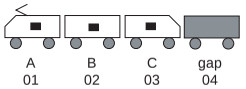
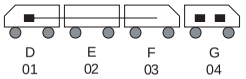
Train composition:

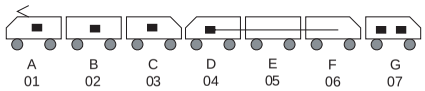
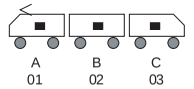
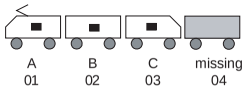
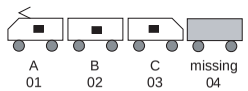
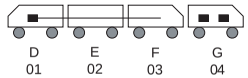
TTC1 as defined in point H.3.1 - page 127. The key positions are the train bus nodes of vehicles A, C, D/E/F and G.

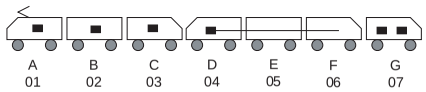
NB : Nothing to note.

Test sequence:

Step	Action	Eexpected result	Checks	Expected configuration
1	turn A - C on	TCN/UIC inauguration with A - C: ordering according to rule "c"	• unconf. NADI (3 entries) • no leading vehicle	
2	confirm ordering with A - C	UIC inauguration with A - C: ordering according to rule "c"	• inauguration data (A - C: "UIC address set") • conf. NADI (3 entries) • no leading vehicle	see above
3	set inhibit for A	INHIBIT is set	• UWTM Status (Inhibit)	see above
4	turn D - G on	TCN/UIC inauguration with D - G: ordering according to rule "c"	composition 1 (A - C): • conf. NADI (3 entries) • no leading vehicle composition 2 (D - G): • unconf. NADI (5 entries) • no leading vehicle	 
5	confirm ordering with D - G	UIC inauguration with D - G: ordering according to rule "c"	composition 1 (A - C): • conf. NADI (3 entries) • no leading vehicle composition 2 (D - G): • inauguration data (D - G: "UIC address set") • conf. NADI (5 entries) • no leading vehicle	see above
6	reset inhibit for A	TCN/UIC inauguration with A - G: ordering according to rule "c" (the conf. numbers of veh. differ)	• inauguration data (A - G: "conf. composition", different confirmed number of vehicles) • unconf. NADI (8 entries) • no leading vehicle	

Step	Action	Eexpected result	Checks	Expected configuration
7	turn D - G off	TCN/UIC inauguration with A - C: ordering according to rule "c"	• unconf. NADI (3 entries) • no leading vehicle	
8	correct ordering with A, B, C, gap	UIC inauguration with A - C: ordering according to rule "c"	• inauguration data (A - C: "UIC address set", info on conf. configuration) • conf. NADI (4 entries) • no leading vehicle	
9	set Inhibit for A	INHIBIT is set	• UWTM status (Inhibit)	see above
10	turn D - G on	TCN/UIC inauguration with D - G: ordering according to rule "c"	composition 1 (A - C): • conf. NADI (4 entries) • conf. missing vehicle 04 • no leading vehicle composition 2 (D - G): • unconf. NADI (5 entries) • no leading vehicle	 
11	confirm ordering with D - G	UIC inauguration with D - G: ordering according to rule "c"	composition 1 (A - C): • conf. NADI (4 entries) • no leading vehicle composition 2 (D - G): • inauguration data (D - G: "UIC address set") • conf. NADI (5 entries) • no leading vehicle	see above

Step	Action	Eexpected result	Checks	Expected configuration
12	reset Inhibit for A	TCN/UIC inauguration with A - G: ordering according to rule "c" (the lists of conf. missing veh. differ)	• inauguration data (A - G: "conf. configuration"; different confirmed missing vehicles lists • unconf. NADI (8 entries) • no leading vehicle	
13	turn D - G off	TCN/UIC inauguration with A - C: ordering according to rule "c"	• unconf. NADI (3 entries) • no leading vehicle	
14	correct ordering with A, B, C, D	UIC inauguration with A - C: ordering according to rule "c"	• inauguration data (A - C: "UIC address set"; info on conf. configuration) • conf. NADI (4 entries; missing veh.) • no leading vehicle	
15	set Inhibit for A	INHIBIT is SET	• UWTM status (Inhibit)	see above
16	turn D - G on	TCN/UIC inauguration with D - G: ordering according to rule "c"	composition 1 (A - C): • conf. NADI (4 entries; missing veh.) • no leading vehicle composition2 (D - G): • unconf. NADI (5 entries) • no leading vehicle	 

Step	Action	Eexpected result	Checks	Expected configuration
17	confirm ordering with D - G	UIC inauguration with D - G: ordering according to rule "c"	composition 1 (A - C): • conf. NADI (4 entries; missing veh.) • no leading vehicle composition 2 (D - G): • inauguration data (D - G: "UIC address set") • conf. NADI (5 entries) • no leading vehicle	see above
18	reset Inhibit for A	TCN/UIC inauguration with A - G: ordering according to rule "c" (the number of conf. veh. is larger than the conf. number of veh.)	• inauguration data (A - G: "conf. configuration"; info on conf. configuration) • unconf. NADI (8 entries) • no leading vehicle	

H.6.3.1.6 - Setting the vehicle number for seat reservation

Purpose:

Test the correct handling of setting the vehicle reservation number.

Reference to UIC 556:

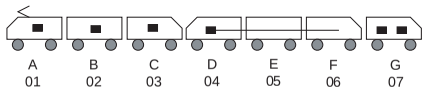
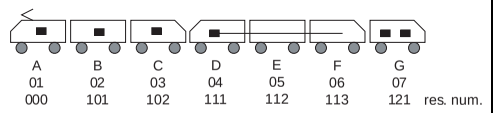
Point 5.1 - page 10	operational mode: INAUGURATION/REGULAR	
Point C.2.5 - page 65	UMS user services / E-Telegrams:	
Appendix A - page 35, No 15.03	write_res_nr:	UIC_FC_WRITE_RES_NR
Appendix A - page 35, No 15.07	multicast_create:	UIC_FC_MULTICAST_CREATE
Point C.2.6.2.2 - page 71	vehicle reservation handling	

Train composition:

TTC1 as defined in point H.3.1 - page 127. The key positions are the train bus nodes of vehicles A, C, D/E/F and G.

NB : Nothing to note.

Test sequence:

Step	Action	Eexpected result	Checks	Expected configuration
1	turn A - G on	TCN/UIC inauguration with A - G: ordering according to rule "c"	• unconf. NADI (8 entries) • no leading vehicle	
2	set veh. no. for seat reservation to (000, 101, 102, 111, 112, 113, 121)	UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - G: new vehicle reservation number) • unconf. NADI (8 entries) • no leading vehicle	

H.6.3.2 - UIC redundancy test suite

Purpose:

Test the features for redundancy switch-over, especially

- for unconfirmed compositions
- for confirmed compositions
- duration.

This includes redundancy switch-over for end nodes, intermediate nodes, and vehicles with multiple gateways.

NB : Nothing to note.

H.6.3.2.1 - Redundancy switch-over for unconfirmed compositions.

Purpose:

Test the correct handling of redundancy for unconfirmed compositions:

test case/test step	*2*	*3*	*4*
• intermediate node		X	
• end node	X		X
• vehicle with one gateway	X		
• trainset		X	
• vehicle with multiple gateways			X

Reference to UIC 556:

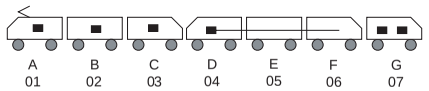
Point 4.1.1 und 4.1.2 - page 6	Bus structure
Point 5.1 - page 10	operational mode: INAUGURATION
Point 6.2 - page 27	failure tolerance
Point C.2.6 - page 67	UMS user services / messages:
Point C.2.8 - page 82	leading: (local)

Train compositions:

TTC1 as defined in point H.3.1 - page 127. The key positions are the train bus nodes of vehicles A, C, D/E/F and G_2.

NB : Nothing to note.

Test sequence:

Step	Action	Expected result	Checks	Expected configuration
1	turn A - G on	TCN/UIC inauguration with A - G: ordering according to rule "c"	• unconf. NADI (8 entries) • no leading vehicle	
2	trigger failure of active node for A	TCN/UIC inauguration with A - G: ordering according to rule "c" (successful redundancy switch-over of A)	• inauguration data (A: "Redundancy switch-over") • unconf. NADI (8 entries) • no leading vehicle	see above
3	trigger failure of active node for D - F	TCN/UIC inauguration with A - G: ordering according to rule "c" (successful redundancy switch-over of D/E/F)	• inauguration data (D-F: "Redundancy switch-over") • unconf. NADI (8 entries) • no leading vehicle	see above
4	trigger failure of active node for G_2	TCN/UIC inauguration with A - G: ordering according to rule "c" (redundancy switch-over of G_2)	• inauguration data (G_2: "Redundancy switch-over") • unconf. NADI (8 entries) • no leading vehicle	see above

H.6.3.2.2 - Redundancy switch-over for confirmed compositions

Purpose:

Test the correct handling of redundancy for confirmed compositions:

Test case/test step	3	6	9	13	17	20	23	27	30
• Intermediates nodes	X	X			X	X			
• end nodes			X	X		X	X	X	X
• vehicles with one node						X	X	X	X
• trainsets	X	X		X	X				
• vehicles with multiple nodes			X	X					
• simple gaps	X		X			X			
• multiple gaps (u: unconfirmed; c: confirmed gap)		u		(X)	c		u	c	u + c

Reference to UIC 556:

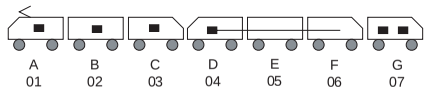
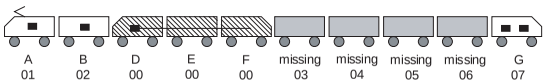
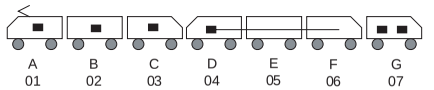
Point 4.1.1 and 4.1.2 - page 6	Bus structure	
Point 5.1 - page 10	operational mode: INAUGURATION/REGULAR	
Point 6.2 - page 27	failure tolerance	
Point C.1.3.5 - page 47	description of algorithm: redundant train bus nodes	
Point C.2.5 - page 65	UMS user services / messages:	
Appendix A - page 35, No 15.02	write_correction:	UIC_FC_CORRECTION
Appendix A - page 35, No 15.07	multicast_create:	UIC_FC_MULTICAST_CREATE
Point C.2.6.2.2 - page 71	correction handling	

Train composition:

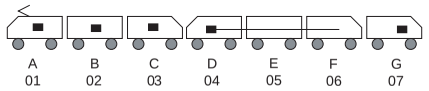
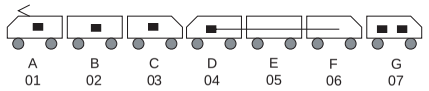
TTC1 as defined in point H.3.1 - page 127. The key positions are the train bus nodes of vehicles A, D/E/F and G.

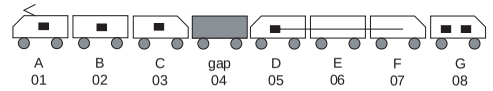
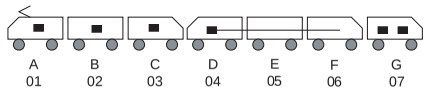
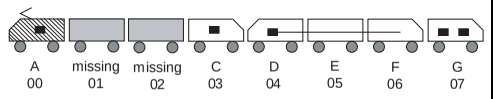
NB : Nothing to note.

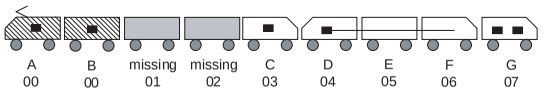
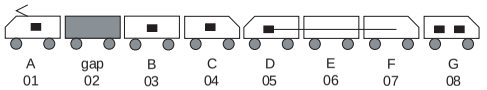
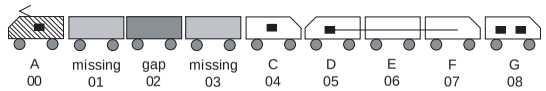
Test sequence:

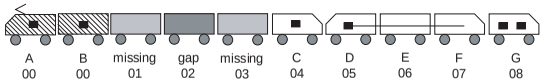
Step	Action	Expected result	Checks	Expected configuration
1	turn A - G on	TCN/UIC inauguration with A - G: ordering according to rule "c"	• unconf. NADI (8 entries) • no leading vehicle	
2	confirm ordering	UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - G: "UIC address set") • conf. NADI (8 entries) • no leading vehicle	see above
3	trigger failure of active node of D/E/F	TCN/UIC inauguration with A - G: ordering according to rule "c" (successful redundancy switch-over of D/E/F)	• inauguration data (A - C, G: "conf. configuration", info on conf. configuration; D/E/F: "Redundancy switch-over") • conf. NADI (8 entries) • no leading vehicle	see above
4	prepare deactivated standby node for D/E/F [a]	no UIC inauguration	• UWTM topography (not changed)	see above
5	turn C off	depends on position of TCN master	none	---
6	trigger failure of active node for D/E/F	TCN/UIC inauguration with A - B, D - G: ordering according to rule "c" (unsuccessful redundancy switch-over of D/E/F: position of D/E/F cannot be determined)	• inauguration data (A - B, G: "conf. configuration", info on conf. configuration; D/E/F: "Redundancy switch-over") • conf. NADI (11 entries; missing + unknown veh) • no leading vehicle	
7	turn C on	TCN/UIC inauguration with A - G: ordering according to rule "c" (Position of C, D/E/F can be determined)	• inauguration data (A - B, G: "conf. configuration", info on conf. configuration; C - F: no info on conf. configuration) • conf. NADI (8 entries) • no leading vehicle	

a. What actually needs to be done may depend on the hardware redundancy concept

Step	Action	Expected result	Checks	Expected configuration
8	prepare deactivated standby node for D/E/F	no UIC inauguration	• UWTM Topography (not changed)	see above
9	trigger failure of active node of G_2	TCN/UIC inauguration with A - G: ordering according to rule "c" (successful redundancy switch-over of G_2)	• inauguration data (A - G_1: conf. configuration info on conf. configuration; G_2: "redundancy switch-over") • conf. NADI (8 entries) • no leading vehicle	see above
10	prepare deactivated standby node for G_2	no UIC inauguration	• UWTM Topography (not changed)	see above
11	confirm ordering	UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - G: "UIC address set") • conf. NADI (8 entries) • no leading vehicle	see above
12	turn G_1 off	depends on position of TCN master	none	---
13	trigger failure of active node of G_2	TCN/UIC inauguration with A - F, G_2: ordering according to rule "c" (successful redundancy switch-over of G_2)	• inauguration data (A - F: "conf. configuration", info on conf. configuration; G_2 "redundancy switch-over" • conf. NADI (7 entries) • no leading vehicle	
14	turn G_1 on	TCN/UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - F; G_2: "conf. configuration", info on conf. configuration; G_1: no info on conf. configuration) • conf. NADI (8 entries) • no leading vehicle	
15	prepare deactivated standby node for G_2	no UIC inauguration	• UWTM Topography (not changed)	see above

Step	Action	Expected result	Checks	Expected configuration
16	correct ordering with A, B, C, gap, D, E, F, G	UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - G: "UIC address set", info on conf. configuration) • conf. NADI (9 entries) • no leading vehicle	
17	trigger failure of active node for D/E/F	TCN/UIC inauguration with A - G: ordering according to rule "c" (successful redundancy switch-over of D/E/F)	• inauguration data (A - C, G: "conf. configuration", info on conf. configuration; D/E/F: "Redundancy switch-over") D/E/F: "Redundancy switch-over") • conf. NADI (8 entries) • no leading vehicle	see above
18	prepare deactivated standby node for D/E/F	no UIC inauguration	• UWTM Topography (not changed)	see above
19	confirm ordering	UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - G: "UIC address set") • conf. NADI (8 entries) • no leading vehicle	
20	trigger failure of active node of A	TCN/UIC inauguration with A - G: ordering according to rule "c" (successful redundancy switch-over of A)	• inauguration data (B - G: "conf. configuration", info on conf. configuration; A: "redundancy switch-over") • conf. NADI (8 entries) • no leading vehicle	see above
21	prepare deactivated standby node for A	no UIC inauguration	• UWTM Topography (not changed)	see above
22	turn B off	depends on position of TCN master	none	---
23	trigger failure of active node of A	TCN/UIC inauguration with A, C - G: no inversion of ordering (A has unknown UIC address) (unsuccessful redundancy switch-over of A: position of A cannot be determined)	• inauguration data (C - G: "conf. configuration", info on conf. configuration; A: "redundancy switch-over") • conf. NADI (9 entries; missing + unknown vehicles) • no leading vehicle	

Step	Action	Expected result	Checks	Expected configuration
24	turn B on	TCN/UIC inauguration with A - G: no inversion of ordering (A and B have unknown UIC address)	• inauguration data (C - G: "conf. configuration", info on conf. configuration; A, B: no info on conf. configuration) • conf. NADI (10 entries; missing + unknown vehicles) • no leading vehicle	
25	prepare deactivated standby node for A	no UIC inauguration	• UWTM Topography (not changed)	see above
26	correct ordering with A, gap, B, C, D, E, F, G	UIC inauguration with A - G: ordering according to rule "c"	• inauguration data (A - G: "UIC address set", info on conf. configuration) • conf. NADI (9 entries) • no leading vehicle	
27	trigger failure of active node of A	TCN/UIC inauguration with A - G: ordering according to rule "c" (successful redundancy switch-over of A)	• inauguration data (B - G: "conf. configuration", info on conf. configuration; A: "Redundancy switch-over") • conf. NADI (9 entries) • no leading vehicle	see above
28	prepare deactivated standby node for A	no UIC inauguration	• UWTM Topography (not changed)	see above
29	turn B off	depends on position of TCN master	none	---
30	trigger failure of active node for A	TCN/UIC inauguration with A, C - G: no inversion of ordering (A has unknown UIC address (unsuccessful redundancy switch-over of A: position of A cannot be determined	• inauguration data (C - G: "conf. configuration", Info on conf. configuration; A: "redundancy switch-over") • conf. NADI (10 entries; unknown + missing vehicles) • no leading vehicle	

Step	Action	Expected result	Checks	Expected configuration
31	turn B on	TCN/UIC inauguration with A - G: no inversion of ordering (A has unknown UIC address) (B cannot be integrated)	• inauguration data (C - G: "conf. configuration", info on conf. configuration; A, B: no info on conf. configuration) • conf. NADI (11 entries; missing + unknown vehicles) • no leading vehicle	

H.6.3.2.3 - Duration of redundancy switch-over

Purpose:

Measure the duration of a UIC redundancy switch-over.

This shall typically take no longer than 1 s. At worst, it may take up to 1,4 s.

The duration of a UIC redundancy switch-over shall be measured with reference to:

Test step	*Reason for inauguration*	*Definition of switch-over time*
2	a simple failure of the redundant gateway (e.g. switch off active gateway)	time between the last PD frame before the node failure and the first valid PD frame (of vehicle A)

Reference to UIC 556:

Point 4.1.1 and 4.1.2 - page 6	bus structure
Point 6.2 - page 27	failure tolerance
Point C.1.3 - page 39	UIC inauguration

Train composition:

TTC1 as defined in point H.3.1 - page 127. All nodes are key positions.

NB : The duration of the redundancy switch-over is measured in a composition of 8, 16, and 22 nodes.

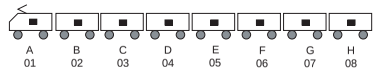
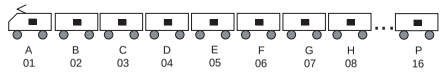
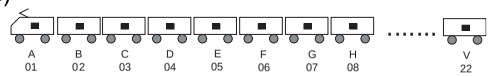
To achieve statistically significant values, the measurement shall be repeated at least 10 times for each configuration.

Table 11 : Duration of redundancy switch-over [a]
(N/min./max./mean/std. deviation)

Results	Number of nodes		
	a) 8 nodes (A-H)	b) 16 nodes (A-P)	c) 22 nodes (A-V)
N			
min.			
max.			
mean			
standard deviation			

a. UIC 556 is only normative with respect to the maximum inauguration duration. The values of the minimum, average values and the standard deviation are only documented to assess the failure coincidence of the measurements.

Test sequence:

Step	Action	Expected result	Checks	Expected configuration
1	turn on: a) A - H b) A - P ou c) A - V	TCN/UIC inauguration: ordering according to rule "c"	• unconf. NADI (a) 8, b) 16 or c) 22 entries)	a)  b)  c) 
2	trigger failure of the active node of A	TCN/UIC inauguration: ordering according to rule "c" (redundancy switch-over of A)	• unconf. NADI (a) 8, b) 16 or c) 22 entries) statistics • duration of redundancy switch-over	see above

H.6.3.3 - Performance test suite

Purpose:

Measure the performance characteristics for a UIC node, especially:

- duration of UIC/TCN and UIC inauguration;
- power consumption in sleep mode.

NB : Nothing to note.

H.6.3.3.1 - Duration of UIC inauguration

Purpose:

Measure the duration of a UIC inauguration.

This shall typically take no longer than 1 s. At worst, it may take up to 1,4 s.

The duration of a UIC inauguration shall be measured with reference to:

Test step	*Reason for inauguration*	*Definition of inauguration time*
2, 4	a new TCN inauguration (e.g. after lengthening or shortening of the train bus)	time between sending the first UNNAME telegram and the first valid PD frame (of vehicle A)
3	change of the UIC configuration (e.g. change of the leading vehicle)	time between sending the first Topography Request frame and the first valid PD frame (of vehicle A)

Reference to UIC 556:

Point 4.1.1 and 4.1.2 - page 6	Bus structure
Point 5.1 - page 10	Operational mode: INAUGURATION (duration for change of leading vehicle)
Point 5.6.2 - page 19	Function address 15
Point 6.2 - page 27	Failure tolerance
Point C.2.5 - page 65	UMS user services / messages:
Point C.2.7.2 - page 80	leading_request: (local)
Point C.1.3.1 - page 39	UIC inauguration

Train composition:

TTC3 as defined in point H.3.3 - page 129. All nodes are key positions.

NB : The duration of the inauguration is measured in a composition of 8, 16, and 22 nodes.

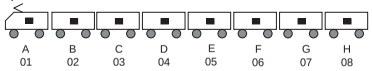
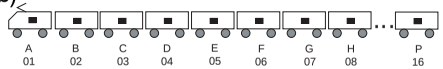
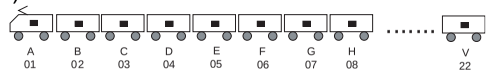
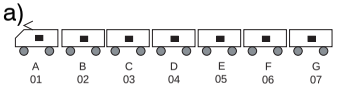
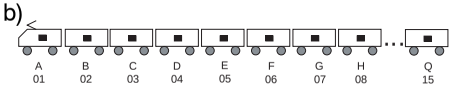
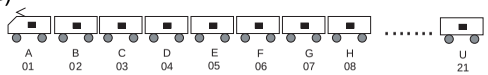
To achieve statistically significant values, the measurement shall be repeated at least 10 times for each configuration.

Table 12 : Duration of UIC inauguration [a]
(N/min/max/mean/std. deviation)

TCN inauguration	Results	Number of nodes (nn)		
		a) 8 nodes (A-H)	b) 16 nodes (A-P)	c) 22 nodes (A-V)
yes/ lengthening (step 2)	N			
	min.			
	max.			
	mean			
	standard deviation			
no (step 3)	N			
	min.			
	max.			
	mean			
	standard deviation			
yes/ shortening (step 4)	N			
	min.			
	max.			
	mean			
	standard deviation			

a. UIC 556 is only normative with respect to the maximum inauguration duration. The values of the minimum, average values and the standard deviation are only documented to assess the failure coincidence of the measurements.

Test sequence:

Step	Action	Expected result	Checks	Expected configuration
1	turn on: a) B - H b) B - P or c) B - V	TCN/UIC inauguration: ordering according to rule "c"	none	---
2	turn A on	TCN/UIC inauguration: ordering according to rule "c"	• unconf. NADI (a) 8, b) 16 or c) 22 entries) • no leading vehicle statistics • duration of TCN/UIC inauguration	a)  b)  c) 
3	set leading request A in direction 1	TCN/UIC inauguration: ordering according to rule "a	• unconf. NADI (a) 8, b) 16 or c) 22 entries) • leading vehicle A = 01 statistics • duration of UIC inauguration	see above
4	turn off: a) H b) P or c) V	TCN/UIC inauguration: ordering according to rule "a"	• unconf. NADI (a) 7, b) 15 or c) 21 entries) • leading vehicle A = 01 statistics • duration of TCN/UIC inauguration	a)  b)  c) 

H.6.3.3.2 - Power consumption

Purpose:

Verify that the power consumption of a single node in SLEEP mode is less than 10 W.

Reference to UIC 556:

Point 4.3 - page 8	power consumption in sleep mode < 10 W
Point 5.1 - page 10	operational mode: SLEEP

Train composition:

The tests concerning the SLEEP mode power consumption may be done with one single gateway.

NB : The power consumption of the train bus node shall be measured at the node's power supply input.

Test sequence:

Step	Action	Expected result	Checks	Expected configuration
1	turn on	train bus node is in operation	• node status: ON	---
2	set SLEEP request to vehicle A	vehicle A enters SLEEP mode	• node status : SLEEP	---
3	measure power consumption	power consumption below 10 W	• power consumption	---

H.6.3.4 - Unconfirmed compositions test suite

H.6.3.4.1 - Unconfirmed composition on TTC1

Purpose:

Test the correct computation of unconfirmed train compositions depending on

- the number of vehicles/nodes in the train;
- the number of vehicles in each trainset within the train;
- the number of double node vehicles (0/1);
- the position of the TCN master within the train;
- the existence and position of a leading vehicle.

Reference to UIC 556:

Point C.1.3.2 - page 39

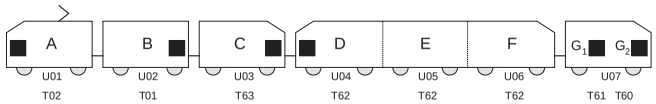
Composition:

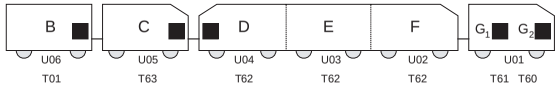
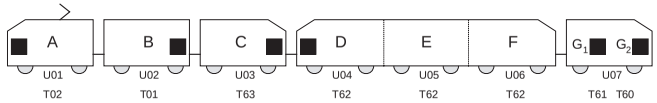
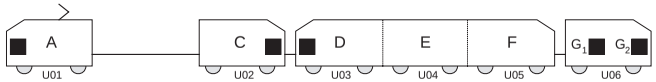
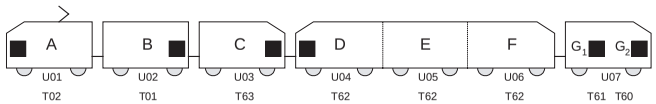
See reference composition.

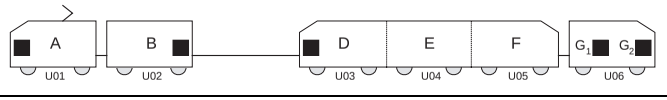
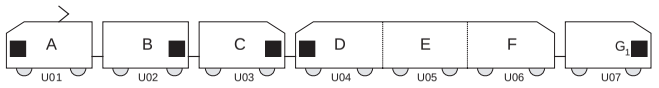
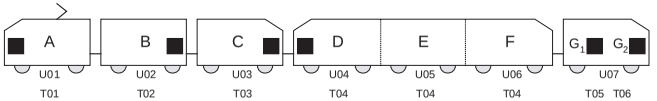
NB : Following cases shall be covered:

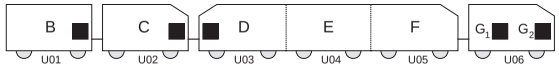
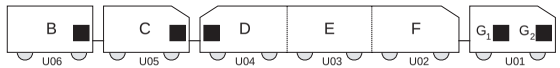
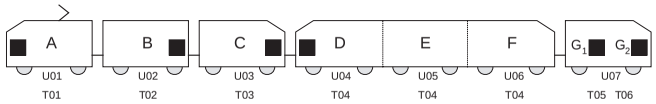
1. failure of single (a) intermediate / (b) end node(s) (master (') / slave nodes)
2. occurrence of single (a) intermediate / (b) end node(s)
3. no leading vehicle
4. leading vehicle (a) in an end node, (b) in the middle vehicle, (c) in another intermediate node/vehicle
5. (a) no / (b) one / (c) two traction unit(s) in end/intermediate node(s)/vehicle(s).

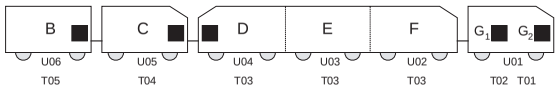
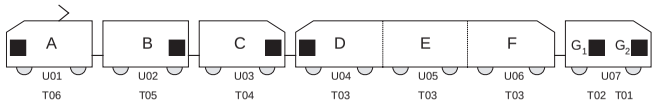
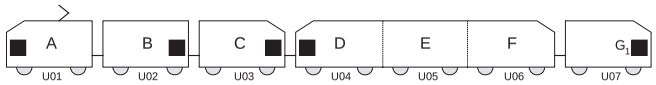
Test sequence:

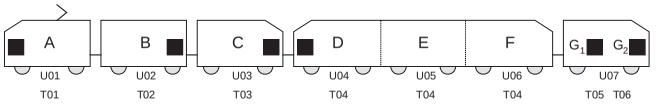
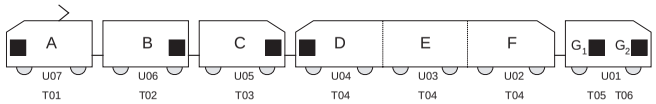
Step	Action	Expected result	Check	Expected configuration
			The results shall be checked by analyzing the NADI: UIC and TCN addresses shall be checked as (far as) specified, as well as leading requests and leading vehicles, other NADI contents shall be compared with the corresponding content of other nodes' NADIs. The reply telegram source function shall be always 15, the reply telegram status shall be always 0, and the reply telegram code shall be checked (0A01 for getNADI, 0A22 for getStatus, 0A03 for setOpMode, 0AF0 for setLeading / resetLeading). Empty entries mean: no check necessary. The entries "… relative to …" refer to NADI entries (counted from 1): 24,0 (global part); 71,0; 71,1 for the first vehicle and those for further vehicles. The entries 0-0 or 1-1 refer to a double node vehicle. The order is always starting at UIC address 1.	TCN addresses are apecified as far as predictable
1 *3,5b*	turn all vehicles on	TCN & UIC inauguration with all nodes	UIC addresses, 8 NADI entries	Standard addresses
2 *3,5b*	set B strong	TCN & UIC inauguration with all nodes	UIC and TCN addresses, 8 NADI entries TCN, relative to UIC: 0 vehicles, relative to TCN: 0, 1, 1, 0, 0, 0, 1-1 (UIC 1…7) vehicles, relative to UIC: 1, 0, 0, 1, 1, 1, 0-0 (UIC 1…7)	

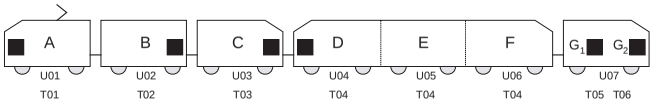
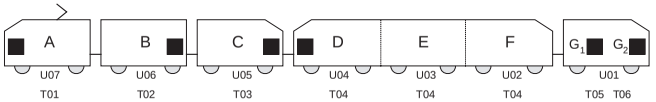
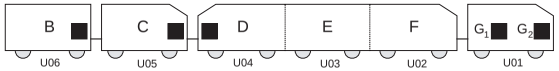
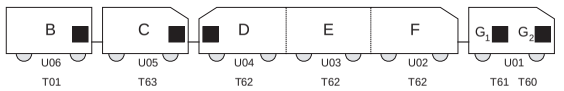
Step	Action	Expected result	Check	Expected configuration
3 *1b,* *3,5a*	turn A off	TCN & UIC inauguration with nodes $BCDG_1G_2$	UIC and TCN addresses, 7 NADI entries TCN, relative to UIC: 1 vehicles, relative to TCN: 1-1, 0, 0, 0, 1, 1 (UIC 1…6) vehicles, relative to UIC: 1-1, 0, 0, 0, 1, 1 (UIC 1…6)	
4 *2b,* *3,5b*	turn A on	TCN & UIC inauguration with all nodes	UIC and TCN addresses, 8 NADI entries TCN, relative to UIC: 0 vehicles, relative to TCN: 0, 1, 1, 0, 0, 0, 1-1 (UIC 1…7) vehicles, relative to UIC: 1, 0, 0, 1, 1, 1, 0-0 (UIC 1…7)	
5 *1a',* *3,5b*	turn B off	TCN & UIC inauguration with nodes $ACDG_1G_2$	UIC addresses, 7 NADI entries	
6 *2a,* *3,5b*	turn B on	TCN & UIC inauguration with all nodes	UIC addresses, 8 NADI entries	Standard addresses
7	set B strong and then set B weak			
8 *1a,* *3,5b*	turn D off	no inauguration	UIC and TCN addreses, 8 NADI entries TCN, relative to UIC: 0 vehicles, relative to TCN: 0, 1, 1, 0, 0, 0, 1-1 (UIC 1…7) vehicles, relative to UIC: 1, 0, 0, 1, 1, 1, 0-0 (UIC 1…7)	see above

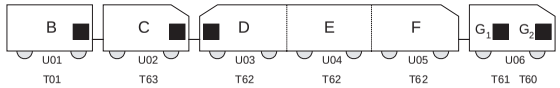
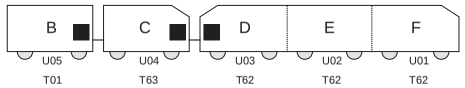
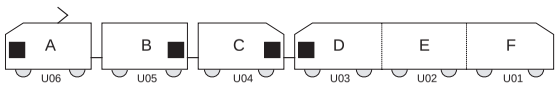
Step	Action	Expected result	Check	Expected configuration
9 *2a, 3,5b*	turn D on	UIC inauguration with all nodes	see above	see above
10 *1a, 3,5b*	turn CD off	no inauguration	see above	see above
11 *2a, 3,5b*	turn D on	TCN & UIC inauguration with nodes ABDG$_1$G$_2$	UIC addresses, 7 NADI entries	
12	set B strong and then set B weak			
13 *2a, 3,5b*	turn C on	TCN & UIC inauguration with all nodes	UIC addresses, 8 NADI entries	Standard addresses
14	set B strong and then set B weak			
15 *1b, 3,5b*	turn G$_2$ off	TCN & UIC inauguration with nodes ABCDG$_1$	UIC addresses, 7 NADI entries	
16 *2b, 3,5b*	turn G$_2$ on	TCN & UIC inauguration with all nodes	UIC addresses, 8 NADI entries	Standard addresses
17 *3,5b*	set A strong	[TCN] & UIC inauguration with all nodes	UIC and TCN addresses, 8 NADI entries TCN, relative to UIC: 1 vehicles, relative to TCN: 1, 0, 0, 1, 1, 1, 0-0 (UIC 1…7) vehicles, relative to UIC: 1, 0, 0, 1, 1, 1, 0-0 (UIC 1…7)	

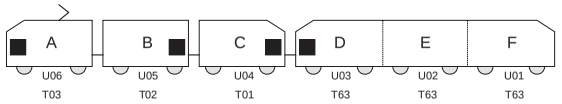
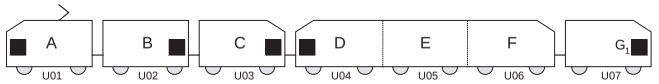
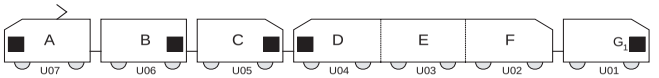
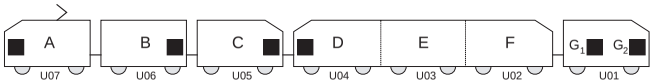
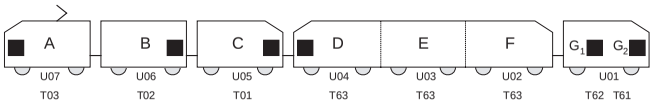
Step	Action	Expected result	Check	Expected configuration
18 *1b', 3,5a*	turn A off	TCN & UIC inauguration with nodes BCDG_1G_2	UIC address tendency corresponds with that of TCN, 7 NADI entries	for TCN master on D:  for TCN master on B, C, G_1 or G_2: 
19 *2b, 3,5b*	turn A on	TCN & UIC inauguration with all nodes	UIC addresses, 8 NADI entries	Standard addresses
20	set A strong and then set A weak			
21 *1a, 3,5b*	turn C off	no inauguration	UIC and TCN addresses, 8 NADI entries TCN, relative to UIC: 1 vehicles, relative to TCN: 1, 0, 0, 1, 1, 1, 0-0 (UIC 1…7) vehicles, relative to UIC: 1, 0, 0, 1, 1, 1, 0-0 (UIC 1…7)	see above
22 *2a, 3,5b*	turn C on	UIC inauguration with all nodes	see above	see above
23	set G_2 strong			

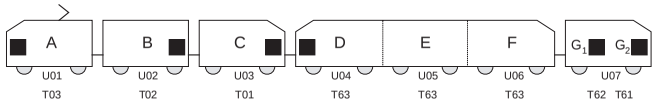
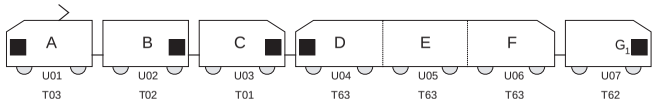
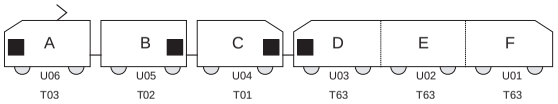
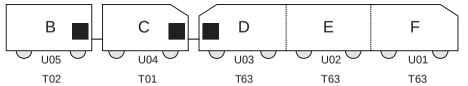
Step	Action	Expected result	Check	Expected configuration
24 *1b,* *3,5a*	turn A off	TCN & UIC inauguration with nodes BCDG$_1$G$_2$	UIC and TCN addresses, 7 NADI entries TCN, relative to UIC: 1 vehicles, relative to TCN: 1-1, 0, 0, 0, 1, 1 (UIC 1…6) vehicles, relative to UIC: 1-1, 0, 0, 0, 1, 1 (UIC 1…6)	
25 *2b,* *3,5b*	turn A on	TCN & UIC inauguration with all nodes	UIC and TCN addresses, 8 NADI entries TCN, relative to UIC: 0 vehicles, relative to TCN: 0, 1, 1, 0, 0, 0, 1-1 (UIC 1…7) vehicles, relative to UIC: 1, 0, 0, 1, 1, 1, 0-0 (UIC 1…7)	
26 *1a,* *3,5b*	turn G$_1$ off	no inauguration	see above	see above
27 *2a,* *3,5b*	turn G$_1$ on	UIC inauguration with all nodes	see above	see above
28 *1b',* *3,5b*	turn G$_2$ off	TCN & UIC inauguration with nodes ABCDG$_1$	UIC addresses, 7 NADI entries	
29 *2b,* *3,5b*	turn G$_2$ on	TCN & UIC inauguration with all nodes	UIC addresses 8 NADI entries	Standard addresses

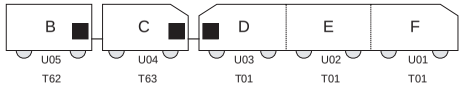
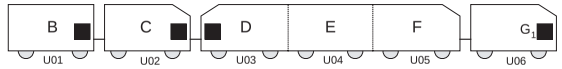
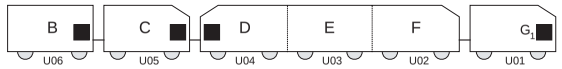
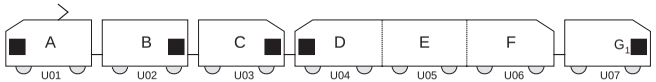
Step	Action	Expected result	Check	Expected configuration
30	set A strong and then set A weak	[TCN &] UIC inauguration with all nodes	UIC and TCN addresses, 8 NADI entries TCN, relative to UIC: 1 vehicles, relative to TCN: 1, 0, 0, 1, 1, 1, 0-0 (UIC 1…7) vehicles, relative to UIC: 1, 0, 0, 1, 1, 1, 0-0 (UIC 1…7)	
31 *4a, 5b*	leading request on A for direction 1	UIC inauguration with all nodes	UIC and TCN addresses, 8 NADI entries A requ. leadership and is leading	see above
32 *3, 5b*	reset leading request on A	UIC inauguration with all nodes	UIC and TCN addresses, 8 NADI entries	see above
33 *4c, 5b*	leading request on C for direction 2	UIC inauguration with all nodes	UIC and TCN addresses, 8 NADI entries C requ. leadership and is leading	see above
34 *3, 5b*	reset leading request on C	UIC inauguration with all nodes	UIC and TCN addresses, 8 NADI entries	see above
35 *4c, 5b*	leading request on F (direction 2)	UIC inauguration with all nodes	UIC and TCN addresses, 8 NADI entries F requ. leadership and is leading TCN, relative to UIC: 0 vehicles, relative to TCN: 0-0, 1, 1, 1, 0, 0, 1 (UIC 1…7) vehicles, relative to UIC: 1-1, 0, 0, 0, 1, 1, 0 (UIC 1…7)	

Step	Action	Expected result	Check	Expected configuration
36 *4b,* *5b*	leading request on D (direction 1)	UIC inauguration with all nodes	UIC and TCN addresses, 8 NADI entries D requ. leadership and is leading TCN, relative to UIC: 1 vehicles, relative to TCN: 1, 0, 0, 1, 1, 1, 0-0 (UIC 1…7) vehicles, relative to UIC: 1, 0, 0, 1, 1, 1, 0-0 (UIC 1…7)	
37 *3,* *5b*	reset leading request on D	UIC inauguration with all nodes	UIC and TCN addresses, 8 NADI entries	see above
38 *4a,* *5b*	leading request on G_2 for direction 2	UIC inauguration with all nodes	UIC and TCN addresses, 8 NADI entries G_2 requ. leadership and is leading TCN, relative to UIC: 0 vehicles, relative to TCN: 0-0, 1, 1, 1, 0, 0, 1 (UIC 1…7) vehicles, relative to UIC: 1-1, 0, 0, 0, 1, 1, 0 (UIC 1…7)	
39 *1b'* *4a,* *5a*	turn offA	TCN & UIC inauguration with nodes BCDG$_1$G$_2$	UIC addresses, 7 NADI entries G_2 requ. leadership and is leading	
40 *4a,* *5a*	set B strong	[TCN &] UIC inauguration with nodes BCDG$_1$G$_2$	UIC and TCN addresses, 7 NADI entries G_2 requ. leadership and is leading TCN, relative to UIC: 1 vehicles, relative to TCN: 1-1, 0, 0, 0, 1, 1 (UIC 1…6) vehicles, relative to UIC: 1-1, 0, 0, 0, 1, 1 (UIC 1…6)	

Step	Action	Expected result	Check	Expected configuration
41 *3,5a*	reset leading request on G_2	UIC inauguration with nodes BCDG_1G_2	UIC and TCN addresses, 7 NADI entries TCN, relative to UIC: 1 vehicles, relative to TCN: 1-1, 0, 0, 0, 1, 1 (UIC 1…6) vehicles, relative to UIC: 1-1, 0, 0, 0, 1, 1 (UIC 1…6)	see above
42 *4c,* *5a*	leading request on D (direction 1)	UIC inauguration with nodes BCDG_1G_2	UIC and TCN addresses, 7 NADI entries D requ. leadership and is leading TCN, relative to UIC: 0 vehicles, relative to TCN: 1, 1, 0, 0, 0, 1-1 (UIC 1…6) vehicles, relative to UIC: 0, 0, 1, 1, 1, 0-0 (UIC 1…6)	
43 *1ab* *4b,* *5b*	turn G_1G_2 off	TCN & UIC inagurationwith nodes BCD	UIC and TCN addresses, 5 NADI entries D requ. leadership and is leading TCN, relative to UIC: 1 vehicles, relative to TCN: 0, 0, 0, 1, 1 (UIC 1…5) vehicles, relative to UIC: 0, 0, 0, 1, 1 (UIC 1…5)	
44	set B weak			
45 *2b,* *4c,* *5c*	turn A on	TCN & UIC inauguration with nodes ABCD	UIC addresses, 6 NADI entries D requ. leadership and is leading	

Step	Action	Expected result	Check	Expected configuration
46 *4c,* *5c*	set C strong and then set C weak	[TCN &] 2 UIC inaugurations with nodes ABCD	UIC and TCN addresses, 6 NADI entries D requ. leadership and is leading TCN, relative to UIC: 1 vehicles, relative to TCN: 0, 0, 0, 1, 1, 0 (UIC 1…6) vehicles, relative to UIC: 0, 0, 0, 1, 1, 0 (UIC 1…6)	
47 *2b,* *4b,* *5b*	turn G_1 on	TCN & UIC inauguration with nodes ABCDG_1	TUIC address tendency corresponds with that of TCN, 7 NADI entries D requ. leadership and is leading	for TCN master on A or D:  for TCN master on B, C or G_1: 
48 *2b,* *4b,* *5b*	turn G_2 on	TCN & UIC inauguration with all nodes	UIC address tendency corresponds with that of TCN, 8 NADI entries D requ. leadership and is leading	for TCN master on A or D: standard addresses for TCN master on B, C, G_1 or G_2: 
49 *4b,* *5b*	set C strong	[TCN &] UIC inauguration with all nodes	UIC and TCN addresses, 8 NADI entries D requ. leadership and is leading TCN, relative to UIC: 1 vehicles, relative to TCN: 1-1, 0, 0, 0, 1, 1, 0 (UIC 1…7) vehicles, relative to UIC: 1-1, 0, 0, 0, 1, 1, 0 (UIC 1…7)	

Step	Action	Expected result	Check	Expected configuration
50 *3, 5b*	reset leading request on D	UIC inauguration with all nodes	UIC and TCN addresses, 8 NADI entries TCN, relative to UIC: 0 vehicles, relative to TCN: 0, 1, 1, 0, 0, 0, 1-1 (UIC 1…7) vehicles, relative to UIC: 1, 0, 0, 1, 1, 1, 0-0 (UIC 1…7)	
51 *1b, 3,5b*	turn G_2 off	TCN & UIC inauguration with nodes ABCDG_1	UIC and TCN addresses, 7 NADI entries TCN, relative to UIC: 0 vehicles, relative to TCN: 0, 1, 1, 0, 0, 0, 1 (UIC 1…7) vehicles, relative to UIC: 1, 0, 0, 1, 1, 1, 0 (UIC 1…7)	
52 *1b, 3,5c*	turn G_1 off	TCN & UIC inauguration with nodes ABCD	UIC and TCN addresses, 6 NADI entries TCN, relative to UIC: 1 vehicles, relative to TCN: 0, 0, 0, 1, 1, 0 (UIC 1…6) vehicles, relative to UIC: 0, 0, 0, 1, 1, 0 (UIC 1…6)	
53 *1b, 3,5b*	turn A off	TCN & UIC inauguration with nodes BCD	UIC and TCN addresses, 5 NADI entries TCN, relative to UIC: 1 vehicles, relative to TCN: 0, 0, 0, 1, 1 (UIC 1…5) vehicles, relative to UIC: 0, 0, 0, 1, 1 (UIC 1…5)	

Step	Action	Expected result	Check	Expected configuration
54 *3,* *5b*	set C weak; set D strong; set D weak	TCN & 3 UIC inauguration with nodes BCD	UIC and TCN addresses, 5 NADI entries TCN, relative to UIC: 0 vehicles, relative to TCN: 1, 1, 1, 0, 0 (UIC 1…5) vehicles, relative to UIC: 0, 0, 0, 1, 1 (UIC 1…5)	
55 *2b,* *3,5a*	turn G_1 on	TCN & UIC inauguration with nodes $BCDG_1$	UIC address tendency corresponds with that of TCN, 6 NADI entries	for TCN master on D:  for TCN master on B, C or G_1: 
56 *2b,* *3,5b*	turn A on	TCN & UIC inauguration with nodes $ABCDG_1$	UIC addresses, 7 NADI entries	
57 *2b,* *3,5b*	turn G_2 on	TCN & UIC inauguration with all nodes	UIC addresses, 8 NADI entries	standard adresses
58 *4c,* *5b*	leading request on C for direction 1	UIC inauguration with all nodes	UIC addresses, 8 NADI-entries C requ. leadership and is leading	standard addresses

H.6.3.4.2 - Unconfirmed compositions with trainsets of all possible lengths

Purpose:

Test the correct computation of unconfirmed train compositions depending on:

- the number of vehicles/nodes in the train;
- the number of vehicles in each trainset within the train;
- the position of the TCN master within the train;
- the existence and position of a leading vehicle.

Reference to UIC 556:

Point C.1.3.2 - page 39

Composition:

See Fig. 21 - page 185.

NB : Following cases shall be covered:

1. failure of single (a) intermediate / (b) end node(s) (master/slave nodes);
2. occurrence of single (a) intermediate / (b) end node(s)
3. no leading vehicle;
4. leading vehicle (a) in an end node, (b) in an intermediate node/vehicle;
5. (a) no / (b) one / (c) two traction unit(s) in end/intermediate node(s)/vehicle(s)

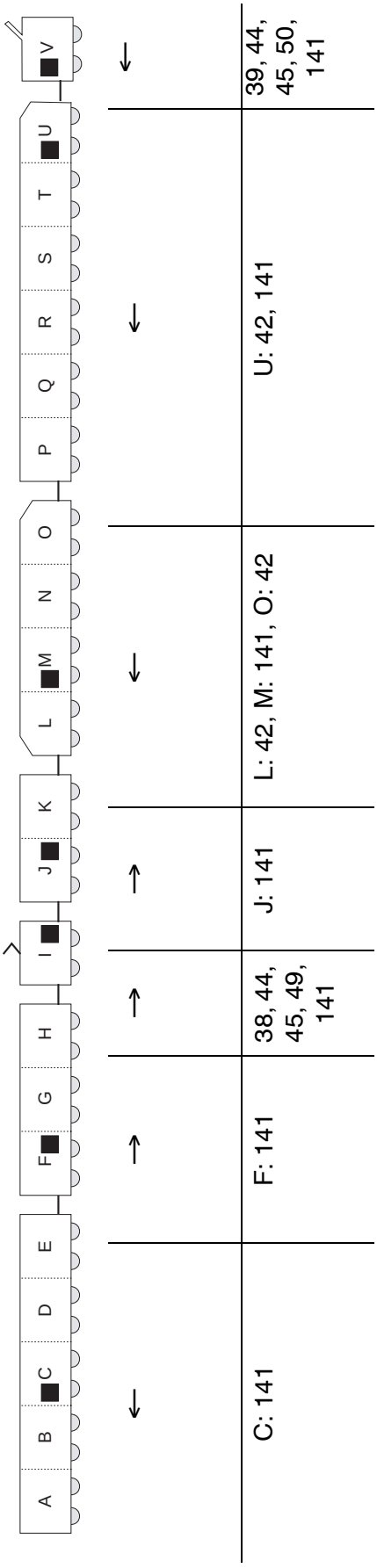


Fig. 21 - Long train with 16 and more vehicles

Test sequence:

Step	Action	Expected result	Check	Expected configuration
			The calculation shall be checked by analyzing the NADI. UIC addresses shall be checked as (far as) specified, other NADI contents shall be compared with the corresponding content of other nodes' NADIs.	Due to its length, the train is symbolized by a string, containing (subsets of) the vehicle letters (A to V) and after each vehicle letter the UIC address as far as predictable.
1 *3,5b*	turn IJ on	TCN & UIC inauguration with nodes IJ	UIC addresses 2 nodes, 3 vehicles, 3 NADI entries	I01 J02 K03
2 *2b, 3,5a*	turn F on	TCN & UIC inauguration with nodes FIJ	UIC address tendency corresponds wuth that of TCN 3 nodes, 6 vehicles, 6 NADI entries	F06 G05 H04 I03 J02 K01
3 *2b, 3,5a*	turn M on	TCN & UIC inauguration with nodes FIJM	UIC address tendency corresponds with that of TCN 4 nodes, 10 vehicles, 10 NADI entries	for TCN master on M: F01 G02 H03 I04 J05 K06 L07 M08 N09 O10 for TCN master on F, I or J: F10 G09 H08 I07 J06 K05 L04 M03 N02 O01
4 *1b, 3,5b*	turn F off	TCN & UIC inauguration with nodes IJM	UIC addresses 3 nodes, 7 vehicles, 7 NADI entries	I01 J02 K03 L04 M05 N06 O07
5 *4a, 5b*	leading request on O (direction 2)	UIC inauguration with nodes IJM	UIC addresses 3 nodes, 7 vehicles, 7 NADI entries leading vehicle: O	I07 J06 K05 L04 M03 N02 O01
6 *2b, 4b 5c*	turn UV on	TCN & UIC inauguration with nodes IJMUV	UIC addresses 5 nodes, 14 vehicles, 14 NADI entries leading vehicle: O	I01 J02 K03 L04 M05 N06 O07 P08 Q09 R10 S11 T12 U13 V14
7 *4b, 5c*	leading request on L (direction1)	UIC inauguration with nodes IJMUV	UIC addresses 5 nodes, 14 vehicles, 14 NADI entries leading vehicle: L	I01 J02 K03 L04 M05 N06 O07 P08 Q09 R10 S11 T12 U13 V14

Step	Action	Expected result	Check	Expected configuration
8 *2b, 4b, 5b*	turn F on	TCN & UIC inauguration with nodes FIJMUV	UIC addresses 6 nodes, 17 vehicles, 17 NADI entries leading vehicle: L	F01 G02 H03 I04 J05 K06 L07 M08 N09 O10 P11 Q12 R13 S14 T15 U16 V17
9 *2b, 4b, 5b*	turn C on	TCN & UIC inauguration with all nodes	UIC addresses 7 nodes, 22 vehicles, 22 NADI entries leading vehicle: L	A22 B21 C20 D19 E18 F17 G16 H15 I14 J13 K12 L11 M10 N09 O08 P07 Q06 R05 S04 T03 U02 V01
10 *3,5b*	reset leading request on L	UIC inauguration with all nodes	UIC addresses 7 nodes, 22 vehicles, 22 NADI entries	see above
11 *1b, 3,5a*	turn V off	TCN & UIC inauguration with nodes CFIJMU	TUIC address tendency corresponds with that of TCN 6 nodes, 21 vehicles, 21 NADI entries	for TCN master on C, M, U or V: A01 B02 C03 D04 E05 F06 G07 H08 I09 J10 K11 L12 M13 N14 O15 P16 Q17 R18 S19 T20 U21 for TCN master on F, I or J: A21 B20 C19 D18 E17 F16 G15 H14 I13 J12 K11 L10 M09 N08 O07 P06 Q05 R04 S03 T02 U01
12 *4a, 5a*	leading request on U (direction 2)	UIC inauguration with all nodes	UIC addresses 6 nodes, 21 vehicles, 21 NADI entries leading vehicle: U	A21 B20 C19 D18 E17 F16 G15 H14 I13 J12 K11 L10 M09 N08 O07 P06 Q05 R04 S03 T02 U01

Step	Action	Expected result	Check	Expected configuration
13 *1b,* *3,5a/1b,* *4a,* *5b*	uncouple between F and I	TCN & UIC inauguration with nodes CF and TCN & UIC inauguration with nodes IJMU	UIC addresses vehicle A … H: Tendency corresponding with that of TCN 2 nodes, 8 vehicles, 8 NADI entries vehicles I … U: 4 nodes, 13 vehicles, 13 NADI entries leading vehicle: U	vehicle A … H: for TCN master on C: A01 B02 C03 D04 E05 F06 G07 H08 for TCN-Master on F: A08 B07 C06 D05 E04 F03 G02 H01 vehicle I … U: I13 J12 K11 L10 M09 N08 O07 P06 Q05 R04 S03 T02 U01
14 *3,5a/3,5b*	reset leading request on U	UIC inauguration with nodes IJMU	UIC addresses vehicle I … U: 4 nodes, 13 vehicles, 13 NADI entries	vehicle. I … U: I01 J02 K03 L04 M05 N06 O07 P08 Q09 R10 S11 T12 U13
15 *3,5a/3,5b*	set I strong	TCN & UIC inauguration with nodes IJMU	UIC addresses vehicle I … U: 4 nodes, 13 vehicles, 13 NADI entries	see above
16 *3,5a/2b,* *3,5c*	turn V on	TCN & UIC inauguration with nodes IJMUV	UIC addresses vehicle I … V: 5 nodes, 14 vehicles, 14 NADI entries	vehicle I … V: for TCN master on M, U or V: I01 J02 K03 L04 M05 N06 O07 P08 Q09 R10 S11 T12 U13 V14 for TCN master on I or J: I14 J13 K12 L11 M10 N09 O08 P07 Q06 R05 S04 T03 U02 V01
17 *3,5a/3,5c*	set I weak	UIC inauguration with nodes IJMUV	UIC addresses vehicle. I … V: 5 nodes, 14 vehicles, 14 NADI entries	see above
18 *2b/3,5b*	couple between F and I	TCN & UIC inauguration with all nodes	UIC addresses	A22 B21 C20 D19 E18 F17 G16 H15 I14 J13 K12 L11 M10 N09 O08 P07 Q06 R05 S04 T03 U02 V01

Step	Action	Expected result	Check	Expected configuration
19 *1a, 3,5b,*	turn M off	TCN & UIC inauguration with nodes CFIJUV	UIC addresses	A18 B17 C16 D15 E14 F13 G12 H11 I10 J09 K08 P07 Q06 R05 S04 T03 U02 V01
20 *2a, 3,5b*	turn M on	TCN & UIC inauguration with all nodes	UIC addresses	A22 B21 C20 D19 E18 F17 G16 H15 I14 J13 K12 L11 M10 N09 O08 P07 Q06 R05 S04 T03 U02 V01

H.6.3.5 - Leading vehicle test suite

Purpose:

Test the correct handling of a leadership request on a node and of its withdrawal, including the detection and the correct handling of a request conflict.

The following table shows the necessary test cases to be covered by the test steps as well as the expected results.

Reference:

Point C.1.3.2 - page 39

Composition:

See reference composition.

NB : Following basic cases shall be covered (each case is carried out by a line or a group of lines within the table below):

1. no node is leading, and no node requests leadership;
2. no node is leading, and a single node requests leadership;
3. one node is leading and another node requests leadership (end node & end node, intermediate node & intermediate node, end node & intermediate node);
4. no node is leading and two nodes request leadership (end node & end node, intermediate node & intermediate node, end node & intermediate node);
5. no node is leading and nodes $ACDG_2$ request leadership;
6. two nodes were leading (end node & end node, intermediate node & intermediate node, end node & intermediate node).

The transitions between these cases shall cover the following situations:

A) no node is leading, at least two nodes request leadership and one of them resets its leadership request;

B) one node is leading, at least one other node requests leadership, and the leading node resets its leadership request;

C) one node is leading, at least one other node requests leadership, and a non-leading node resets its leadership request;

D) one node is leading, at least one other node requests leadership, and the leading node receives a leadership request for the opposite direction.

Line No	Vehicle A	Vehicle B	Vehicle C	Vehicle D	Vehicle F	Node G_1	Node G_2	leadership	Summary description
1	- [a]	-	-	-	-	-	-	none	no request
2a	requested	-	-	-	-	-	-	Vehicle A	no conflict
2b	-	-	requested	-	-	-	-	Vehicle C	no conflict
2c	-	-	-	requested	-	-	-	Vehicle D	no conflict
2d	-	-	-	-	requested	-	-	Vehicle F	no conflict
2e	-	-	-	-	-	-	requested	Vehicle G_2	no conflict
3a	requested & leading	-	requested	-	-	-	-	Vehicle A	with one leadership [b]
3b	requested & leading	-	-	-	-	-	requested	Vehicle A	with one leadership
3c	requested	-	requested & leading	-	-	-	-	Vehicle C	with one leadership
3d	-	-	requested & leading	-	requested	-	-	Vehicle C	with one leadership
3e	-	-	requested & leading	-	-	-	requested	Vehicle C	with one leadership
3f	requested	-	-	requested & leading	-	-	-	Vehicle D	with one leadership
3g	-	-	requested	requested & leading	-	-	-	Vehicle D	with one leadership
3h	-	-	-	requested & leading	-	-	requested	Vehicle D	with one leadership
3i	requested	-	-	-	requested & leading	-	-	Vehicle F	with one leadership
3j	-	-	requested	-	requested & leading	-	-	Vehicle F	with one leadership
3k	-	-	-	-	requested & leading	-	requested	Vehicle F	with one leadership
3l	requested	-	-	-	-	-	requested & leading	Vehicle G_2	with one leadership
3m	-	-	requested	-	-	-	requested& leading	Vehicle G_2	with one leadership

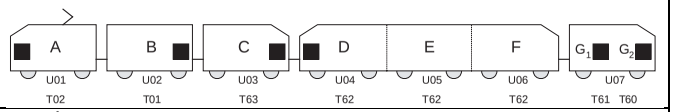
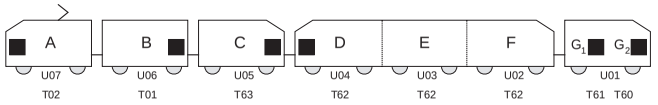
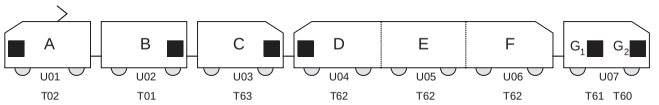
a. no request.
b. shaded field indicate conflict cases.

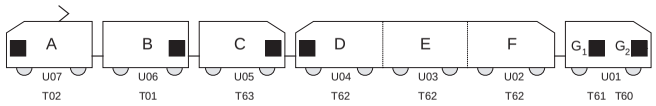
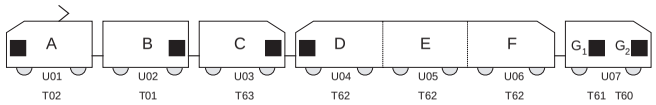
Line No	Vehicle A	Vehicle B	Vehicle C	Vehicle D	Vehicle F	Node G_1	Node G_2	leadership	Summary description
4a	requested	-	-	-	-	-	requested	none	without leadership
4b	-	-	requested	requested	-	-	-	none	without leadership
4c	requested	-	requested	-	-	-	-	none	without leadership
4d	-	-	requested	-	requested	-	-	none	without leadership
4e	-	-	requested	-	-	-	requested	none	without leadership
4f	requested	-	-	requested	-	-	-	none	without leadership
4g	-	-	-	requested	-	-	requested	none	without leadership
4h	requested	-	-	-	requested	-	-	none	without leadership
4i	-	-	-	-	requested	-	requested	none	without leadership
5a	requested	-	requested	requested	-	-	requested	none	without leadership
5b	requested	-	requested	-	requested	-	requested	none	without leadership
6a	requested & leading	-	-	-	-	-	requested & leading	Vehicle A	with multiple leadership
6b	-	-	requested & leading	requested & leading	-	-	-	Vehicle B	with multiple leadership
6c	requested & leading	-	requested & leading	-	-	-	-	VehicleC	with multiple leadership
6d	-	-	requested & leading	-	requested & leading	-	-	Vehicle C	with multiple leadership
6e	-	-	requested & leading	-	-	-	requested & leading	Vehicle C	with multiple leadership
6f	requested & leading	-	-	requested & leading	-	-	-	Vehicle D	with multiple leadership
6g	requested & leading	-	-	-	requested & leading	-	-	Vehicle F	with multiple leadership
6h	-	-	-	-	requested & leading	-	requested & leading	Vehicle F	with multiple leadership

General remark:

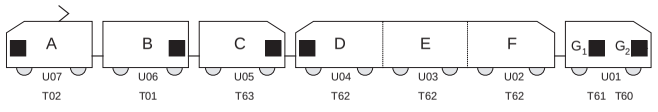
After each step the leading vehicle(s) shall be determined or it shall be detected that there is none. This shall be done by the corresponding NADI entries (dynamic property 511: "requests leading", dynamic property 512: "is leading"), through the distributed inauguration frames (dynamic property 510: "was leading", dynamic property 511: "requests leading"), and through the type of R telegrams distributed by the different nodes ("is leading"). After each step the number of nodes shall also be checked in the content of the topography response telegram (0A04, and this code itself too) and the number of vehicles through the NADI content (0A01, ditto). The reply telegram code shall also be checked for setOpMode (0A03) and multicastRequest (FA07). The reply telegram source function shall always be 15, the reply telegram status shall always be 0.

Test sequence:

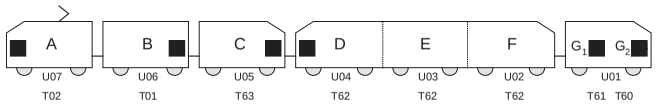
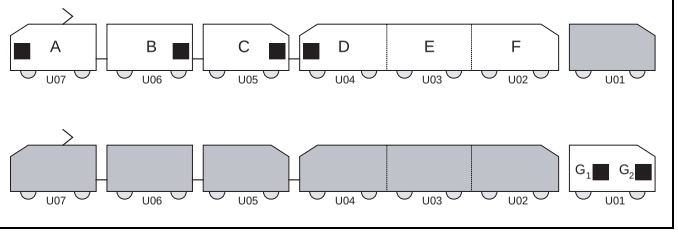
Step & ref.	Action	Expected result	Check			Expected configuration
			All NADIs are unconfirmed and shall contain 8 entries. Empty entries mean: no check necesary			TCN addresses as specified as far as predictable
1 *1*	turn on all vehicles	TCN & UIC inauguration with all nodes	510: -	511: -	512: -	Standard addresses
2 *1*	set B strong and then set B weak	[TCN &] 2 UIC inaugurations with all nodes	510: -	511: -	512: -	
3 *2a*	leading request on A for direction 1	UIC inauguration with all nodes	510: -	511: A	512:A	see above
4 *3b*	leading request on G_2 for direction 1	UIC inauguration with all nodes	510: A	511: A, G_2	512: A	see above
5 *2e, B*	reset leading request on A	UIC inauguration with all nodes	510: -	511:G_2	512: G_2	
6 *3l*	leading request on A for direction 1	UIC inauguration with all nodes	510: G_2	511: A, G_2	512:G_2	see above
7	leading request on C for direction 1	UIC inauguration with all nodes	510: G_2	511: A, C, G_2	512: G_2	see above
8 *3m, C*	reset leading request on A	UIC inauguration with all nodes	510: G_2	511: C, G_2	512:G_2	see above
9 *2b, B*	reset leading request on G_2	UIC inauguration with all nodes	510: -	511: C	512: C	
10 *3c*	leading request on A for direction 1	UIC inauguration with all nodes	510: C	511: A, C	512: C	see above
11	leading request on F for direction 2	UIC inauguration with all nodes	510: C	511: A, C, F	512: C	see above
12 *3d, C*	reset leading request on A	UIC inauguration with all nodes	510: C	510: C, F	512: C	see above

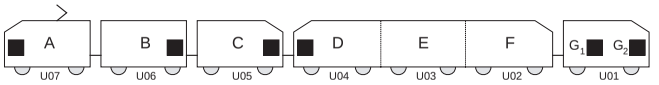
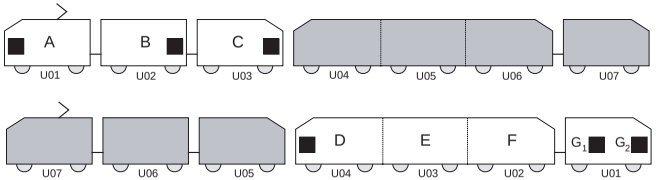
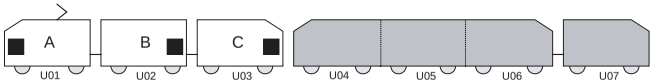
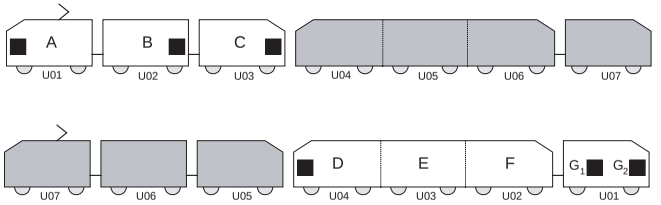
Step &ref.	Action	Expected result	Check			Expected configuration
13 *2d, B*	reset leading request on C	UIC inauguration with all nodes	510: -	511:F	512: F	
14 *3i*	leading request on A for direction 1	UIC inauguration with all nodes	510:F	511: A, F	512: F	see above
15 *3f, D*	leading request on D (direction 1)	UIC inauguration with all nodes	510: D	511: A, D	512: D	see above
16 *2c, C*	reset leading request on A	UIC inauguration with all nodes	510: D	511: D	512:D	see above
17 *3g*	leading request on G_2 for direction 1	UIC inauguration with all nodes	510: D	511: C, D	512: D	see above
18	leading request on G_2 for direction 1	UIC inauguration with all nodes	510: D	511:C, D, G_2	512: D	see above
19 *3h, C*	reset leading request on C	UIC inauguration with all nodes	510: D	511: D, G_2	512:D	see above
20 *3k, D*	leading request on F (direction 2)	UIC inauguration with all nodes	510: F	511: F, G_2	512: F	see above
21	leading request on C for direction 1	UIC inauguration with all nodes	510: F	511: C, F, G_2	512:F	see above
22 *3j, C*	reset leading request on G_2	UIC inauguration with all nodes	510: F	511: C, F	512: F	see above
23 *2b, B*	reset leading request on F	UIC inauguration with all nodes	510: -	511: C	512: C	
24 *3e*	leading request on G_2 for direction 1	UIC inauguration with all nodes	510: C	511: C, G_2	512: C	see above
25	leading request on A for direction 1	UIC inauguration with all nodes	510: C	511: A, C, G_2	512: C	see above

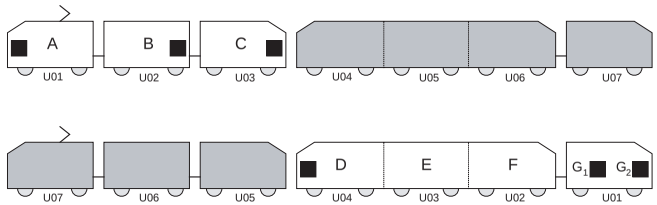
Step &ref.	Action	Expected result	Check			Expected configuration
26 *4a, B*	reset leading request on C	UIC inauguration with all nodes	510: -	511: A, G_2	512: -	see above
27	leading request on C for direction 1	UIC inauguration with all nodes	510: -	511: A, C, G_2	512: -	se above
28 *5a*	leading request on D (direction 1)	UIC inauguration with all nodes	510: -	511: A, C, D, G_2	512: -	see above
29 *5b*	leading request on F (direction 2)	UIC inauguration with all nodes	510: -	511: A, C, F, G_2	512:-	see above
30 *A*	reset leading request on G_2	UIC inauguration with all nodes	510: -	511: A, C, F	512: -	see above
31 *4c, A*	reset leading request on F	UIC inauguration with all nodes	510: -	511: A, C	512: -	see above
32	leading request on F (direction 2)	UIC inauguration with all nodes	510: -	511: A, C, F	512:-	see above
33 *4d, A*	reset leading request on A	UIC inauguration with all nodes	510: -	511: C, F	512: -	see above
34 *4b*	leading request on D (direction 1)	UIC inauguration with all nodes	510: -	511: C, D	512:-	see above
35	leading request on G_2 for direction 1	UIC inauguration with all nodes	510: -	511: C, D, G_2	512: -	see above
36 *4e, A*	reset leading request on D	UIC inauguration with all nodes	510: -	511: C, G_2	512: -	see above
37	leading request on F (direction 2)	UIC inauguration with all nodes	510: -	511: C, F, G_2	512: -	see above
38 *4j, A*	reset leading request on C	UIC inauguration with all nodes	510: -	510: F, G_2	512: -	see above

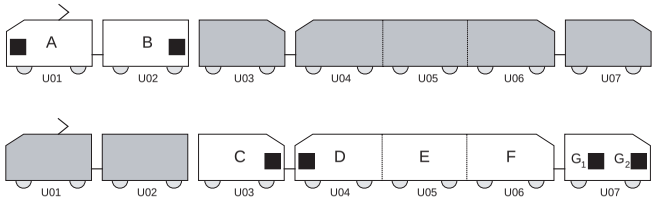
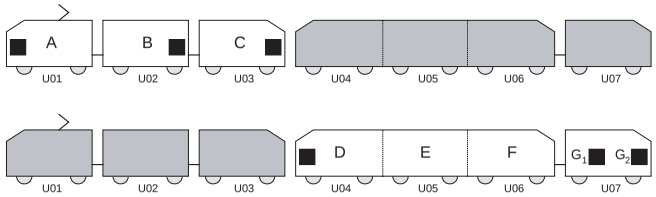
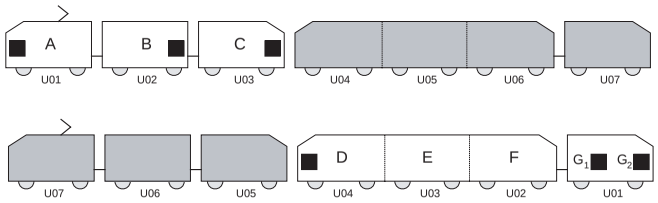
Step & ref	Action	Expected result	Check	Expected configuration
39 *4g*	leading request on D (direction 1)	UIC inauguration with all nodes	510: - 511: D, G_2 512: -	see above
40 *2c, A*	reset leading request on G_2	UIC inauguration with all nodes	510: - 511: D 512: D	
41	leading request on F (direction 2)			
42	leading request on G_2 for direction 1			

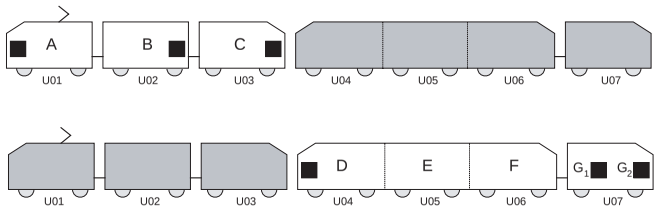
Continuation for confirmed confguration:

Step	Action	Expected result	Check	Expected configuration
			The specified results are valid for confirmed configurations and - if specified - for unknown (actual) configurations only. If not explicitly specified differently, the NADI(s) is/are confirmed.	If a cell contains two train segments, the upper one represents the NADI of the left train segment, and the lower one represents the NADI of the right train segment. If one segment has not been changed, only the other one is shown.
43c	confirm configuration	UIC inauguration with all nodes	510: F 511: F, G_2 512: F	
44c *B, C*	uncouple between F and G	TCN & UIC inauguration with nodes ABCD and TCN & UIC inauguration with nodes G_1G_2	left: 510: F 511: F 512: F right: 510: - 511: G_2 512: G_2 number of NADI entries: 7 (leflt) and 8 (right)	

Step	Action	Expected result	Check	Expected configuration
45c *6h*	Couple whole train	TCN & UIC inagguration with all nodes	510: F, G_2 511:F, G_2 512: - number of NADI entries: 8	Standard addresses
46c *2d, A*	reset leading request on G_2	UIC inauguration with all nodes	510: - 511: F 512: F number of NADI entries: 8	
47c	uncouple between C and D	TCN & UIC inauguration with nodes ABC and TCN & UIC inauguration with nodes DG_1G_2	left: 510: - 511: - 512: - right: 510: F 511: F 512: F number of NADI entries: 7 (left) and 8 (right)	
48c	leading request on C for direction 1	UIC inauguration with nodes ABC	left: 510: - 511: C, 512: C number of NADI entries: 7 (left)	left train segment: 
49c *6d*	couple whole train	TCN & UIC inauguration with all nodes	510: C, F 511: C, F 512:- number of actual NADI entries: 8	Standard addresses
50c *4d*	confirm configuration	UIC inauguration with all nodes	510:- 511: C, F 512:- number of NADI entries: 8	Standard addresses
51c *2b, A*	reset leading request on F	UIC inauguration with all nodes	510:- 511: C 512: C number of NADI entries: 8	Standard addresses
52c *3e*	leading request on G_2 for direction 1	UIC inauguration with all nodes	510: C 511:C, G_2 512: C number of NADI entries: 8	Standard addresses
53c *B, C*	uncouple between C and D	TCN & UIC inauguration with nodes ABC and TCN & UIC inauguration with nodes DG_1G_2	left: 510: C 511: C 512: C right: 510:- 511: G_2 512: G_2 number of NADI entries: 7 (lefIt) and 8 (right)	

Step	Action	Expected result	Check	Expected configuration
54c *6e*	couple whole train	TCN & UIC inauguration with all nodes	510: C, G_2 511: C, G_2 512:- number of actual NADI entries: 8	Standard addresses
55c *4e*	confirm configuration	UIC inauguraiton with all nodes	510:- 511: C, G_2 512:- number of NADI entries: 8	Standard addresses
56c	leading request on A for direction 1	UIC inauguration with all nodes	510:- 511: A, C, G_2 512:- number of NADI entries: 8	Standard addresses
57c *A*	uncouple between C and D	TCN & UIC inauguration with nodes ABC and TCN & UIC inauguration with nodes DG_1 G_2	left: 510:- 511: A, C 512:- right: 510:- 511: G_2 512: G_2 number of NADI entries: 7 (left) and 8 (right)	
58c *A*	reset leading request on C	UIC inauguration with nodes ABC	left: 510: - 511: A 512: A number of NADI entries: 7 (left)	see above
59c *6a*	couple whole train	TCN & UIC inauguration with all nodes	510: A, G_2 511: A, G_2 512: - number of actual NADI entries: 8	Standard addresses
60c *3b*	confirm configuration	UIC inauguration with all nodes	510: - 511: A, G_2 512: - number of NADI entries: 8	Standard addresses
61c *2a,* *A*	reset leading request on G_2	UIC inauguration with all nodes	510:- 511: A 512: A number of NADI entries: 8	Standard addresses
62c *3a*	leading request on C for direction 1	UIC inauguration with all nodes	510: A 511: A, C 512: A number of NADI entries: 8	Standard addresses

Step	Action	Expected result	Check	Expected configuration
63c *B, C*	uncouple between B and C	TCN & UIC inauguration with nodes AB and TCN & UIC inauguration with nodes CDG_1G_2	left: 510: A 511: A 512: A right: 510: - 511: C 512: C number of NADI entries: 7 (left) and 8 (right)	
64c *6c*	couple whole train	TCN & UIC inauguration with all nodes	510: A, C 511: A, C 512:- number of NADI entries: 8	Standard addresses
65c	leading request on D (direction 1)	UIC inauguration with all nodes	510:- 511: A, C, D 512:- number of NADI entries: 8	Standard addresses
66c *4f, A*	reset leading request on C	UIC inauguration with all nodes	510:- 511: A, D 512:- number of NADI entries: 8	Standard addresses
67c *A*	uncouple between C and D	TCN & UIC inauguration with nodes ABC and TCN & UIC inauguration with nodes DG_1 G_2	left: 510:- 511: A 512: A right: 510:- 511: D 512: D number of NADI entries: 7 (left) and 8 (right)	
68c *6f*	couple whole train	TCN & UIC inauguration with all nodes	510: A, D 511: A, D 512:- number of NADI entries: 8	Standard addresses
69c *4h*	leading request on F (direction 2)	UIC inauguration with all nodes	510:- 511: A, F 512:- number of NADI entries: 8	Standard addresses
70c *A*	uncouple between C and D	TCN & UIC inauguration with nodes ABC and TCN & UIC inauguration with nodes DG_1 G_2	left: 510:- 511: A 512: A right: 510:- 511: F 512: F number of NADI entries: 7 (left) and 8 (right)	left train segment 

Step	Action	Expected result	Check	Expected configuration
71c *6g*	couple whole train	TCN & UIC inauguration with all nodes	510: A, F 511: A, F 512:- number of actual NADI entries: 8	Standard addresses
72c *4h*	confirm configuration	UIC inauguration with all nodes	510:- 511: A, F 512:- number of NADI entries: 8	Standard addresses
73c *4f*	leading request on D (direction1)	UIC inauguration with all nodes	510:- 511: A, D 512:- number of NADI entries: 8	Standard addresses
74c	leading request on C for direction 1	UIC inauguration with all nodes	510:- 511: A, C, D 512:- number of NADI entries: 8	Standard addresses
75c *A*	uncouple between C and D	TCN & UIC inauguration with nodes ABC and TCN & UIC inauguraiton with nodes DG_1G_2	left: 510:- 511: A, C 512:- right: 510:- 511: D 512: D number of NADI entries: 7 (left) and 8 (right)	
76c *A*	reset leading request on A	UIC inauguration with nodes ABC	left: 510: - 511: C 512: C number of NADI entries: 7 (left)	see above
77c *6b*	couple whole train	TCN & UIC inauguration with all nodes	510: C, D 511: C, D 512:- number of NADI entries: 8	Standard addrsses

H.6.3.6 - Inhibit functionality test suite

Purpose:

Test the correct handling of an inhibit request on a node and its withdrawal.

Reference:

Appendix A - page 35

Composition:

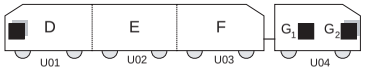
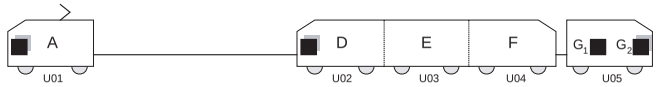
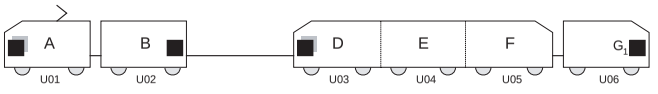
See reference composition.

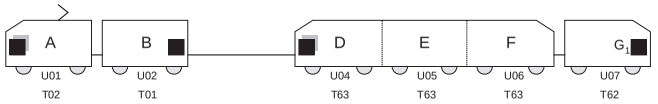
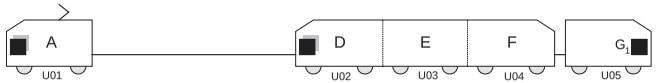
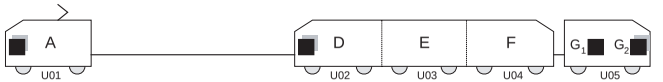
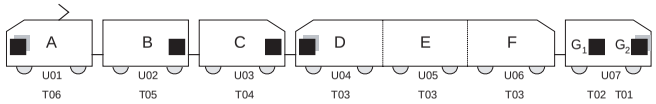
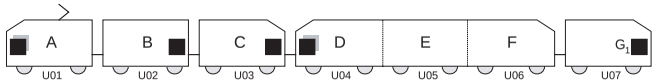
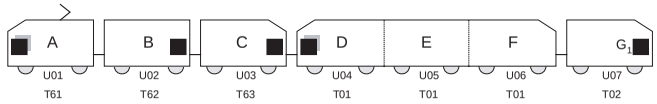
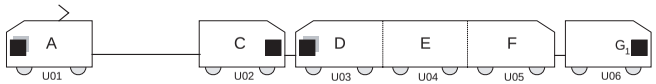
NB : Following basic cases shall be covered:

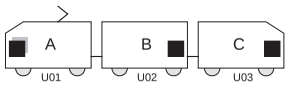
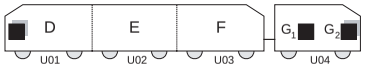
	Test case	*Reference to test steps*
1.	non-master end node failure	4, 8 [a]
2.	non-master intermediate node failure	3, 17
3.	master end node failure	8 [a], 15
4.	master intermediate node failure	10
5.	occurrence of end node(s)	11, 20
6.	occurrence of intermediate node(s)	7, 18

a. In case 8 it is not predictable if node G_2 is master or not. So, this step may be test case 1 or 3.

Test sequence:

Step	Action	Expected result	Check	Expected configuration
			No vehicle is leading, the configuration is unconfirmed. Checks will be done on the NADI. The inhibit bit shall be checked within the mapping server status reply telegram from the TCN master. The reply telegram source function shall always be 15, the reply telegram status shall always be 0, and the reply telegram code shall be checked (0A01 for getNADI, 0A22 for getStatus, 0A03 for setOpMode, FA05 for set inhibit / cancel inhibit). Empty entries mean: no check necessary.	
1	turn on all nodes	TCN & UIC inauguration with all nodes	NADI contains 8 entries no inhibit bit set	Standard addresses
2	set D strong; set D weak; send inhibit request to G_1			
3	turn B, C off	no inauguration	NADI contains 8 entries inhibit bit set	Standard addresses
4	turn A off	TCN & UIC inauguration with nodes DG_1G_2	NADI contains 5 entries no inhibit bit set	
5	turn A on	TCN & UIC inauguration with nodes $ADG_1\ G_2$	NADI contains 6 entries no inhibit bit set	
6	send inhibit request to D	no inauguration	NADI contains 6 entries no inhibit bit set	see above
7	turn B on	no inauguration	NADI contains 6 entries no inhibit bit set	see above
8	turn G_2 off	TCN & UIC inauguration with nodes $ABDG_1$	NADI contains 6 entries no inhibit bit set	

Step	Action	Expected result	Check	Expected configuration
9	set B strong and then set B weak	[TCN &] 2 UIC inaugurations with nodes ABDG$_1$	NADI contains 6 entries no inhibit bit set	
10	send inhibit request to A; turn B off	TCN & UIC inauguration with nodes ADG$_1$	NADI contains 5 entries no inhibit bit set	
11	send inhibit request to A; turn G$_2$ on	no inauguration	NADI contains 5 entries no inhibit bit set	see above
12	reset inhibit request on A	TCN & UIC inauguration with nodes ADG$_1$ G$_2$	NADI contains 6 entries no inhibit bit set	
13	turn all nodes on			
14	set G$_2$ strong ; set G$_2$ weak ; send inhibit request to B	[TCN &] 2 UIC inaugurations with all nodes	NADI contains 8 entries inhibit bit set	
15	turn G$_2$ off	TCN & UIC inauguration with nodes ABCDG$_1$	NADI contains 7 entries no inhibit bit set	
16	set D strong; set D weak	[TCN &] 2 UIC inaugurations with nodes ABCDG$_1$	NADI contains 7 entries no inhibit bit set	
17	send inhibit request to A turn B, C off	no inauguration	NADI contains 7 entries no inhibit bit set	see above
18	turn C on	TCN & UIC inauguration with nodes ACDG$_1$	NADI contains 6 entries no inhibit bit set	
19	turn on all nodes and uncouple between C and D			

Step	Action	Expected result	Check	Expected configuration
20	send inhibit request to B and couple whole train	no inauguration	NADI contains 3 entries (left) and 5 entries (right) inhibit bit set (left)	left train segment:  right train segment: 
21	reset inhibit rquest on B	TCN & UIC inauguration with all nodes	NADI contains 8 entries no inhibit bit set	Standard addresses

H.6.3.7 - Group handling test suite

Purpose:

Test the correct administration of read, write and delete group telegrams for single groups and for lists of groups.

Reference:

Appendix A - page 35.

Composition:

See reference composition.

NB : Following basic cases shall be covered:

	Test case	*Reference to test steps*
1.	write single group to one node	1
2.	write single group to all nodes (multicast)	15
3.	write list of groups to one node	2, 4
4.	write list of groups to all nodes (multicast)	3
5.	read single group (from one node)	5
6.	read group list (from one node)	8
7.	delete single group in one node	11
8.	delete all groups in one node	10
9.	delete all groups in all nodes (multicast)	16
10.	write single existing group to one node	9
11.	write list of groups with existing groups to one node	12
12.	write single illegal group number	13, 14
13.	read non-existing group	6
14.	delete non-existing group	7

Test sequence:

Step	Action	Expected result
	The syntax of a group in this column is: <group number> '('<list of UIC ids>')' The group identifier (32 unicode characters) shall be initialized with blanks and shall start with the decimal representation of the group number	The only relevant result is the group directory of the specified nodes. The result shall be checked by means of the readAllGroups telegram. The group identifiers shall always be set as defined on the left. The reply telegram source function shall always be 15, the reply telegram status shall always be 0, and the reply telegram code shall be checked (BA01 for readGroup, BA02 for readAllGroups, BA03 for writeGroup, BA04 for writeGroupList, BA05 for deleteGroup, BA06 for deleteAllGroups, and FA07 for multiCastRequest).
1	write group 201 (24001, 24002, 24003, …, 24022) to vehicle A	**node A:** 201 (24001, 24002, 24003, …, 24022)
2	write group list: 240 (24001, 24002, 24003, …, 24022), 241 (24101, 24102, 24103, …, 24122), 242 (24201, 24202, 24203, …, 24222), 243 (24301, 24302, 24303, …, 24322), 244 (24401, 24402, 24403, …, 24422), 245 (24501, 24502, 24503, …, 24522), 246 (24601, 24602, 24603, …, 24622), 247 (24701, 24702, 24703, …, 24722), 248 (24801, 24802, 24803, …, 24822), 249 (24901, 24902, 24903, …, 24922), 250 (25001, 25002, 25003, …, 25022), 251 (25101, 25102, 25103, …, 25122), 252 (25201, 25202, 25203, …, 25222), 253 (25301, 25302, 25303, …, 25322), 254 (25401, 25402, 25403, …, 25422) to vehicle B	**node A:** 201 (24001, 24002, 24003, …, 24022) **node B:** 240 (24001, 24002, 24003, …, 24022), 241 (24101, 24102, 24103, …, 24122), 242 (24201, 24202, 24203, …, 24222), 243 (24301, 24302, 24303, …, 24322), 244 (24401, 24402, 24403, …, 24422), 245 (24501, 24502, 24503, …, 24522), 246 (24601, 24602, 24603, …, 24622), 247 (24701, 24702, 24703, …, 24722), 248 (24801, 24802, 24803, …, 24822), 249 (24901, 24902, 24903, …, 24922), 250 (25001, 25002, 25003, …, 25022), 251 (25101, 25102, 25103, …, 25122), 252 (25201, 25202, 25203, …, 25222), 253 (25301, 25302, 25303, …, 25322), 254 (25401, 25402, 25403, …, 25422)

Step	Action	Expected result
3	write group list: 220 (22001, 22002, 22003, …, 22022), 221 (22101, 22102, 22103, …, 22122), 222 (22201, 22202, 22203, …, 22222), 223 (22301, 22302, 22303, …, 22322), 224 (22401, 22402, 22403, …, 22422), 225 (22501, 22502, 22503, …, 22522), 226 (22601, 22602, 22603, …, 22622), 227 (22701, 22702, 22703, …, 22722), 228 (22801, 22802, 22803, …, 22822), 229 (22901, 22902, 22903, …, 22922), 230 (23001, 23002, 23003, …, 23022), 231 (23101, 23102, 23103, …, 23122), 232 (23201, 23202, 23203, …, 23222), 233 (23301, 23302, 23303, …, 23322), 234 (23401, 23402, 23403, …, 23422) to vehicle A as multicast telegram	**node A:** 201 (24001, 24002, 24003, …, 24022), 220 (22001, 22002, 22003, …, 22022), 221 (22101, 22102, 22103, …, 22122), 222 (22201, 22202, 22203, …, 22222), 223 (22301, 22302, 22303, …, 22322), 224 (22401, 22402, 22403, …, 22422), 225 (22501, 22502, 22503, …, 22522), 226 (22601, 22602, 22603, …, 22622), 227 (22701, 22702, 22703, …, 22722), 228 (22801, 22802, 22803, …, 22822), 229 (22901, 22902, 22903, …, 22922), 230 (23001, 23002, 23003, …, 23022), 231 (23101, 23102, 23103, …, 23122), 232 (23201, 23202, 23203, …, 23222), 233 (23301, 23302, 23303, …, 23322), 234 (23401, 23402, 23403, …, 23422) **node B:** 220 (22001, 22002, 22003, …, 22022), 221 (22101, 22102, 22103, …, 22122), 222 (22201, 22202, 22203, …, 22222), 223 (22301, 22302, 22303, …, 22322), 224 (22401, 22402, 22403, …, 22422), 225 (22501, 22502, 22503, …, 22522), 226 (22601, 22602, 22603, …, 22622), 227 (22701, 22702, 22703, …, 22722), 228 (22801, 22802, 22803, …, 22822), 229 (22901, 22902, 22903, …, 22922), 230 (23001, 23002, 23003, …, 23022), 231 (23101, 23102, 23103, …, 23122), 232 (23201, 23202, 23203, …, 23222), 233 (23301, 23302, 23303, …, 23322), 234 (23401, 23402, 23403, …, 23422), 240 (24001, 24002, 24003, …, 24022), 241 (24101, 24102, 24103, …, 24122), 242 (24201, 24202, 24203, …, 24222), 243 (24301, 24302, 24303, …, 24322), 244 (24401, 24402, 24403, …, 24422), 245 (24501, 24502, 24503, …, 24522), 246 (24601, 24602, 24603, …, 24622), 247 (24701, 24702, 24703, …, 24722), 248 (24801, 24802, 24803, …, 24822), 249 (24901, 24902, 24903, …, 24922), 250 (25001, 25002, 25003, …, 25022), 251 (25101, 25102, 25103, …, 25122), 252 (25201, 25202, 25203, …, 25222), 253 (25301, 25302, 25303, …, 25322), 254 (25401, 25402, 25403, …, 25422) **nodes C, D, G_1, G_2:** 220 (22001, 22002, 22003, …, 22022), 221 (22101, 22102, 22103, …, 22122), 222 (22201, 22202, 22203, …, 22222), 223 (22301, 22302, 22303, …, 22322), 224 (22401, 22402, 22403, …, 22422), 225 (22501, 22502, 22503, …, 22522), 226 (22601, 22602, 22603, …, 22622), 227 (22701, 22702, 22703, …, 22722), 228 (22801, 22802, 22803, …, 22822), 229 (22901, 22902, 22903, …, 22922), 230 (23001, 23002, 23003, …, 23022), 231 (23101, 23102, 23103, …, 23122), 232 (23201, 23202, 23203, …, 23222), 233 (23301, 23302, 23303, …, 23322), 234 (23401, 23402, 23403, …, 23422)

Step	Action	Expected result
4	write group list:: 201 (20101, 20102, 20103, …, 20122), 202 (20201, 20202, 20203, …, 20222), 203 (20301, 20302, 20303, …, 20322), 202 (20401, 20402, 20403, …, 20422), 205 (20501, 20502, 20503, …, 20522), 206 (20601, 20602, 20603, …, 20622), 207 (20701, 20702, 20703, …, 20722), 208 (20801, 20802, 20803, …, 20822), 209 (20901, 20902, 20903, …, 20922), 210 (21001, 21002, 21003, …, 21022), 211 (21101, 21102, 21103, …, 21122), 212 (21201, 21202, 21203, …, 21222), 213 (21301, 21302, 21303, …, 21322), 214 (21401, 21402, 21403, …, 21422) to vehicle B write group list:: 215 (21501, 21502, 21503, …, 21522), 216 (21601, 21602, 21603, …, 21622), 217 (21701, 21702, 21703, …, 21722), 218 (21801, 21802, 21803, …, 21822), 219 (21901, 21902, 21903, …, 21922), 235 (23501, 23502, 23503, …, 23522), 236 (23601, 23602, 23603, …, 23622), 237 (23701, 23702, 23703, …, 23722), 238 (23801, 23802, 23803, …, 23822), 239 (23901, 23902, 23903, …, 23922) to vehicle B	**node A:** 201 (24001, 24002, 24003, …, 24022), 220 (22001, 22002, 22003, …, 22022), 221 (22101, 22102, 22103, …, 22122), 222 (22201, 22202, 22203, …, 22222), 223 (22301, 22302, 22303, …, 22322), 224 (22401, 22402, 22403, …, 22422), 225 (22501, 22502, 22503, …, 22522), 226 (22601, 22602, 22603, …, 22622), 227 (22701, 22702, 22703, …, 22722), 228 (22801, 22802, 22803, …, 22822), 229 (22901, 22902, 22903, …, 22922), 230 (23001, 23002, 23003, …, 23022), 231 (23101, 23102, 23103, …, 23122), 232 (23201, 23202, 23203, …, 23222), 233 (23301, 23302, 23303, …, 23322), 234 (23401, 23402, 23403, …, 23422) **node B:** 201 (20101, 20102, 20103, …, 20122), 202 (20201, 20202, 20203, …, 20222), 203 (20301, 20302, 20303, …, 20322), 202 (20401, 20402, 20403, …, 20422), 205 (20501, 20502, 20503, …, 20522), 206 (20601, 20602, 20603, …, 20622), 207 (20701, 20702, 20703, …, 20722), 208 (20801, 20802, 20803, …, 20822), 209 (20901, 20902, 20903, …, 20922), 210 (21001, 21002, 21003, …, 21022), 211 (21101, 21102, 21103, …, 21122), 212 (21201, 21202, 21203, …, 21222), 213 (21301, 21302, 21303, …, 21322), 214 (21401, 21402, 21403, …, 21422), 215 (21501, 21502, 21503, …, 21522), 216 (21601, 21602, 21603, …, 21622), 217 (21701, 21702, 21703, …, 21722), 218 (21801, 21802, 21803, …, 21822), 219 (21901, 21902, 21903, …, 21922), 220 (22001, 22002, 22003, …, 22022), 221 (22101, 22102, 22103, …, 22122), 222 (22201, 22202, 22203, …, 22222), 223 (22301, 22302, 22303, …, 22322), 224 (22401, 22402, 22403, …, 22422), 225 (22501, 22502, 22503, …, 22522), 226 (22601, 22602, 22603, …, 22622), 227 (22701, 22702, 22703, …, 22722), 228 (22801, 22802, 22803, …, 22822), 229 (22901, 22902, 22903, …, 22922), 230 (23001, 23002, 23003, …, 23022), 231 (23101, 23102, 23103, …, 23122), 232 (23201, 23202, 23203, …, 23222), 233 (23301, 23302, 23303, …, 23322), 234 (23401, 23402, 23403, …, 23422), 235 (23501, 23502, 23503, …, 23522), 236 (23601, 23602, 23603, …, 23622), 237 (23701, 23702, 23703, …, 23722), 238 (23801, 23802, 23803, …, 23822), 239 (23901, 23902, 23903, …, 23922), 240 (24001, 24002, 24003, …, 24022), 241 (24101, 24102, 24103, …, 24122), 242 (24201, 24202, 24203, …, 24222), 243 (24301, 24302, 24303, …, 24322), 244 (24401, 24402, 24403, …, 24422), 245 (24501, 24502, 24503, …, 24522), 246 (24601, 24602, 24603, …, 24622), 247 (24701, 24702, 24703, …, 24722), 248 (24801, 24802, 24803, …, 24822), 249 (24901, 24902, 24903, …, 24922), 250 (25001, 25002, 25003, …, 25022), 251 (25101, 25102, 25103, …, 25122), 252 (25201, 25202, 25203, …, 25222), 253 (25301, 25302, 25303, …, 25322), 254 (25401, 25402, 25403, …, 25422)

Step	Action	Expected result
		nodes C, D, G_1, G_2: 220 (22001, 22002, 22003, …, 22022), 221 (22101, 22102, 22103, …, 22122), 222 (22201, 22202, 22203, …, 22222), 223 (22301, 22302, 22303, …, 22322), 224 (22401, 22402, 22403, …, 22422), 225 (22501, 22502, 22503, …, 22522), 226 (22601, 22602, 22603, …, 22622), 227 (22701, 22702, 22703, …, 22722), 228 (22801, 22802, 22803, …, 22822), 229 (22901, 22902, 22903, …, 22922), 230 (23001, 23002, 23003, …, 23022), 231 (23101, 23102, 23103, …, 23122), 232 (23201, 23202, 23203, …, 23222), 233 (23301, 23302, 23303, …, 23322), 234 (23401, 23402, 23403, …, 23422)
5	read group 202 from vehicle B	Returns 202 (20201, 20202, 20203, …, 20222), no change within the nodes' group directories
6	read group 202 from vehicle A	returns status > 200 (within UIC header of reply telegram): group does not exist no change within the nodes' group directories
7	delete group 202 in vehicle A	returns status > 200 (within UIC header of reply telegram): group does not exist no change within the nodes' group directories
8	read all groups from vehicle B	returns 201 (20101, 20102, 20103, …, 20122), 202 (20201, 20202, 20203, …, 20222), 203 (20301, 20302, 20303, …, 20322), 202 (20401, 20402, 20403, …, 20422), 205 (20501, 20502, 20503, …, 20522), 206 (20601, 20602, 20603, …, 20622), 207 (20701, 20702, 20703, …, 20722), 208 (20801, 20802, 20803, …, 20822), 209 (20901, 20902, 20903, …, 20922), 210 (21001, 21002, 21003, …, 21022), 211 (21101, 21102, 21103, …, 21122), 212 (21201, 21202, 21203, …, 21222), 213 (21301, 21302, 21303, …, 21322), 214 (21401, 21402, 21403, …, 21422), 215 (21501, 21502, 21503, …, 21522), 216 (21601, 21602, 21603, …, 21622), 217 (21701, 21702, 21703, …, 21722), 218 (21801, 21802, 21803, …, 21822), 219 (21901, 21902, 21903, …, 21922), 220 (22001, 22002, 22003, …, 22022), 221 (22101, 22102, 22103, …, 22122), 222 (22201, 22202, 22203, …, 22222), 223 (22301, 22302, 22303, …, 22322), 224 (22401, 22402, 22403, …, 22422), 225 (22501, 22502, 22503, …, 22522), 226 (22601, 22602, 22603, …, 22622), 227 (22701, 22702, 22703, …, 22722), 228 (22801, 22802, 22803, …, 22822), 229 (22901, 22902, 22903, …, 22922), 230 (23001, 23002, 23003, …, 23022), 231 (23101, 23102, 23103, …, 23122), 232 (23201, 23202, 23203, …, 23222), 233 (23301, 23302, 23303, …, 23322), 234 (23401, 23402, 23403, …, 23422), 235 (23501, 23502, 23503, …, 23522), 236 (23601, 23602, 23603, …, 23622), 237 (23701, 23702, 23703, …, 23722), 238 (23801, 23802, 23803, …, 23822),

Step	Action	Expected result
		239 (23901, 23902, 23903, …, 23922), 240 (24001, 24002, 24003, …, 24022), 241 (24101, 24102, 24103, …, 24122), 242 (24201, 24202, 24203, …, 24222), 243 (24301, 24302, 24303, …, 24322), 244 (24401, 24402, 24403, …, 24422), 245 (24501, 24502, 24503, …, 24522), 246 (24601, 24602, 24603, …, 24622), 247 (24701, 24702, 24703, …, 24722), 248 (24801, 24802, 24803, …, 24822), 249 (24901, 24902, 24903, …, 24922), 250 (25001, 25002, 25003, …, 25022), 251 (25101, 25102, 25103, …, 25122), 252 (25201, 25202, 25203, …, 25222), 253 (25301, 25302, 25303, …, 25322), 254 (25401, 25402, 25403, …, 25422) no change within the nodes' group directories
9	write group 201 (1, 2, 3, …, 22) to vehicle A	returns status > 200 (within UIC header of reply telegram): group already exists no change within the nodes' group directories
10	delete all groups in vehicle A delete all groups in vehicle C delete all groups in vehicle D delete all groups in vehicle G_1 delete all groups in vehicle G_2	**node B:** 201 (20101, 20102, 20103, …, 20122), 202 (20201, 20202, 20203, …, 20222), 203 (20301, 20302, 20303, …, 20322), 202 (20401, 20402, 20403, …, 20422), 205 (20501, 20502, 20503, …, 20522), 206 (20601, 20602, 20603, …, 20622), 207 (20701, 20702, 20703, …, 20722), 208 (20801, 20802, 20803, …, 20822), 209 (20901, 20902, 20903, …, 20922), 210 (21001, 21002, 21003, …, 21022), 211 (21101, 21102, 21103, …, 21122), 212 (21201, 21202, 21203, …, 21222), 213 (21301, 21302, 21303, …, 21322), 214 (21401, 21402, 21403, …, 21422), 215 (21501, 21502, 21503, …, 21522), 216 (21601, 21602, 21603, …, 21622), 217 (21701, 21702, 21703, …, 21722), 218 (21801, 21802, 21803, …, 21822), 219 (21901, 21902, 21903, …, 21922), 220 (22001, 22002, 22003, …, 22022), 221 (22101, 22102, 22103, …, 22122), 222 (22201, 22202, 22203, …, 22222), 223 (22301, 22302, 22303, …, 22322), 224 (22401, 22402, 22403, …, 22422), 225 (22501, 22502, 22503, …, 22522), 226 (22601, 22602, 22603, …, 22622), 227 (22701, 22702, 22703, …, 22722), 228 (22801, 22802, 22803, …, 22822), 229 (22901, 22902, 22903, …, 22922), 230 (23001, 23002, 23003, …, 23022), 231 (23101, 23102, 23103, …, 23122), 232 (23201, 23202, 23203, …, 23222), 233 (23301, 23302, 23303, …, 23322), 234 (23401, 23402, 23403, …, 23422), 235 (23501, 23502, 23503, …, 23522), 236 (23601, 23602, 23603, …, 23622), 237 (23701, 23702, 23703, …, 23722), 238 (23801, 23802, 23803, …, 23822), 239 (23901, 23902, 23903, …, 23922), 240 (24001, 24002, 24003, …, 24022), 241 (24101, 24102, 24103, …, 24122), 242 (24201, 24202, 24203, …, 24222), 243 (24301, 24302, 24303, …, 24322), 244 (24401, 24402, 24403, …, 24422), 245 (24501, 24502, 24503, …, 24522), 246 (24601, 24602, 24603, …, 24622), 247 (24701, 24702, 24703, …, 24722), 248 (24801, 24802, 24803, …, 24822), 249 (24901, 24902, 24903, …, 24922), 250 (25001, 25002, 25003, …, 25022), 251 (25101, 25102, 25103, …, 25122), 252 (25201, 25202, 25203, …, 25222), 253 (25301, 25302, 25303, …, 25322), 254 (25401, 25402, 25403, …, 25422)

Step	Action	Expected result
11	delete group 201 in vehicle B	**node B:** 201 (20101, 20102, 20103, …, 20122), 202 (20201, 20202, 20203, …, 20222), 203 (20301, 20302, 20303, …, 20322), 202 (20401, 20402, 20403, …, 20422), 205 (20501, 20502, 20503, …, 20522), 206 (20601, 20602, 20603, …, 20622), 207 (20701, 20702, 20703, …, 20722), 208 (20801, 20802, 20803, …, 20822), 209 (20901, 20902, 20903, …, 20922), 210 (21001, 21002, 21003, …, 21022), 211 (21101, 21102, 21103, …, 21122), 212 (21201, 21202, 21203, …, 21222), 213 (21301, 21302, 21303, …, 21322), 214 (21401, 21402, 21403, …, 21422), 215 (21501, 21502, 21503, …, 21522), 216 (21601, 21602, 21603, …, 21622), 217 (21701, 21702, 21703, …, 21722), 218 (21801, 21802, 21803, …, 21822), 219 (21901, 21902, 21903, …, 21922), 220 (22001, 22002, 22003, …, 22022), 221 (22101, 22102, 22103, …, 22122), 222 (22201, 22202, 22203, …, 22222), 223 (22301, 22302, 22303, …, 22322), 224 (22401, 22402, 22403, …, 22422), 225 (22501, 22502, 22503, …, 22522), 226 (22601, 22602, 22603, …, 22622), 227 (22701, 22702, 22703, …, 22722), 228 (22801, 22802, 22803, …, 22822), 229 (22901, 22902, 22903, …, 22922), 230 (23001, 23002, 23003, …, 23022), 231 (23101, 23102, 23103, …, 23122), 232 (23201, 23202, 23203, …, 23222), 233 (23301, 23302, 23303, …, 23322), 234 (23401, 23402, 23403, …, 23422), 235 (23501, 23502, 23503, …, 23522), 236 (23601, 23602, 23603, …, 23622), 237 (23701, 23702, 23703, …, 23722), 238 (23801, 23802, 23803, …, 23822), 239 (23901, 23902, 23903, …, 23922), 240 (24001, 24002, 24003, …, 24022), 241 (24101, 24102, 24103, …, 24122), 242 (24201, 24202, 24203, …, 24222), 243 (24301, 24302, 24303, …, 24322), 244 (24401, 24402, 24403, …, 24422), 245 (24501, 24502, 24503, …, 24522), 246 (24601, 24602, 24603, …, 24622), 247 (24701, 24702, 24703, …, 24722), 248 (24801, 24802, 24803, …, 24822), 249 (24901, 24902, 24903, …, 24922), 250 (25001, 25002, 25003, …, 25022), 251 (25101, 25102, 25103, …, 25122), 252 (25201, 25202, 25203, …, 25222), 253 (25301, 25302, 25303, …, 25322), 254 (25401, 25402, 25403, …, 25422)
12	write group list: 201 (20101, 20102, 20103, …, 20122), 202 (1, 2, 3) to vehicle B	returns status > 200 (within UIC header of reply telegram): group already exists **node B:** 201 (20101, 20102, 20103, …, 20122), 202 (20201, 20202, 20203, …, 20222), 203 (20301, 20302, 20303, …, 20322), 202 (20401, 20402, 20403, …, 20422), 205 (20501, 20502, 20503, …, 20522), 206 (20601, 20602, 20603, …, 20622), 207 (20701, 20702, 20703, …, 20722), 208 (20801, 20802, 20803, …, 20822), 209 (20901, 20902, 20903, …, 20922), 210 (21001, 21002, 21003, …, 21022), 211 (21101, 21102, 21103, …, 21122), 212 (21201, 21202, 21203, …, 21222), 213 (21301, 21302, 21303, …, 21322), 214 (21401, 21402, 21403, …, 21422), 215 (21501, 21502, 21503, …, 21522), 216 (21601, 21602, 21603, …, 21622), 217 (21701, 21702, 21703, …, 21722), 218 (21801, 21802, 21803, …, 21822), 219 (21901, 21902, 21903, …, 21922), 220 (22001, 22002, 22003, …, 22022), 221 (22101, 22102, 22103, …, 22122), 222 (22201, 22202, 22203, …, 22222),

Step	Action	Expected result
		223 (22301, 22302, 22303, …, 22322), 224 (22401, 22402, 22403, …, 22422), 225 (22501, 22502, 22503, …, 22522), 226 (22601, 22602, 22603, …, 22622), 227 (22701, 22702, 22703, …, 22722), 228 (22801, 22802, 22803, …, 22822), 229 (22901, 22902, 22903, …, 22922), 230 (23001, 23002, 23003, …, 23022), 231 (23101, 23102, 23103, …, 23122), 232 (23201, 23202, 23203, …, 23222), 233 (23301, 23302, 23303, …, 23322), 234 (23401, 23402, 23403, …, 23422), 235 (23501, 23502, 23503, …, 23522), 236 (23601, 23602, 23603, …, 23622), 237 (23701, 23702, 23703, …, 23722), 238 (23801, 23802, 23803, …, 23822), 239 (23901, 23902, 23903, …, 23922), 240 (24001, 24002, 24003, …, 24022), 241 (24101, 24102, 24103, …, 24122), 242 (24201, 24202, 24203, …, 24222), 243 (24301, 24302, 24303, …, 24322), 244 (24401, 24402, 24403, …, 24422), 245 (24501, 24502, 24503, …, 24522), 246 (24601, 24602, 24603, …, 24622), 247 (24701, 24702, 24703, …, 24722), 248 (24801, 24802, 24803, …, 24822), 249 (24901, 24902, 24903, …, 24922), 250 (25001, 25002, 25003, …, 25022), 251 (25101, 25102, 25103, …, 25122), 252 (25201, 25202, 25203, …, 25222), 253 (25301, 25302, 25303, …, 25322), 254 (25401, 25402, 25403, …, 25422)
13	write group 200 (111) to vehicle A	returns status > 200 (within UIC header of reply telegram): group number is invalid no change within the nodes' group directories
14	write group 255 (111) to vehicle B	returns status > 200 (within UIC header of reply telegram): group number is invalid no change within the nodes' group directories
15	*write group 202 (…)* *with all UIC IDs that exist in the actual current* *train composition to vehicle B as multicast telegram*	**node A:** 202 (…) **node B:** 201 (20101, 20102, 20103, …, 20122), 202 (20201, 20202, 20203, …, 20222), 203 (20301, 20302, 20303, …, 20322), 202 (20401, 20402, 20403, …, 20422), 205 (20501, 20502, 20503, …, 20522), 206 (20601, 20602, 20603, …, 20622), 207 (20701, 20702, 20703, …, 20722), 208 (20801, 20802, 20803, …, 20822), 209 (20901, 20902, 20903, …, 20922), 210 (21001, 21002, 21003, …, 21022), 211 (21101, 21102, 21103, …, 21122), 212 (21201, 21202, 21203, …, 21222), 213 (21301, 21302, 21303, …, 21322), 214 (21401, 21402, 21403, …, 21422), 215 (21501, 21502, 21503, …, 21522), 216 (21601, 21602, 21603, …, 21622), 217 (21701, 21702, 21703, …, 21722), 218 (21801, 21802, 21803, …, 21822), 219 (21901, 21902, 21903, …, 21922), 220 (22001, 22002, 22003, …, 22022), 221 (22101, 22102, 22103, …, 22122), 222 (22201, 22202, 22203, …, 22222), 223 (22301, 22302, 22303, …, 22322), 224 (22401, 22402, 22403, …, 22422), 225 (22501, 22502, 22503, …, 22522), 226 (22601, 22602, 22603, …, 22622), 227 (22701, 22702, 22703, …, 22722), 228 (22801, 22802, 22803, …, 22822), 229 (22901, 22902, 22903, …, 22922), 230 (23001, 23002, 23003, …, 23022), 231 (23101, 23102, 23103, …, 23122), 232 (23201, 23202, 23203, …, 23222), 233 (23301, 23302, 23303, …, 23322), 234 (23401, 23402, 23403, …, 23422),

Step	Action	Expected result
		235 (23501, 23502, 23503, …, 23522), 236 (23601, 23602, 23603, …, 23622), 237 (23701, 23702, 23703, …, 23722), 238 (23801, 23802, 23803, …, 23822), 239 (23901, 23902, 23903, …, 23922), 240 (24001, 24002, 24003, …, 24022), 241 (24101, 24102, 24103, …, 24122), 242 (24201, 24202, 24203, …, 24222), 243 (24301, 24302, 24303, …, 24322), 244 (24401, 24402, 24403, …, 24422), 245 (24501, 24502, 24503, …, 24522), 246 (24601, 24602, 24603, …, 24622), 247 (24701, 24702, 24703, …, 24722), 248 (24801, 24802, 24803, …, 24822), 249 (24901, 24902, 24903, …, 24922), 250 (25001, 25002, 25003, …, 25022), 251 (25101, 25102, 25103, …, 25122), 252 (25201, 25202, 25203, …, 25222), 253 (25301, 25302, 25303, …, 25322), 254 (25401, 25402, 25403, …, 25422) **nodes C, D, G_1, G_2:** 202 (…) Group 202 (except in vehicle B) shall contain valid UIC and TCN addresses according to the actual current train composition
16	delete all groups in all nodes	-

H.6.3.8 - E telegram error handling test suite

Check of the E telegram error handling

Purpose:

Test the handling of erroneous E telegrams that are addressed to the UMS (function 15).

Reference:

Point 5.7.3 - page 23,
Appendix A - page 35.

Composition:

See reference composition.

NB : It shall be ensured that invalid function codes will be rejected with status > 200.

Test sequence:

Step	action	Expected result
	all telegrams shall be sent to function 15 (UIC mapping server)	checks will be done on the UIC header
1	send getNADI reply telegram (1 entry) to vehicle A	Reply telegram: UIC-Code: 0, 1. reserved Byte: 0, source function: 15, status > 200 (no reply telegram expected)
2	send getTopography reply telegram (2 nodes) to vehicle G_1	see above
3	send getStatus reply telegram to vehicle B	see above
4	send UIC header with code 0 to vehicle C	see above
5	send UIC header with code 0x1007 to vehicle D	see above

H.6.3.9 - Sleep mode test suite

Purpose:

Test the correct handling of a sleep mode request on a node and its withdrawal.

Reference:

Appendix A - page 35

Composition:

See reference composition.

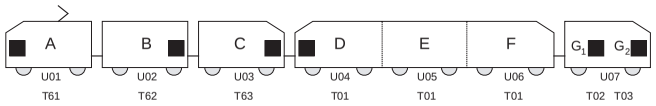
NB : Following basic cases shall be covered:

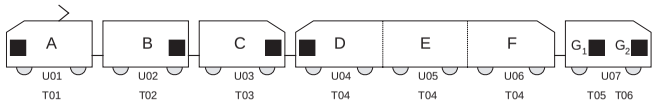
	Test cases	*References to test steps*
1.	intermediate node is last active node before sleeping mode	4-6
2.	end node is last active node before sleeping mode	1-3
3.	turn off the master when not all other nodes have set sleep requests	7-12
4.	all nodes except one intermediate non-master node request sleeping mode and that intermediate node fails	23-26
4.	coupling and uncoupling of trains in sleeping mode	21-22
6.	resetting of sleep requests	13-20

Test sequence:

Step	action	Expected result	Check	Expected configuration
			No vehicle is leading, the configuration is unconfirmed. Checks will be done on the NADI. The sleep bit shall be checked within the mapping server status reply telegrams of all nodes. The reply telegram source function shall always be 15, the reply telegram status shall always be 0, and the reply telegram code shall be checked (0A01 for getNADI, 0A22 for getStatus, 0A03 for setOpMode, FA04 for setSleepRequest /cancelSleepRequest). Empty entries mean: no check necessary.	TCN addresses are specified as far as predictable.
1	turn all nodes on	TCN & UIC inauguration with all nodes	NADI contains 8 entries no sleep bit set	Standard addresses
2	send sleep requests to $ABCDG_1$ [a]	no inauguration	NADI contains 8 entries sleep bit set in $ABCDG_1$	Standard addresses
3	send sleep request to G_2	train in sleep modus	All nodes are in sleep mode	-
4	turn A off and then turn A on	TCN & UIC inauguration with all nodes	NADI contains 8 entries no sleep bit set	Standard addresses
5	send sleep requests to $ABCG_1\ G_2$	no inauguration	NADI contains 8 entries sleep bit set in ABC $G_1\ G_2$	Standard addresses
6	send sleep request to D	train in sleep modus	All nodes are in sleep mode	-

a. The order of sleep request E telegrams to each node should be meaningless (as long as one remaining node does not request sleeping mode), e.g. DCG_1BA should have the same effect

Step	Action	Expected result	Check	Expected configuration
7	turn D off and then turn D on	TCN & UIC inauguration with all nodes	NADI contains 8 entries no sleep bit set	Standard addresses
8	set D strong and then set D weak	[TCN &] 2 UIC inaugurations with all nodes	NADI contains 8 entries no sleep bit set	
9	send sleep requests to B	no inauguration	NADI contains 8 entries sleep bit set in B	see above
10	uncouple between B and C		left train segment: no sleep bit set right train segment: not checked	
11	send sleep request to A		left train segment: sleep bit set in A right train segment: not checked	
12	couple between B and C	TCN & UIC inauguration with all nodes	NADI contains 8 entries no sleep bit set	Standard addresses
13	send sleep request to $ABCDG_1$	no inauguration	NADI contains 8 entries sleep bit set in : $ABCDG_1$	Standard addresses
14	reset sleep request on G_1	no inauguration	NADI contains 8 entries sleep bit set in ABCD	Standard addresses
15	send sleep request to G_2	no inauguration	NADI contains 8 entries sleep bit set in : $ABCDG_2$	Standard addresses
16	reset sleep request on D	no inauguration	NADI contains 8 entries sleep bit set in : $ABCG_2$	Standard addresses
17	send sleep request to G_1	no inauguration	NADI contains 8 entries sleep bit set in: $ABCG_1G_2$	Standard addresses
18	reset sleep request on A	no inauguration	NADI contains 8 entries sleep bit set in : BCG_1G_2	Standard addresses
19	send sleep request to D	no inauguration	NADI contains 8 entries sleep bit set in : $BCDG_1G_2$	Standard addresses

Step	Action	Expected result	check	Expected configuration
20	send sleep request to A	train in sleeping mode	All nodes are in sleep mode	-
21	uncouple between B and C	train in sleeping mode	All nodes are in sleep mode	-
22	couple whole train	train in sleeping mode	All nodes are in sleep mode	-
23	turn G_2 off and then turn G_2 on	TCN & UIC inauguration with all nodes	NADI contains 8 entries no sleep bit set	Standard addresses
24	set A strong and then set A weak	[TCN &] 2 UIC inaugurations with all nodes	NADI contains 8 entries no sleep bit set	
25	send sleep request to $ACDG_1G_2$	no inauguration	NADI contains 8 entries sleep bit set in : $ACDG_1G_2$	see above
26	turn B off	no inauguration	NADI contains 8 entries sleep bit set in : $ACDG_1G_2$	see above

Record of updates

Version	Date	Change	Reason for the change
001.01	01.11.2004	Change of the layout according to M1	New layout based on the UIC guide M1
		Adoption of a history of updates	Increasing the usability
		Adoption of a revision number	Enhancement and redesign of the versioning
001.02	01.08.2005	Change of the layout	UIC requirement
		Modification of the numbering of the Appendices	

Appendix I - Type approved train bus nodes in vehicles that run in international services (Version 001.03, valid from 01.03.2009)

The list of type approved train bus nodes in vehicles that run in international services is to be found on the UIC Website (http://www.uic.org) Technology and Research/Products/UIC leaflets/Appendices/556.

Appendix J - Modification list (Version 002.03, valid from 01.03.2009)

The modification list is to be found on the UIC Website (http://www.uic.org) Technology and Research/Products/UIC leaflets/Appendices/556.

Appendix K - (Reserved)

Appendix L - Safe Data Transmission (Version 001.00, valid from 01.03.2009)

L.1 - Foreword

Most of the R-Data transmitted over the UIC Train Bus are related to operational and comfort functions, but some of these R-data may be used by functions which are safety related. Hence UIC Train Bus operation has to support the transmission of safety critical R-data if defined so by the vehicle property "Vehicle supports Safe Data Transmission" (see Appendix E - page 99). The transmission of safety related R-data however requires additional measures for the detection of invalid R-Data.

Besides the responsibility to transmit R-Data safely, there exists also the requirement to determine the train composition and the train directions, namely the orientation of vehicles in a train with respect to the train reference direction as defined in this leaflet, in a safe manner.

Because all this information is stored in the NADI (Node Attribute and Address Directory) as defined in Appendix C - page 37, the safety requirement can be interpreted as defining a safe way for generating, storing and distributing the NADI. The safe generation and distribution of the NADI includes the safe transmission of the inauguration data telegram as defined in point C.3 - page 84, and the safe transmission of related E-telegrams as defined in Appendix A - page 35 of the leaflet.

This appendix shall present the concepts behind a safe R-data transmission over the UIC Train Bus, shall specify the protocol extensions, namely the vital UIC R-Telegram, needed to support the safe R-Data transmission and shall provide rules for a safe generation, storage and distribution of the NADI.

L.2 - Fulfilment of Safety Requirements

Requirements for a safe data communication are published in many documents, as for instance in the European Standard *EN 50159-1* (see Bibliography - page 236). Those requirements are however mostly related to "static" communication network topologies. For the special case of the UIC Train Bus, which bases on a dynamic train bus topology, additional requirements need to be defined which take the train bus dynamics into account. The complete list of top level requirements is given in Table 13.

Table 13 : UIC Trainbus Safety Requirements

SREQ1 Detect an unacceptable delay or the deletion of R-Telegrams
SREQ2 Detect the re-sequencing of R-Telegrams
SREQ3 Detect the repetition of R-Telegrams
SREQ4 Detect the modification of R-Telegrams
SREQ5 Detect the modification of R-Telegrams source identification
SREQ6 Detect the insertion of R-Telegrams
SREQ7 Detect an invalid NADI
SREQ8 Detect the sending of R-Telegrams produced on the base of an outdated NADI

R-Telegrams are transmitted cyclically, meaning that a loss or an unacceptable delay (SREQ1) of R-Telegrams can be detected by timeout supervision at the data sink. The evaluation of the time stamp in the R-telegrams shall be used to fulfil the requirements SREQ2 and SREQ3. Modified R-Telegrams are detectable (SREQ4) by the Safety Code which is a Cyclic Redundancy Code (CRC32) sequence appended to each R-telegram. The source identification of an R-Telegram consisting of the source vehicle information is used in the calculation of the Safety Code and an alteration is therefore detectable (SREQ5). The R-Telegram itself does not contain source vehicle information. Inserted R-Telegrams are detectable by checking the source identification and checking the time stamp (SREQ6). The assumption is that insertion only occurs resulting from a software or hardware fault and not because of a malevolent intrusion. A special requirement for the UIC Train Bus operation is SREQ7 and SREQ8.

The result of the train inauguration is stored in the NADI (see Appendix C - page 37), and the content of the NADI is used for the production of side-selective and vehicle-selective R-Data. Such it is decisive that this production of R-Data is based on the actually valid and safely determined version of the NADI. By using global NADI version information for the calculation of the Safety Code can the receivers verify whether the R-Data were produced on the base of the actually valid NADI version.

L.3 - Concept of Safe UIC Train Bus Communication

The principle of safe communication is to provide a communication channel end-to-end between a vital source function and a vital destination function. All data exchanged are encapsulated by the source function in a "vital" UIC R-Telegram, which provides means for protection and error detection, and sent over the communication channel to the vital destination function. Here is the integrity of the received R-Telegram checked, and if ok, the data are unpacked and processed.

The "vital" functions are abstract functions provided by the vehicles, and they are thought to be located on a "virtual" end device (VED). It depends on the vehicle design how this virtual end device is physically implemented. So it may be a separate device or it may be the UIC Gateway itself which provides this function. It is also not defined how vital functions shall be implemented, meaning, how they collect and distribute R-Data inside the vehicle. It is only defined that the vital source function (producer of R-Data) shall assemble the vital R-Telegram and that the vital destination functions (consumer of R-Data) shall check and disassemble the received vital R-Telegram. It is quite obvious that these vital functions are safety related and that the safety integrity level of these functions defines the maximal safety integrity level of the transmitted R-Data.

Another important aspect is that vital R-Data are transmitted on the base of the actual train topology. So it shall not happen that R-Data, which were produced on the base of old information about the train topology, are transmitted and processed. Imagine a case where the orientation of a vehicle has changed and the R-Data intended to release the left doors is misinterpreted and releases the right doors instead. It is also in the responsibility of the vital functions to ensure that the vital R-Data, which they collect or distribute, are produced and consumed on the base of the actual train topology.

The actual train topology is determined by the WTB Gateway during the train inauguration and the results are stored in the Node Address and Attribute Directory (NADI), which is a database located in the WTB Gateway. Computation of the NADI is safety related as well, and the contents of the NADI shall be protected against undesired modifications. Also access to the NADI for retrieving some information out of it shall be safety protected.

Whether a vehicle supports safe data transmission or not can defined with a static vehicle property as listed in Appendix E - page 99.

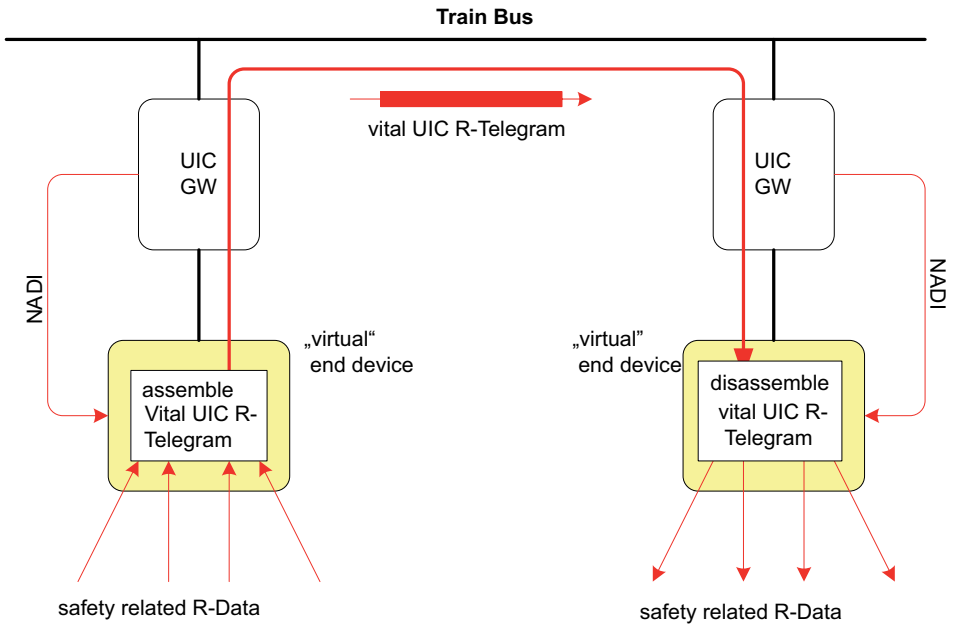


Fig. 22 - Vital UIC R-Telegram Transmission

L.4 - Vital UIC R-Telegram

The R-Telegrams as defined in Appendix B - page 36 are already prepared for the transmission of safety related R-Data. All R-Telegrams contain a safety code, which is implemented as a Cyclic Redundancy Code (CRC) and which checksums the complete R-Telegram (see Fig. 23 - page 227) plus an "implicit" field. This safety code may be set to 0 in applications which require backward compatibility or in application requiring no vital R-Telegrams (e.g. application with an independent communication path for vital data). In those cases the safety code is not used to determine the validity of the R-Telegram. The "implicit" field contains a Safety Identity (SID) which authenticates the "source" of the vital R-Telegram. As this information must be known by the receiver of the vital R-Telegram, is the Safety Identity not transmitted, instead only included in the CRC calculation. Finally the time stamp is used to determine sequence errors.

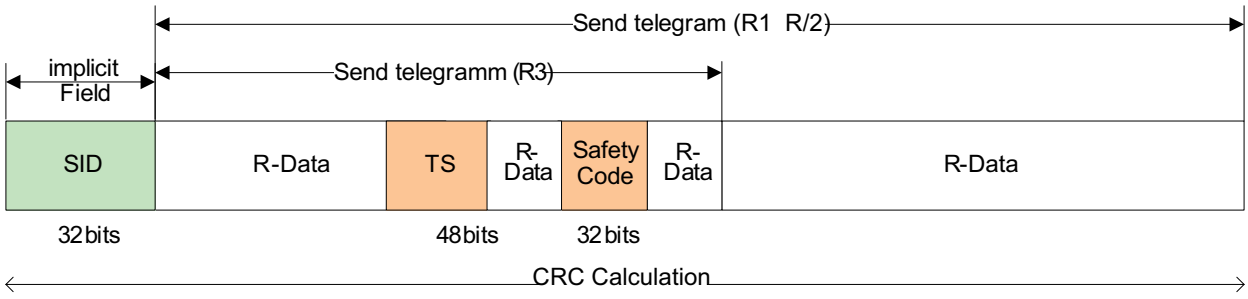


Fig. 23 - Vital UIC R-Telegram

Table 14 : Vital UIC R-Telegram

Entry	Description	Telegram Offset	Data type	Value
SID	The Safety Identity is used to verify that the information comes from the correct producer. Both Producer and consumer know the Safety Identity, why the Safety Identity is not transmitted over the bus, but only appended for calculating the Safety Code.	n/a	see below	see below
R-Data	safety related data	different	any	any
TS	Time stamp	13-18	TIMEDATE48	computed The producer of the vital UIC R-Telegram has to ensure that the time stamp is strictly monotonically increasing (see note).
Safety Code	CRC32 as defined in point L.9 - page 231. Start value = 0xFFFFFFFF Fill value = NADI check sum as defined in point L.7 - page 230.	35-38	UINT32	computed and inserted.

NB : 1. Data are represented in big-endian format.

2. If there is a change of the time reference leading to a "backward jump" in the local clock, a synchronization mechanism shall be used for slowly synchronizing the local clock to the new time reference so that the criteria for the time stamp of "strictly monotonically increasing" will not be violated. Otherwise sequencing errors will be detected which can force the system to enter the fail safe state.

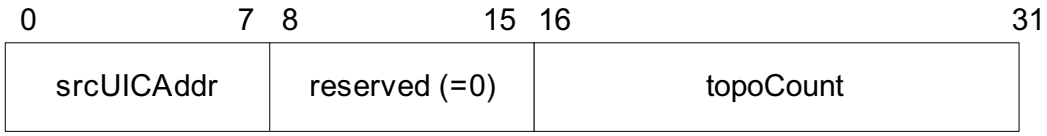


Fig. 24 - Vital UIC R-Telegram - Implicit Field (Safety Identity)

Table 15 : Vital UIC R-Telegram - Implicit Field

Entry	Description	Telegram Offset	Data type	Value
srcUIC Addr	UIC Address of the vehicle with the active (sending) UIC Gateway. For WTB: the receiving function can derive this information from the WTB address of the sending WTB node by looking-up the NADI.	0	UINT8	1..22
reserved	reserved field	8	UINT8	0
topo Count	Actual global version of the NADI which has been used for R-Data production. The 12-bit master_topo value sent by the WTB master in the topography request telegram (IEC61375-1 clause 4.5.3.12.2).	16	UINT16	0..4095

L.5 - Vital UIC R-Telegram Validation

Received R-Telegrams shall be validated before they are used for processing. The validation process shall be performed in two steps:

First is the received R-Telegram's integrity checked and incorrect UIC R-Telegrams are discarded ("filtered"). Filtering shall be done if

- calculated and received Safety Code mismatch,
- the R-Telegram is not a real successor to the previous received R-Telegram, meaning that the time stamp of currently received R-Telegram is not greater than time stamp of previously received R-Telegram,
- UIC Application Type and Telegram Type (Octet 1 in vital UIC R-Telegram) are invalid.

After filtering is the reception of correct R-Telegrams temporally supervised. If there are no correct R-Telegrams received for a time expiring T_{RXSafe}, the receiving application shall assume a disconnection to the related Consist and shall react in accordance to the train safety concept.

It is recommended to set T_{RXSafe} to 2,5 ± 0,5 s [1].

1. This recommendation takes into account that train inaugurations may occur during data transmission which interrupt the data communication over the train bus.

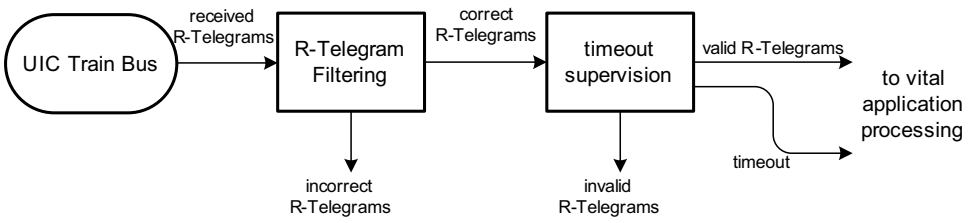


Fig. 25 - Vital UIC R-Telegram Validation

All filtering events should be captured by statistical counters:

- total number of received R-Telegrams,
- number of Safety Code failures,
- number of sequencing failures (wrong sequence),
- number of R-Telegrams with type error.

The statistical counters may be used by the local VED to supervise the quality of the safe data transmission.

L.6 - Safe UIC Inauguration

The UIC Inauguration as defined in Appendix C - page 37 uses the information distributed with the inauguration frames (point C.3 - page 84) to compute the NADI, which contains safety critical information like the train vehicle sequence and the train vehicle orientation. The inauguration frames are (pre-)computed and locally stored by each UIC gateway, and are sent in the payload of WTB topology response frames to all other UIC gateways during the UIC inauguration. In order to protect the inauguration frames after they are computed and stored, a safety code shall be added to the inauguration frame as it is listed in Table 16.

Table 16 : Safe inauguration frame - Safety Code

Octet	Bit	Data type / Scope of values	Code	Meaning	Source of gateway (= trainset)	Source vehicle	Serial No. from Appendix E
35 to 37		BITSET8		Reserve for gateway specific properties	x		
38 to 41		UNSIGNED32		CRC32 as defined in point L.9 - page 231 Start value = 0xFFFFFFFF Fill value = 0xFFFFFFFF	x		
42 to 43		BITSET8		Reserve for national gateway specific properties	x		
44				General reserve	x		

Each UIC gateway shall check the safety code of the received inauguration frame before it computes the NADI. If it receives an invalid inauguration frame[1], it shall:

- invalidate its NADI
- report an inauguration error to the local VED

The local VED decides on the base of the safety concept for that vehicle which action to take, for example:

- forcing a repetition of the UIC inauguration,
- entering a fail safe state

L.7 - NADI Protection

The NADI contains safety sensitive information like vehicle sequence number and vehicle orientation and shall therefore be protected against undetected modifications after it has been computed. The safety integrity level of the NADI computation procedure, the safety integrity level of the NADI protection algorithm and the safety integrity level of the protocols for retrieving information from the NADI determine the maximal possible safety integrity level for vital R-Data which rely on information from the NADI.

UIC Gateway manufacturers shall state for their UIC Gateway which safety integrity level for the NADI is supported.

The NADI is computed de-centrally by each UIC Gateway during UIC inauguration based on the information exchanged by the inauguration frame (see point C.3 - page 84). In order to check the equality of the computed NADIs in the different UIC Gateways, each UIC Gateway shall compute a CRC32 of its computed NADI and insert this checksum in the R-data telegram (see point L.3 - page 225) before calculating the checksum of the R-data telegram according to the following rules:

- the CRC32 is computed over the data range as defined in E-telegram 0.01A (see Appendix A - page 35), starting with octet 11 and ending with octet (26 +n*46 -1) with n being the number of vehicles, handled as byte stream in big endian format. The value of "topo_count" (octet 14) shall be set to 0 for the computation (and restored afterwards).
- Start Value of CRC32 = 0xFFFFFFFF.
- the obtained CRC32 value is inserted in the R-data telegram (octets 35-38) before the computation of the CRC32 of the R-telegram (which then overwrites octets 35-38).

1. This does not mean that other UIC gateways receive also an invalid inauguration telegram. So other UIC gateways may compute a valid NADI !

L.8 - UIC E-Telegram Protection

Although E-Telegrams shall not be used for the transmission of vital R-Data[1], can it well be that the content of some E-Telegrams is used during the production or consumption of safety critical R-Data. Examples are the E-Telegram 0.01A (NADI), which contains information about the sequence and orientation of vehicles used for the addressing and side selection, or the E-Telegram group 15 which deals with inauguration correction and confirmation. This might also be the case for other E-Telegrams. In order to protect those E-Telegrams against undetected modifications, a CRC32 shall be appended at the sending function and checked by the receiving function.

Table 17 : Protected E-Telegram

1..10	E-Telegram header as defined in point 5.7 - page 20.
11..k	Variable E-Telegram content as defined in Appendix A.
k+1 .. k+4	CRC32 as defined in point L.9. Start value: 0xFFFFFFFF

L.9 - CRC32 Safety Code

The CRC32 safety code is defined as the 32-bit CRC polynomial:

$G_{32}(x) = x^{32}+x^{31}+x^{30}+x^{29}+x^{28}+x^{26}+x^{23}+x^{21}+x^{19}+x^{18}+x^{15}+x^{14}+x^{13}+x^{12}+x^{11}+x^{9}+x^{8}+x^{4}+x^{1}+1$

$= 1F4ACFB13$

Details about this polynomial and hints about the implementation of the code generator/checker can be found in *IEC 61784-3-3* (see Bibliography - page 236).

Change history

Version	Date	Change	Reason for the change
001.00	01.03.2009	New Appendix	

1. E-telegrams cannot be used for vital data transmission because they don't fulfil SREQ1..3 and SREQ6 without changes in the E-telegram definition.

List of abbreviations

BMI	Bus Management Interface
E-Data	Event-oriented data
FTF	Frame Type Field
GW	Gateway-train bus node
IDI	Inauguration Data Interface
IEC	International Electrotechnical Commission
IGZ	Train bus industry group (Industriegruppe Zugbus)
ISBN	International Standard Book Number
ISO	International Standardization Organization
JDP	Joint Development Project TCN
LL	Link Layer
LME	Layer Management Entity
LMI	Link Layer to Messenger Interface
LPI	Link Layer Process Data Interface
LSO	Least Significant Octet
MC	Multicast (Sending of messages to several train bus nodes)
MD	Message Data
MDB	Marshalling Data Base
MSO	Most Significant Octet
MVB	Multifunction Vehicle Bus
NADI	Node Address & Attribute Directory
NMA	Network Management Agent
NS	Node Supervisor
NSDB	Node Supervisor Database (Database of the GW - control module)

PD	Process Data
PDM	Process Data Marshalling (bus spreading transfer (shunting) of process variables)
PIL	Processor Interface Library
PV	Process Variable
R-Daten	Regular data
RIC	(Regolamento Internazionale Carrozze) "Agreement on the reciprocal use of passenger and baggage vehicles in international traffic"
RTP	Real Time Protocols
SC	Single cast (Sending of messages only to one other train bus node)
Single Node	Single node
SMAP	Station Management Application Process
Strong Master	Strong master
TC9	Technical Committee 9 (of IEC)
TCN	Train Communication Network
TCN Adress	Number between 1 and 63 which is given during the TCN inauguration
UAGT	UIC Agent
UIC	International Union of Railways
UIC Adress	Also called the UIC sequence number, a number which is given on the basis of the UIC address allocation algorithm
UIC ID	Clear identification number for coaches, 11 significant places
UIMCS	UIC Mapping Server Intelligent Multicast Server
UMS	UIC Mapping Server
UMSI	UIC Mapping Server Remote Interface
UNGS	UIC NADI & Group Server
UTBC	UIC Train Bus Configurator
UWTM	UIC WTB Manager
Weak Master	Weak master
WG22	Working Group 22 (of TC9 of IEC)
WTB	Wired Train Bus
WTB-LL	Wired Train Bus Link Layer

Glossary

Bus capability

In order to guarantee the transmission of information within passenger trains over the train bus, the vehicles need to be suitably equipped. Vehicles, which have none of the following features, must not be allowed to run in these trains.
Regarding the capability of the vehicles to participate in the bus traffic, the following features should be differentiated and marked as shown in point:
- fully bus suitable vehicles
- only vehicles capable of train inauguration
- vehicles that are not capable of train inauguration, which, however, are fitted with a cable for data transmission (cable vehicles).

Driving trailer

A vehicle with a driving cab at least at one end of the vehicle from which the traction and braking functions of a train can be controlled.

Leaflet Edition

This is not a version number, but an indication that the leaflet has been modified and reissued.

Locomotive

A traction unit with no passenger or freight capacity. The traction and braking functions of a train can be controlled from the cabs of the locomotives.

Multiple Unit

A number of vehicles put together to form a unit for the conveyance of passengers and/or freight. The driving cabs of the multiple unit can be located in spe-cial power cars (e.g. locomotives with only one cab) or in end vehicles fitted with driving cabs. Groups of vehicles of the multiple unit can be assembled into trainsets.

UIC Inauguration frame number

This corresponds to the edition number of the UIC Leaflet and the overall version number indicated in the Modification List in Appendix J.

UIC R telegram version number

This corresponds to the edition number of the UIC Leaflet and the overall version number indicated in the Modification List in Appendix J.

Passenger coach

A passenger carrying vehicle without its own propulsion system.

Railcar

A traction unit for the carriage of passengers and/or freight. The traction and braking functions of a train can be controlled from the cabs of the railcar.

Traction Unit

Vehicle with its own power equipment.

Trainset

Groups of coaches and/or railcars, which only have one common connection to the train bus.

UIC Leaflet

UIC Leaflets are documents in which the results of work conducted in the UIC study bodies and at UIC Headquarters, in some cases with the support of industry, are officially set out. They are the product of joint work and are approved by the Association's active members in the relevant bodies (see Internal Regulation A17). The aim of UIC Leaflets is to set down guidelines, standards and processes in the form of clear provisions, in order to ensure interoperability and compatibility and to foster technical progress and communication between members of the Association in all areas of railway activity.

Bibliography

1. UIC leaflets

International union of railways
UIC Leaflet 176: Specifications for passenger information displayed electronically in trains, 1st edition, July 2001

UIC Leaflet 541-5: Brakes - Electropneumatic brake (ep brake) - Electropneumatic emergency brake override (EBO), 4th edition, May 2006

UIC Leaflet 550: Power supply installations for passenger stock, 11th edition, April 2005

UIC Leaflet 550-1: Electrical switch cabinets on passenger stock, 1st edition of 1.1.90

UIC Leaflet 553: Heating, ventilation and air-conditioning in coaches*,* 6th edition, February 2004

UIC Leaflet 555: Electric lighting in passenger rolling stock, 1st edition of 1.1.78 and 7 Amendments

UIC Leaflet 557: Diagnostics on passenger rolling stock, 2nd edition of 1.1.98

UIC Leaflet 558 : Remote control and data cable - Standard technical features for the equipping of RIC coaches, 1st edition of 1.1.96

UIC Leaflet 560: Doors, footboards, windows, steps, handles and handrails of coaches and luggage vans, 12th edition, March 2002

UIC Leaflet 568: Loudspeaker and telephone systems in RIC coaches - Standard technical characteristics, 3rd edition of 1.1.96

UIC Leaflet 647: Functional model for the remote control of traction units, 1st edition, March 2006

UIC Leaflet 660: Measures to ensure the technical compatibility of high-speed trains, 2nd edition, August 2002

2. European standards

European Committee for normalisation (CEN)
EN 50155:2007: Railway applications - Electronic equipment used on rolling stock, August 2007

EN 50159-1:2001: Railway applications - Communication, signalling and processing systems - Part 1: Safety-related communication in closed transmission systems, May 2001

3. International standards

International Organization for Standardization (ISO) /
International Electrotechnical Commission (IEC)
ISO/IEC 8859-1: Information technology - 8-bit single-byte coded graphic character sets - Part 1: Latin alphabet No. 1, Edition 1-0, April 1998

ISO/IEC 10646: Information technology - Universal multiple-octet coded character set (UCS), December 2003

International Electrotechnical Commission (IEC)
IEC 61375-1: Electric railway equipment - Train communication network - Part 1: Train Communication Network. General architecture, June 2009

IEC 61375-2-1: Electronic railway equipment. Train communication network. Part 2-1. WTB. Wire Train Bus, June 2009

IEC 61375-2-2: Electronic railway equipment. Train communication network. Part 2-2. WTB. Wire Train Bus conformance testing, June 2009

IEC 61784-3-3: Industrial communication networks - Profiles - Part 3-3 : functional safety fieldbuses - Additional specifications for CPF 3, Edition 1.0, December 2007

4. Technical reports

International Union of Railways (UIC)
B 108.3/DT 273: Transmission of information in the train by a central data line - Transmission of information through the UIC 12-core train line: test with a token transmission system (SNCF/Alsthom), August 1992

5. Miscellaneous

IGZ Doc. No. 16
UIC-Mapping-Server Specification V 1.0,

JDP / **Adtranz**
WTB link Layer 1.9 User's Manual,

JDP / **Siemens Transportation Systems**
NS Data Base Handling, Rev. A

PDM User's Manual, Rev. B.01

Warning

Printed by the International Union of Railways (UIC)
16, rue Jean Rey 75015 Paris - France, August 2009
Dépôt Légal August 2009

ISBN 978-2-7461-1713-6 (French version)
ISBN 978-2-7461-1714-3 (German version)
ISBN 978-2-7461-1715-0 (English version)