Text communication protocol for synchronization of databases with scales and balances

ver: 0.0.0.4

2015-09-07

RADWAG BALANCES AND SCALES
RADWAG 26 – 600 Radom, 28 Bracka Street, POLAND
Phone +48 48 384 88 00, fax +48 48 385 00 10
Export department +48 48 366 80 06
www.radwag.com

1. Intended use

The protocol has been designed to support synchronization of databases for RADWAG devices. The protocol unifies as many aspects as possible in a way making databases synchronization uniform and similar for all devices.

The protocol has been optimised. As a result of optimisation, typical operation schemes are available: acquisition of all table records, single record replacement, all table records replacement, incremental acquisition of records (not been acquired yet).

The protocol is clear and readable, it can be tested by means of a particular terminal, using it does not require getting familiar with complex database structure, which can be vary for each device.

2. Protocol specification

2.1 Format: command and response

Command format:
COMMAND<main_parameter=value><optional_parameter=value> CR LF

Responce format:
COMMAND<main_parameter=value><optional_parameter=value><STS=status> CR LF

Command content cannot comprise either control characters (0x00 – 0x1F) or ASCII characters: ‘<’, ‘>’, ‘#’ (0x3C, 0x3E and 0x23). Exception to this rule are precisely specified positions. The main parameter, optional parameters and status of command and response are put in angle bracket. Command, parameters and status are adjacent (no character can separate them).

COMMAND is always written with capital letters, it cannot contain space character.

<main parameter> - main parameter indicates within which group optional parameters are comprised. Main parameter is an obligatory component of a request and a response.

<optional parameters> - these are command-related values, they complete main parameter, e.g. for database name being a main parameter, the database records may serve as the optional parameters.

Depending on the needs, **<main parameter>** and **<optional parameters>** may be characterized with parameter-assigned value. For such a case, parameter format is designed as follows:
<NAME=value> - name and value are separated with „=” character, and put in angle bracket. The rule is valid for both: command and response.

<status> - value specifies whether the command has been completed or not, it provides information on reasons for not completing the command.

2.1.1 Protocol field types

Types of fields describing values in communication protocol have been precisely specified. No unspecified field type can be used. List of currently used field types includes: text field, integer field, floating-point number field, date field, enum field, indication field and bit field.

2.1.1.1 Text field

Text field displays all but protocol-reserved characters (control characters and ‘<’ ‘>’ ‘#’).

Two character coding methods are used:
- UTF-8 (basic and recommended),
- Extended ASCII (intended exclusively for devices not operating UNICODE).

Coding method has been clearly specified for a particular device. Given device type MUST apply one of the said methods for all text fields.

To eliminate impermissible characters display within particular field, each text value MUST be subjected to coding operation performed in course of record, and decoding operation performed in course of readout.
Coding method is based on so called ‘byte stuffing’:

Coding operation algorithm:

```
For each X character the following condition shall be met:
If X character is not restricted (control character or ‘<’ ‘>’ ‘#’)
        Use unchanged X character
If X character is restricted
        Use ‘#’ character
        Use = (X ^ 0x40) character
```

Decoding operation algorithm:

```
For each X character the following condition shall be met:
If X != ‘#’
        Use unchanged X character
If X == ‘#’
        Read the next character Y
        Use = (Y ^ 0x40) character
```

An example:
String of characters „**Wanted candidate:\r\nProgrammer C# or Java**” (the string comprises hash character, not sharp) becomes replaced with: „**Wanted candidate:#M#JProgrammer C#c or Java**”. Upon being subjected to decoding the string takes its initial form.

When it comes to UTF-8 coding type, it is theoretically possible to use restricted characters in a 2-byte form (with this the characters are no longer restricted), however UTF-8 standard requires use of the shortest possible form therefore such coding method is NOT RECOMMENDED.

As mentioned before, each text field value MUST be coded first, decoded next. This is necessary even if it is probable that no restricted characters are to appear. With such practice, occurrence of characters disturbing transmission and parsing is prevented.

2.1.1.2 Integer field

Integer field value is a string of digits with ‘-‘ (minus) sign placed in front.

An example:
<COUNT=123>

2.1.1.3 Floating-point number field

Floating-point number field value can take „Fixed-point" form or „Scientific" form. It can be completed with texts indicating states.

- NaN – value is not a number
- + Infinity – plus infinity
- - Infinity – minus infinity

Floating-point number field precision is not strictly specified. It shall be selected in a way proving that text notation does not affect precision loss in relation to binary notation.

Examples:
```
<MIN=123.45>
<FACTOR=1.2345E-5>
<FACTOR=NaN>
```

2.1.1.4 Date field

Date field value takes clearly specified form, the form does not depend on local settings: YYYY-MM-DD HH:MM:SS

An example:
```
<DATE=2014-10-24 17:39:13>
```

2.1.1.5 Enum field

Enum field values are natural numbers. It is possible to place text representation next to number notation. Each enum type is clearly defined in a protocol. All devices MUST feature implemented conversion functions, facilitating conversion of their own enum versions to protocol standardized version. If a particular device receives enum value that does not correspond with the device's enum value than the default enum value is set.

Exemplary enum definition:
```
ESTATUS
{
   NONE  = 0,
   MIN  = 1,
   OK   = 2,
   MAX = 3,
}
```

Field example:
```
<ESTATUS=2>
<ESTATUS=2 OK>
```

2.1.1.6 Indication field

Indication field has been designed to enable transfer of the weighing result (and more). It may contain all information providing result printout that is accordant with Weights and Measures standard. Indication fields value takes the following form:
[numerical value] [unit] [supplementary information], separated with space character.
Numerical value – weighing result, provided with balance readability, it may comprise square brackets determining witch units are verified, which are not.

Unit – unit symbol, clearly specified, all balance units included.
Supplementary information – string of characters providing extra information STRICTLY referred to the result, e.g.:
? – unstable result
! – result determined with air buoyancy taken into account
2 – II weighing range
3 – III weighing range

Caution!
Indication field requires unit symbol display each time.

Examples:
<TARE=23.456 g>
<DENSITY=3.8567 g/cm3>
<MASS= -23.4[56] lb ?!2>

2.1.1.7 Bit field

Bit field value shall be provided in a hexadecimal form in 0xFF format.

An example
< DOSING_MASK=0x0A>

3. Databases support

3.1 General information

3.1.1 Message format

For commands supporting databases **<main parameter>** is table name. Optional parameters are usually table records.

Command format for database support takes the following form:

Command format:
COMMAND<TABLE=TableName><optional parameters=value> CR LF

Response format:
COMMAND<TABLE=TableName><optional parameters=value><STS=status> CR LF

Table names are clearly specified and they are common for all devices. The names are provided in English, they are unique and written in capital letters, they CANNOT contain space character.
Examples:
USERS - users, operators
PRODUCTS - products
CODE_EAN – Ean code

3.1.2 Table records guidelines

Records of each table MUST comprise unique ID field (the primary key). The field size shall be designed in a way preventing overflow. The field shall be no-character field type. At least uint 32 type is recommended.
ID field uniqueness for reports tables (not subjected to modification but to readout and deleting operation only) is provided by the device.
For Data Dictionary Table it is the device and the client that provide ID field uniqueness.

0 value is disabled for the primary key (when enabled for foreign key – it means that there is no relation).

All records consist of set of fields formed in line with the rule: <NAME=value> (field name field and value are separated with „=” character, and put in angle bracket).
An example:
<NAME=onion>

Field name is given in capital letters, it is unique and it **CANNOT** comprise space character. The protocol GUARANTEES that string of characters “<NAME=” (i.e. name adjacent to angle bracket and ‘=’ character) does not occur elsewhere within the whole command (unless intended). It means that fields can be searched using such strings of characters.

An example of a record:
<ID=12345><NAME=Apple><MIN=23.5><MAX=24.5>

The protocol defines set of fields for each table along with fileds’ names and data types that can be contained within a particular field. It is FORBIDDEN to use other than protocol-specified fields.

Record fields can take any order, nevertheless it is recommended to place ID and NAME field, providing that they are present, at the very beginning.
As a response to record readout command: **DBREAD, DBREADID**, the device sends fields, implemented to self-designed table, in a request-defined order. If for the command, field not implemented into device has been defined, then <FieldName=#NOT_EXIST> message is given as a response.
If there are no readout fields defined for a command, then the device sends ALL fields implemented to self-designed table.

The device when acquiring record, reads ALL fields implemented to self-designed table – the remaining fields are ignored. If the sent set does not comprise a specified field, then the device sets the field to a default value. Likewise, if the device receives field, value of which is out of implemented set of values, then the device sets the field to a default value.

3.1.3 Response statutes

Set of response formats:

<OK>	Operation completed successfully
<TAB_NOT_EXIST>	Table does not exist on a particular device
<TAB_FULL>	Table full (new record addition impossible)
<REC_NOT_EXIST>	Record does not exist
<NOT_SUPPORTED>	Operation not possible to be carried out for a particular table, or with given parameters (wrong parameter)
<NO_PERMISSION>	Particular operation performance not allowed (inadequate permissions levels)

3.2 Commands for database operation

3.2.1 DBINFO - Information on table

Format:
DBINFO<TABLE=TableName><PARAM=parameter value>CRLF

Depending on the parameter, the command returns:

- fields available for record of TableName-defined table
- table records quantity

Parameter	Value	Overview
TABLE	PRODUCTS	Products table
	USERS	Operators / users table
	PACKAGES	Packaging table
	CUSTOMERS	Customers table
	WAREHOUSES	Warehouses table
	ADD_VAR	Extra variables table
	UNIV_VAR	Universal variables table
	VEHICLES	Vehicles table
	WEIGHMENTS	Weighings table
	REP_DOSING	Table of reports on dosing
	REP_RECIPES	Table of reports on formulations
	REP_VEH_TRANS	Table of reports on vehicle scale transaction
	REP_DIFF_WEIGHMENTS	Table of reports on differential weighings
	DIFF_WEIGHMENTS	Table of differential weighings
	REP_DENSITY	Table of reports on density
PARAM	COLUMNS	Columns of a particular table
	COUNT	Particular table records quantity

Response options:

DBINFO<TABLE=TableName><COLUMNS=Column1Name Column2Name Column3Name…ColumnNName><STS=OK>CRLF	- (for PARAM=COLUMNS) command understood, names of columns available for a particular table are given as a response
DBINFO<TABLE=TableName><COUNT=RecordsQuantity><STS=OK>CRLF	- (for PARAM=COUNT) command understood, table records quantity given as a response
DBINFO<TABLE=TableName><STS=TAB_NOT_EXIST>CRLF	- table does not exist
DBINFO<TABLE=TableName><STS=NOT_SUPPORTED>CRLF	- operation cannot be carried out for given parameters

Example1:

Command:
DBINFO<TABLE=WEIGHMENTS><PARAM=COUNT>CRLF – inquiry about weighings table records quantity

Response:
DBINFO<TABLE=WEIGHMENTS><COUNT=321><STS=OK>CRLF - Command carried out, weighings table comprises 321 records

Example2:

Command:
DBINFO<TABLE=WEIGHMENTS><PARAM=COLUMNS>CR LF – inquiry about weighings table available columns.

Response:
DBINFO<TABLE=WEIGHMENTS><COLUMNS=ID TIME MASS_CAL MASS_ACT TARE PLATFORM CHECKWEIGHING ID_USER ID_PRODUCT ID_VEHICLE ID_PACKAGE ID_WH_DEST ID_WH_SOURCE ID_CUSTOMER LOT BATCH COUNTER_ST COUNTER_USER REF_MASS UNIT_MASS PRICE VAT DISCOUNT VALUE VAR1 VAR2 VAR3 MIN MAX MIN2 MAX2><STS=OK>CR LF – Command carried out, names of columns available for weighings table are given as a response.

3.2.2 DBREADID – read table record by the primary key ID

Format:
DBREADID<TABLE=TableName><KEY=KEYValue><COLUMNS= Column1Name Column2Name…ColumnNName >CRLF

or

DBREADID<TABLE=TableName><KEY=KEYValue>CRLF

Record with KEY-specified ID, or with ID adjacent to KEY-specified ID (reports tables), but not lower, is given as a response. In case there were no records matching the specified criteria, <REC_NOT_EXIST> message is displayed.

As a response to **DBREADID** command, the device sends fields, implemented to self-designed table, in a request-defined order. **ID field is returned each time, whether defined by the request or not**.
If for the command, field not implemented into device has been defined, then <FieldName=#NOT_EXIST> message is given as a response.
If there are no readout fields defined for a command, then the device sends ALL fields implemented to self-designed table.

Parameter	Value	Overview
TABLE	PRODUCTS	Products table
	USERS	Operators / users table
	PACKAGES	Packaging table
	CUSTOMERS	Customers table
	WAREHOUSES	Warehouses table
	ADD_VAR	Extra variables table
	UNIV_VAR	Universal variables table
	VEHICLES	Vehicles table
	WEIGHMENTS	Weighings table
	REP_DOSING	Table of reports on dosing
	REP_RECIPES	Table of reports on formulations
	REP_VEH_TRANS	Table of reports on vehicle scale transaction
	REP_DIFF_WEIGHMENTS	Table of reports on differential weighings
	DIFF_WEIGHMENTS	Table of differential weighings
	REP_DENSITY	Table of reports on density
KEY	Integer	Primary key value – ID of record about which the device is to be inquired.
COLUMNS	Particular table columns names	Parameter defining, which record columns are to given as a response. Columns names are separated by space character.

Response options

DBREADID<TABLE=TableName><KEY=KEYValue> <ID=IDValue><COLUMN1=Field1Value> <COLUMN2=Field2Value> … <COLUMNn=FieldNValue><STS=OK>CRLF	- Command understood, table record of KEY-specified ID, or with ID adjacent to KEY-specified ID (reports tables), but not lower, is given as a response
DBREADID<TABLE=TableName><KEY= KEYValue> <ID=IDValue><COLUMN1=Field1Value> <COLUMN2=#NOT_EXIST> … <COLUMNn=VFieldNValue><STS=OK>CRLF	- Command understood, table record of KEY-specified ID, or with ID adjacent to KEY-specified ID (reports tables), but not lower, is given as a response, wherein the device does not feature field named COLUMN2
DBREADID<TABLE=TableName> <STS=TAB_NOT_EXIST>CRLF	- table does not exist
DBREADID<TABLE=TableName> <STS=REC_NOT_EXIST>CRLF	- record does not exist
DBREADID<TABLE=TableName> <STS=NOT_SUPPORTED>CRLF	- operation cannot be carried out for given parameters

Example1:

Command:
DBREADID<TABLE=WEIGHMENTS><KEY=1100>CRLF – read weighings table record of ID 1100, return all record fields

Response:
DBREADID<TABLE=WEIGHMENTS><KEY=1100><ID=1129><TIME=2015-08-27 12:14:07><MASS_CAL=0.142 kg><MASS_ACT=0.142 kg><TARE=0.261 kg><PLATFORM=1><CHECKWEIGHING=2><ID_USER=1><ID_PRODUCT=1><ID_CUSTOMER=1073741825><ID_VEHICLE=0><ID_PACKAGE=1073741826><ID_WH_DEST=0><ID_WH_SOURCE=0><LOT=123abc><BATCH=def345><COUNTER_ST=13><COUNTER_USER=206><REF_MASS=0 kg><UNIT_MASS=0.14 kg><PRICE=100 €><VAT=0><DISCOUNT=0><VALUE=101.43€><VAR1=><VAR2=><VAR3=><MIN=0.14><MAX=0.144><MIN2=0.105><MAX2=0.125><STS=OK>CR LF – Command completed successfully, no record of ID 1100 found, record of adjacent but not lower ID, ID=1129, is given as a response.

Example2:

Command:
DBREADID<TABLE=WEIGHMENTS><KEY=1129><COLUMNS=MASS_ACT TIME>CRLF
– read weighings table record of ID 1129, return values of current weight and time fields

Response:
DBREADID<TABLE=WEIGHMENTS><KEY=1129><ID=1129><MASS_ACT=0.142 kg> <TIME=2015-08-27 12:14:07><STS=OK> CR LF – Command completed successfully, values of current weight and time fields of ID=1129 record, given as a response.

3.2.3 DBREADN – read table record by N index

Format:
DBREADN<TABLE=TableName><KEY=KEYValue><COLUMNS=Column1Name Column2Name …ColumnNName>CRLF

or

DBREADN<TABLE=TableName><KEY=KEYValue>CRLF

Record with KEY-specified N index, is given as a response. In case there was no record matching the specified criteria, <REC_NOT_EXIST> message is displayed.

As a response to **DBREADID** command, the device sends fields, implemented to self-designed table, in a request-defined order. **ID field is returned each time, whether defined by the request or not**.
If for the command, field not implemented into device has been defined, then <FieldName=#NOT_EXIST> message is given as a response.

If there are no readout fields defined for a command, then the device sends ALL fields implemented to self-designed table.

Parameter	Value	Overview
TABLE	PRODUCTS	Products table
	USERS	Operators / users table
	PACKAGES	Packaging table
	CUSTOMERS	Customers table
	WAREHOUSES	Warehouses table
	ADD_VAR	Extra variables table
	UNIV_VAR	Universal variables table
	VEHICLES	Vehicles table
	WEIGHMENTS	Weighings table
	REP_DOSING	Table of reports on dosing
	REP_RECIPES	Table of reports on formulations
	REP_VEH_TRANS	Table of reports on vehicle scale transaction
	REP_DIFF_WEIGHMENTS	Table of reports on differential weighings
	DIFF_WEIGHMENTS	Table of differential weighings
	REP_DENSITY	Table of reports on density
KEY	Integer	Index value – N of record about which the device is to be inquired.
COLUMNS	Particular table columns names	Parameter defining, which record columns are to given as a response. Columns names are separated by space character.

Response options

DBREADN<TABLE=TableName><KEY=KEYValue> <ID=IDValue><COLUMN1=ValuePola1> <COLUMN2=ValuePola2> … <COLUMNn=ValuePolaN><STS=OK>CRLF	- Command understood, record of index N given as a response
DBREADN<TABLE=TableName><KEY=KEYValue> <ID=IDValue><COLUMN1=ValuePola1> <COLUMN2=#NOT_EXIST> … <COLUMNn=ValuePolaN><STS=OK>CRLF	- Command understood, record of index N given as a response, wherein the device does not feature field named COLUMN2
DBREADN<TABLE=TableName> <STS=TAB_NOT_EXIST>CRLF	- table does not exist
DBREADN<TABLE=TableName> <STS=REC_NOT_EXIST>CRLF	- rekord does not exist
DBREADN<TABLE=TableName> <STS=NOT_SUPPORTED>CRLF	- operation cannot be carried out, incorrect parameters

<u>Example1</u>:

Command:
DBREADN<TABLE=WEIGHMENTS><KEY=104>CRLF – read record of index N=104, return all record fields.

Response:
DBREADN<TABLE=WEIGHMENTS><KEY=104><ID=1129><TIME=2015-08-27 12:14:07>
<MASS_CAL=0.142 kg><MASS_ACT=0.142 kg><TARE=0.261 kg><PLATFORM=1>
<CHECKWEIGHING=2><ID_USER=1><ID_PRODUCT=1><ID_CUSTOMER=1073741825>
<ID_VEHICLE=0><ID_PACKAGE=1073741826><ID_WH_DEST=0><ID_WH_SOURCE=0>
<LOT=123abc><BATCH=def345><COUNTER_ST=13><COUNTER_USER=206>
<REF_MASS=0 kg><UNIT_MASS=0.14 kg><PRICE=100 €><VAT=0><DISCOUNT=0>
<VALUE=101.43 €><VAR1=334><VAR2=123><VAR3=456><MIN=0.14><MAX=0.144>
<MIN2=0.105> <MAX2=0.125><STS=OK>CRLF – Command completed successfully, record of index N=102 given as a response, the record primary key is ID=1129

<u>Example2</u>:

Command:
DBREADN<TABLE=WEIGHMENTS ><KEY=96><COLUMNS=TIME MASS_CAL TARE>CRLF
– read weighings table record of index N=96, return values of the following fields: weight given in calibration unit, time, tare.

Response:
DBREADN<TABLE=WEIGHMENTS><KEY=96><ID=1121><TIME=2015-08-27 11:28:27>
<MASS_CAL=0.142 kg><TARE=0.333 kg><STS=OK>CR LF – Command completed successfully, values of the following fields (of record N=96) given as a response: weight given in calibration unit, time, tare. The response provides ID field regardless of the fact that it has not been defined by the command.

3.2.4 DBADD – Add record

Format:
DBADD<TABLE=TableName><ID=IDValue> <COLUMN1=Field1Value>
<COLUMN2=Field2Value>…<COLUMNn=FieldNValue>CRLF

Command adds record of TableName-defined table. Particular table fields adopt command-supplied values. Command-undefined fields take values default for a given device. If command contains no ID field than it is the device that provides unique ID for the added record.
The device accepts only those fields that are implemented into it, the remaining fields are ignored.
Caution!
Due to optimization aspects the device DOES NOT CHECK whether a record of command-specified ID already exists or not. It is the client who shall be responsible for taking care of providing a unique ID of record added using protocol.

Parameter	Value	Overview
TABLE	PRODUCTS	Products table
	USERS	Operators / users table
	PACKAGES	Packaging table
	CUSTOMERS	Customers table
	WAREHOUSES	Warehouses table
	ADD_VAR	Extra variables table
	UNIV_VAR	Universal variables table
	VEHICLES	Vehicles table
ID	Integer	ID intended for record that is to be added
COLUMN1 COLUMN2 … COLUMN3	Values defined for a particular table column	Table columns that are to take command-defined values, wherein the defining process occurs in course of new record creation.

Response options

DBADD<TABLE=TableName><ID=IDValue> <STS=OK>CRLF	- Command understood, record of specified ID added
DBADD<TABLE=TableName> <STS=TAB_NOT_EXIST> CR LF	- table does not exist
DBADD<TABLE=TableName><STS=TAB_FULL>C RLF	- table full (no more records can be added)
DBADD<TABLE=TableName> <STS=NOT_SUPPORTED>CRLF	- operation cannot be carried out, incorrect parameters
DBADD<TABLE=TableName> <STS=NO_PERMISSION>CRLF	- operation performance not allowed (inadequate permissions levels)

Example:

Command:
DBADD<TABLE=PRODUCTS><ID=854><NAME=apple><CODE=abc12>
<CODE_EAN=1234567890123><MASS=15.36><MIN=15><MAX=15.75>CR LF

- add to products table record of ID 854 providing the following data:
  product name – apple
  product code – abc12
  EAN code – 1234567890123
  weight - 15.36
  MIN threshold – 15
  MAX threshold – 15.75

Response:
DBADD<TABLE=PRODUCTS><ID=854><STS=OK>CR LF - Command completed successfully, record of ID 854 has been added.

3.2.5 DBDELID – Delete record by primary key ID

Format:
DBDELID<TABLE=TableName><KEY=KEYValue>CRLF.

Command enabling to delete record of ID specified by KEY parameter.

Parameter	Value	Overview
TABLE	PRODUCTS	Products table
	USERS	Operators / users table
	PACKAGES	Packaging table
	CUSTOMERS	Customers table
	WAREHOUSES	Warehouses table
	ADD_VAR	Extra variables table
	UNIV_VAR	Universal variables table
	VEHICLES	Vehicles table
KEY	Integer	Primary key value - ID of record that is to be deleted

Response options

DBDELID<TABLE=TableName><KEY=KEYValue><STS=OK>CR LF	- Command understood, table record of ID specified by KEY parameter has been deleted
DBDELID<TABLE=TableName><STS=TAB_NOT_EXIST>CRLF	- table does not exist
DBDELID<TABLE=TableName><STS=REC_NOT_EXIST>CRLF	- rekord does not exist
DBDELID<TABLE=TableName><STS=NOT_SUPPORTED>CRLF	- operation cannot be carried out, incorrect parameters or table
DBDELID<TABLE=TableName><STS=NO_PERMISSION>CR LF	- operation performance not allowed (inadequate permissions levels)

Example:

Command:
DBDELID<TABLE=PRODUCTS><KEY=854> CR LF – delete products table record of ID 854.

Response:

DBDELID<TABLE=PRODUCTS><KEY=854><STS=OK> CR LF - Command completed successfully, products table record of ID 854 has been deleted.

3.2.6 DBDELN – Delete record by index N

Format:
DBDELN<TABLE=TableName><KEY=KEYValue>CRLF

Command enabling to delete record of index N specified by KEY parameter.

Parameter	Value	Overview
TABLE	PRODUCTS	Products table
	USERS	Operators / users table
	PACKAGES	Packaging table
	CUSTOMERS	Customers table
	WAREHOUSES	Warehouses table
	ADD_VAR	Extra variables table
	UNIV_VAR	Universal variables table
	VEHICLES	Vehicles table
KEY	Integer	Index N value of record that is to be deleted

Response options

DBDELN<TABLE=TableName><KEY=KEYValue><STS=OK>CR LF	- Command understood, table record of index N specified by KEY parameter has been deleted
DBDELN<TABLE=TableName><STS=TAB_NOT_EXIST>CRLF	- table does not exist
DBDELN<TABLE=TableName><STS=REC_NOT_EXIST>CRLF	- record does not exist
DBDELN<TABLE=TableName><STS=NOT_SUPPORTED>CRLF	- operation cannot be carried out, incorrect parameters or table
DBDELN<TABLE=TableName><STS=NO_PERMISSION>CR LF	- operation performance not allowed (inadequate permissions levels)

Example:

Command:
DBDELN<TABLE=PRODUCTS><KEY=12>CR LF – delete products table record of index N=12.

Response:
DBDELN<TABLE=PRODUCTS><KEY=12><OK> CR LF - Command completed successfully, products table record of index N=12 has been deleted.

3.2.7 DBCLEAR – Delete all table records

Format:
DBCLEAR<TABLE=TableName>CRLF

Command enabling to delete all records of TableName-defined table.

Parameter	Name	Overview
TABLE	PRODUCTS	Products table
	USERS	Operators / users table
	PACKAGES	Packaging table
	CUSTOMERS	Customers table
	WAREHOUSES	Warehouses table
	ADD_VAR	Extra variables table
	UNIV_VAR	Universal variables table
	VEHICLES	Vehicles table

Response options

DBCLEAR<TABLE=TableName><STS=OK> CRLF	- Command understood, all table records deleted
DBCLEAR<TABLE=TableName><STS=TAB _NOT_EXIST>CRLF	- table does not exist
DBCLEAR<TABLE=TableName><STS=NOT _SUPPORTED>CRLF	- operation cannot be carried out, incorrect parameters or table
DBCLEAR<TABLE=TableName><STS=NO_ PERMISSION>CRLF	- operation performance not allowed (inadequate permissions levels)

<u>Example:</u>

Command:
DBCLEAR<TABLE=PRODUCTS>CRLF – delete all products table records

Response:
DBCLEAR<TABLE=PRODUCTS><STS=OK> CR LF - Command completed successfully, all products table records have been deleted

3.3 Databases tables

The protocol features two table types.

- Tables with records intended for readout, saving and deletion.

TableName	Overview
PRODUCTS	Products table
USERS	Operators / users table
PACKAGES	Packaging table
CUSTOMERS	Customers table
WAREHOUSES	Warehouses table
ADD_VAR	Extra variables table
UNIV_VAR	Universal variables table
VEHICLES	Vehicles table

- Tables with records intended for readout exclusively.

Table Name	Overview
WEIGHMENTS	Weighings table
REP_DOSING	Table of reports on dosing
REP_RECIPES	Table of reports on formulations
REP_VEH_TRANS	Table of reports on vehicle scale transaction
REP_DIFF_WEIGHMENTS	Table of reports on differential weighings
DIFF_WEIGHMENTS	Table of differential weighings
REP_DENSITY	Table of reports on density

3.3.1 Products table

Products database table – columns list.

PRODUCTS – Tabela towarów		
Column name	**Overview**	**Field type**
ID	Record ID	*Integer field*
NAME	Product name	*Text field*
CODE	Product code	*Text field*
CODE_EAN	EAN code for the product	*Integer field*
MASS	Single unit weight in [g] (e.g. weight, single piece weight, reference sample weight – working mode related options)	*Floating-point number field*
MASS_FAST_D	Fast dosing weight value in [g]	*Floating-point number field*
TARE	Product tare value in [g]	*Floating-point number field*
MIN	MIN threshold value in [g]	*Floating-point number field*
MAX	MAX threshold value in [g]	*Floating-point number field*
TOLERANCE	Tolerance expressed in [%]	*Floating-point number field*
MIN2	MIN2 threshold value in [g]	*Floating-point number field*
MAX2	MAX2 threshold value in [g]	*Floating-point number field*
ID_LABEL	ID of label assigned to a product	*Integer field*
ID_LABEL_C	ID of C label assigned to a product	*Integer field*
ID_LABEL_CC	ID of CC label assigned to a product	*Integer field*
MASK_SLOW_D	Mask for outputs for fine dosing	*Bit field* *Least significant bit - output 1*
MASK_FAST_D	Mask for outputs for fast dosing	*Date field*
DATE	Date assigned to product	*Date field*
EXP_DAYS_QNT	Shelf life	*Integer field*
ADD_EXP_DAYS_QNT	Extra shelf life days	*Integer field*
DESCRIPTION	Supplementary product overview	*Text field*
INGREDIENTS	Field for ingredients adding	*Text field*
VAT	Product VAT value, expressed in [%]	*Floating-point number field*
PRICE	Single unit price (expressed in currency provided by CURRENCY field)	*Floating-point number field*
CURRENCY	Currency assigned to product price	*Enum field* 0 – None 1 – Australian Dollar AUD 2 – Bulgarian Dollar BGN 3 – Brazillian Real BRL 4 – Canadian Dollar CAD 5 – Swiss Franc CHF 6 – Chinese Yuan CNY 7 – Czech Koruna CZK 8 – Danish Krone DKK 9 – Euro EUR 10 – Euro € 11 – Pound Sterling GBP 12 – Pound Sterling £

		13 – Hong Kong Dollar HKD 14 – Croatian Kuna HRK 15 – Hungarian Forint HUF 16 – Indonesian Rupiah IDR 17 – Islandic Krone ISK 18 – Japanese Yen JPY 19 – Japanese Yen ¥ 20 – Won (South Korea) KRW 21 – Lithuanian Litas LTL 22 – Latvia Lat LVL 23 – Mexican Peso MXN 24 – Malaysian Ringgit MYR 25 – Norwegian Krone NOK 26 – New Zealand Dollar NZD 27 – Philipinne Peso PHP 28 – Polish Zloty PLN 39 – Romanian Leu RON 30 – Russian Ruble RUB 31 – Swedish Crone SEK 32 – Singapore Dollar SGD 33 – Bat tajlandzki THB 34 – Turkish Lira TRY 35 – Ukrainian Hryvnia UAH 36 – US Dollar USD 37 – US Dollar $ 38 – Rand (South Africa) ZAR
CORRECTION_MAX	Maximum correction in [g]	*Floating-point number field*
DEVIATION_TYPE	Deviation type	*Enum field* 0 – weight 1 – percent
DEVIATION_LOW	Low deviation value expressed in [g] or [%]. DEVIATION_LOW field value is referred to DEVIATION_TYPE field setting.	*Floating-point number field*
DEVIATION_HIGH	High deviation value expressed in [g] or [%]. DEVIATION_LOW field value is referred to DEVIATION_TYPE field setting.	*Floating-point number field*
DENSITY	Density value expressed in [g/cm^3]	*Floating-point number field*
CHARGE	Portion	*Integer field*
PGC_MODE	PGC control mode	*Enum field* 0 – Nondestructive Average Tare 1 – Nondestructive Empty - Full 2 – Destructive Full - Empty 3 – Destructive Empty – Full
PGC_UNIT	Unit for PGC control	*Enum field* 0 – g 1 – ml
BATCH_SIZE	Batch quantity	*Integer field*
T1MIN	Error value [-T] in [g]	*Floating-point number field*
T1MAX	Error value [+T] in [g]	*Floating-point number field*
T2MIN	Error value [-T2] in [g]	*Floating-point number field*
T2MAX	Error value [+T2] in [g]	*Floating-point number field*
DISCQ_2TMIN_QNT	Disqualifying samples quantity [Qn-2T]	*Integer field*
DISCQ_2TMAX_QNT	Disqualifying samples quantity [Qn+2T]	*Integer field*
DISCQ_TMIN_QNT	Disqualifying samples quantity [Qn-T]	*Integer field*
DISCQ_TMAX_QNT	Disqualifying samples quantity [Qn+T]	*Integer field*
AVERAGE_LIMIT_MODE	Average limit value calculation mode	*Enum field* *0 – constant* *1 – automatic*
AVERAGE_MIN	Average limit value [-] in [g]	*Floating-point number field*
AVERAGE_MAX	Average limit value [+] in [g]	*Floating-point number field*
WK_MIN	Coefficient value [-Wk]	*Floating-point number field*
WK_MAX	Coefficient value [+Wk]	*Floating-point number field*
SAMPLE_QNT	Sample quantity	*Integer field*

INTERNAL_CONTROL	Internal control	*Enum field* *0 – disabled* *1 – enabled*
PACKAGE_QNT	Packages quantity	*Integer field*
MEAS_REMINDER	Message reminding about the measurement (refers to PGC mode), expressed in [min]	*Integer field*
CYCLIC_AVERAGE_TARE	Cyclic average tare determination	*Enum field* *0 – function disabled* *1 – function enabled*
CYCLIC_AVERAGE_TARE_INTERVAL	Interval for average tare determination, expressed in [h]	*Integer field*
ID_CATEGORY	ID of category assigned to a product	*Integer field*
ID_CUSTOM_IMG	ID of image assigned to a product	*Integer field*
ID_TRACEABILITY	ID of traceability process assigned to a product	*Integer field*

3.3.2 Users / Operators table

Users/Operators database table – columns list.

USERS – Users/Operators table		
Column name	**Overview**	**Field type**
ID	Record ID	*Integer field*
NAME	Operator name	*Text field*
CODE	Operator code	*Text field*
PSW	Operator password	*Text field*
PERM	Permissions level for the operator	*Enum field* 0 – Brak (gość) 1 – Operator 2 – Operator zaawansowany 3 – Administrator
CARD_NO	Operator card number	*Integer field*
MODE	Working mode assigned to an operator	*Enum field* *0 – None* *1 – Weighing* *2 – Parts counting* *3 – Percent Weighing* *4 – Dosing* *5 – Formulations* *6 – Animal Weighing* *7 – Density* *8 – Solids Density* *9 – Liquids Density* *10 – Peak Hold* *11 – Totalizing* *12 – Checkweighing* *13 – Statistics* *14 – Pipettes calibration* *15 – Differential Weighing* *16 – Statistic Quality Control (SQC)* *17 – Prepackaged Goods Control (PGC)* *18 – Weight Control (Automatic Feeder)* *19 – Drying* *20 – Comparator* *21 – Vehicle Scale*
AUTO_MODE	Auto launch of the most recently operated mode by a particular operator, carried out upon logging operation.	*Enum field* 0 – Function disabled 1 – Function enabled
ID_TRACEABILITY	ID of a traceability process assigned	*Integer field*

	to a particular operator	
ID_PROFILE	ID of a profile assigned to a particular operator	*Integer field*
LANGUAGE	Language assigned to an operator	*Enum field* 0 – Polish 1 – English 2 – German 3 – French 4 – Spanish 5 – Korean 6 – Turkish 7 – Chinese 8 – Italian 9 – Czech 10 – Romanian 11 – Hungarian 12 – Russian

3.3.3 Packages table

Packages database table – columns list.

PACKAGES – Packages table		
Column name	**Overview**	**Field type**
ID	Record ID	*Integer field*
NAME	Package name	*Text field*
CODE	Package code	*Text field*
MASS	Package weight in [g]	*Floating-point number field*

3.3.4 Tabela kontrahentów

Customers database table – columns list.

CUSTOMERS – Customers table		
Column name	**Overview**	**Field type**
ID	Record ID	*Integer field*
NAME	Customer name	*Text field*
CODE	Customer code	*Text field*
TAX_ID	Customer TAX ID	*Text field*
ADDRESS	Customer address	*Text field*
POSTAL_CODE	Custoemr postal code	*Text field*
CITY	Customer city	*Text field*
DISCOUNT	Discount for a customer given in [%]	*Floating-point number field*
ID_LABEL	ID of label assigned to a customer	*Integer field*

3.3.5 Warehouses table

Warehouses database table – columns list.

WAREHOUSES – Warehouses table		
Column name	**Overview**	**Field type**
ID	Record ID	*Integer field*
NAME	Warehouse name	*Text field*
CODE	Warehouse code	*Text field*
DESCRIPTION	Supplementary warehouse overview	*Text field*

3.3.6 Extra variables table

Extra variables table – columns list.

ADD_VAR – Extra variables table		
Column name	**Overview**	**Field type**
ID	Record ID	*Integer field*
CODE	Extra variable code	*Text field*
VALUE	Extra variable value	*Text field*

3.3.7 Universal variable table

Extra variables table – columns list.

UNIV_VAR – Universal variables table		
Column name	**Overview**	**Field type**
ID	Record ID	*Integer field*
NAME	Universal variable name	*Text field*
CODE	Universal variable code	*Integer field*
VALUE	Universal variable value	*Text field*

3.3.8 Vehicles table

Vehicles table – columns list.

VEHICLES – Vehicles table		
Column name	**Overview**	**Field type**
ID	Record ID	*Integer field*
NAME	Vehicle name	*Text field*
CODE	Vehicle code (registration number)	*Text field*
TARE	Vehicle tare value in [g]	*Floating-point number field*
CARD_NO	Transponder card number	*Integer field*
DESCRIPTION	Supplementary Vehicle Overview	*Text field*

3.3.9 Weighings table

Weighings table – columns list.

WEIGHMENTS – Weighings table		
Column name	**Overview**	**Field type**
ID	Record ID	*Integer field*
TIME	Record date and time	*Date field*
MASS_CAL	Calibration unit weight	*Indication field*
MASS_ACT	Current unit weight	*Indication field*
TARE	Tare	*Indication field*
PLATFORM	Weighing platform number	*Integer field*
CHECKWEIGHING	Checkweighing status [min, ok, max]	*Enum field* *0 – None* *1 – MIN* *2 – OK.* *3 – MAX*
ID_USER	Operator ID (User ID)	*Integer field*
ID_PRODUCT	Product ID	*Integer field*
ID_VEHICLE	Vehicle ID	*Integer field*
ID_PACKAGE	Package ID	*Integer field*
ID_WH_DEST	Target warehouse ID	*Integer field*
ID_WH_SOURCE	Source warehouse ID	*Integer field*
ID_CUSTOMER	Customer ID	*Integer field*
MODE	Mode by means of which the weighing has been carried out	*Enum field* *0 – No mode assigned*

		1 – Weighing *2 – Parts Counting* *3 – Percent Weighing* *4 – Dosing* *5 – Formulations* *6 – Animal Weighing* *7 – Density* *8 – Solids Density* *9 – Liquids Density* *10 – Peak Hold* *11 – Totalizing* *12 – Checkweighing* *13 – Statistics* *14 – Pipettes Calibration* *15 – Differential Weighing* *16 – Statistic Quality Control (SQC)* *17 – Prepackaged Goods Control (PGC)* *18 – Weight control (Automatic Feeder)* *19 – Drying* *20 – Comparator* *21 – Vehicle Scale*
LEVELING_STATUS	Levelling Status	*Enum field* *0 – None* *1 – device levelled* *2 – device not levelled*
LOT	Lot number	*Text field*
BATCH	Batch number	*Text field*
COUNTER_ST	Measurements Counter Value (statistics derived counter)	*Integer field*
COUNTER_USER	Measurements Counter Value (master counter)	*Integer field*
REF_MASS	Target weight (predefined value)	*Indication field*
UNIT_MASS	Singele unit weight	*Indication field*
PRICE	Price given in currency assigned to a product.	*Floating-point number field*
VAT	VAT value. Value expressed in [%]	*Floating-point number field*
DISCOUNT	Discount value. Value expressed in [%]	*Floating-point number field*
VALUE	Value (charge), expressed in currency assigned to a product.	*Floating-point number field*
VAR1	Value of universal variable 1	*Text field*
VAR2	Value of universal variable 2	*Text field*
VAR3	Value of universal variable 3	*Text field*
VAR4	Value of universal variable 4	*Text field*
VAR5	Value of universal variable 5	*Text field*
MIN	MIN threshold value	*Indication field*
MAX	MAX threshold value	*Indication field*
MIN2	MIN2 threshold value	*Indication field*
MAX2	MAX2 threshold value	*Indication field*

3.3.10 Dosing reports table

Dosing reports table – columns list.

REP_DOSING – Tabela raportów dozowań		
Column name	**Overview**	**Field type**
ID	Record ID	*Integer field*
ID_DOSAGE	Dosing process ID	*Integer field*
WEIGHING_QNT	Number of weighings carried out in course of a dosing process	*Integer field*
START_DATE	Start Date	*Date field*
END_DATE	End Date	*Date field*

ID_USER	Operator / user ID	*Integer field*
ID_CUSTOMER	*Customer ID*	*Integer field*
STATUS	Dosing process status	*Enum field* *0 – None* *1 – OK* *2 – Aborted*

3.3.11 Formulation reports table

Formulation reports table – columns list

REP_RECIPES – Formulation reports table		
Column name	**Overview**	**Field type**
ID	Record ID	*Integer field*
ID_RECIPE	Formulations ID	*Integer field*
WEIGHING_QNT	Number of weighings carried out in course of a formulation process	*Integer field*
START_DATE	Start Date	*Date field*
END_DATE	End Date	*Date field*
ID_USER	Operator / User ID	*Integer field*
ID_CUSTOMER	*Customer ID*	*Integer field*
STATUS	Formulation process status	*Enum field* *0 – None* *1 – W trakcie procesu* *2 – OK, wykonana prawidłowo* *3 – Błąd receptury* *4 – Przerwany* *5 – Niezapisana* *6 - Zapisana*
ID_FIRST_WEIGHING	First formulation weighing ID,	*Integer field*
TOTAL_MASS	Totalized weight of formulation	*Indication field*
REF_MASS	Target weight (predefined value)	*Indication field*
ID_WAREHOUSE	Warehouse ID	*Integer field*
INGREDIENT_QNT	Formulation ingredients quantity	*Integer field*

3.3.12 Transaction reports table for vehicle scale

Transaction reports table for vehicle scale – columns list

REP_VEH_TRANS – Transaction reports table for vehicle scale		
Column name	**Overview**	**Field type**
ID	Record ID	*Integer field*
ID_VEHICLE	Vehicle ID	*Integer field*
ID_WEIGHING_ENTRY	ID of weighing carried out on entry	*Integer field*
ID_WEIGHING_EXIT	ID of weighing carried out on exit	*Integer field*
TYPE	Transaction type	*Enum field* *0 – None* 1 – entry 2 – exit 3 – control weighing
LOAD_STATUS	Load status	*Enum field* *0 – None* 1 – loading 2 – unloading 3 – load not changed
MASS	Load weight	*Indication field*
STATUS	Transaction status	*Enum field* *0 – None* 1 – in progress

		2 – completed 3 –aborted
ID_PRODUCT	Product ID	*Integer field*
ID_CUSTOMER	Customer ID	*Integer field*
ID_USER_ENTRY	ID of operator responsible for entry weighing performance	*Integer field*
ID_USER_EXIT	ID of operator responsible for exit weighing performance	*Integer field*
MASS_ENTRY	Entry load weight	*Indication field*
MASS_EXIT	Exit load weight	*Indication field*
START_DATE	Transaction start date	*Date field*
END_DATE	Transaction end date	*Date field*

3.3.13 Differential weighings reports table

Differential weighings reports table – columns list.

REP_DIFF_WEIGHMENTS – Differential weighings reports table		
Column name	**Overview**	**Field type**
ID	ID of record of differential weighings reports table.	*Integer field*
ID_LAST_W	ID of last differential weighings table record, related to this report.	*Integer field*
REC_QNT	Quantity of differential weighings table records, related to this differential weighing report.	*Integer field*

3.3.14 Differential weighings table

Differential weighings table – columns list.

DIFF_WEIGHMENTS – Differential weighings table		
Column name	**Overview**	**Field type**
ID	ID of record of differential weighings table.	*Integer field*
ID_WEIGHING	ID of weighings table record, related to particular differential weighing table record.	*Integer field*
ID_REPORT	ID of record of differential weighings reports table, to which a differential weighing record is related.	*Integer field*

3.3.15 Density reports table

Density reports table – columns list

REP_DENSITY – Density reports table		
Column name	**Overview**	**Field type**
ID	Record ID of density reports table	*Integer field*
START_DATE	Process start date	*Date field*
END_DATE	Process end date	*Date field*
SAMPLE_NO	Sample number	*Text field*
METHOD	Density determination method	*Enum field* 1 – solid body 2 – liquid 3 – air 4 – pycnometer 5 – porous solid body

LIQUID	Model liquid	*Enum field* 1 – water 2 – ethanol 3 – other
ID_PRODUCT	Product ID	*Integer field*
ID_USER	Operator ID	*Integer field*
FLUID_DENSITY	Model liquid density	*Indication field*
TEMPERATURE	Temperature	*Indication field*
ID_WEIGHING1	First density determination weighing ID	*Integer field*
ID_WEIGHING2	Second density determination weighing ID	*Integer field*
ID_WEIGHING3	Third density determination weighing ID	*Integer field*
SINKER_VOL	Sinker density	*Indication field*
DENSITY	Process determined density	*Indication field*
MASS_ST	Steel mass standard weight	*Indication field*
MASS_AL	Aluminium mass standard weight	*Indication field*
DENSITY_ST	Steel mass standard density	*Indication field*
DENSITY_AL	Aluminium mass standard density	*Indication field*
VOLUME	Determined volume value	*Indication field*
PYCNOMETER_MASS	Weight of pycnometer used for density determination	*Indication field*
PYCNOMETER _VOL	Volume of pycnometer used for density determination	*Indication field*

MANUFACTURER

O F   E L E C T R O N I C   W E I G H I N G   I N S T R U M E N T S

RADWAG BALANCES AND SCALES

POLAND, 26 – 600 Radom, 28 Bracka Street

Phone: +48 48 384 88 00, fax: + 48 48 385 00 10

Export department +48 48 366 80 06

export@radwag.com

www.radwag.com

DIN EN ISO 9001:2000

CERTIFICATE NO 71 100 C206