OCTO IP-Server

Control package for RDS encoder systems

User Manual

2011, January

Content

1.Introduction....3

2.Manufacturer....4

3.Installation....5

System Requirements 5

Software Overview....6

4.General operation....7

Principle operation of OCTO IP-Server....7

OCTO-IP I/O Overview....9

OCTO IP-Input Settings....10

OCTO IP-Output Settings....13

Protocol Conversion....16

Address Table....17

PSN-TP Table....18

RDS Data Routing....18

TA/PTY31 GPI-Routing (optional):....20

OCTO IP to your system management server....20

GPI/O (optional)....20

Encoder-Check (optional)....21

Arcos Studio (optional)....22

TA Overview....23

TA over TCP/IP with a XML frame:....23

TA wired at the back of the 2wcom RDS C0x encoder:....23

Failsafe Hardware Set-Up for a TX-Station (optional)....24

Appendix XML Interchange format....25

Appendix EC Exchange Format....36

Appendix Logging....37

Input Log....37

Output Log....38

Appendix OCTO_IP.ini....39

1. Introduction

2wcom OCTO IP is software for MS Windows 2000, XP, Vista, or Windows 2003 Server and operates as a data routing center for controlling a system of RDS encoders.

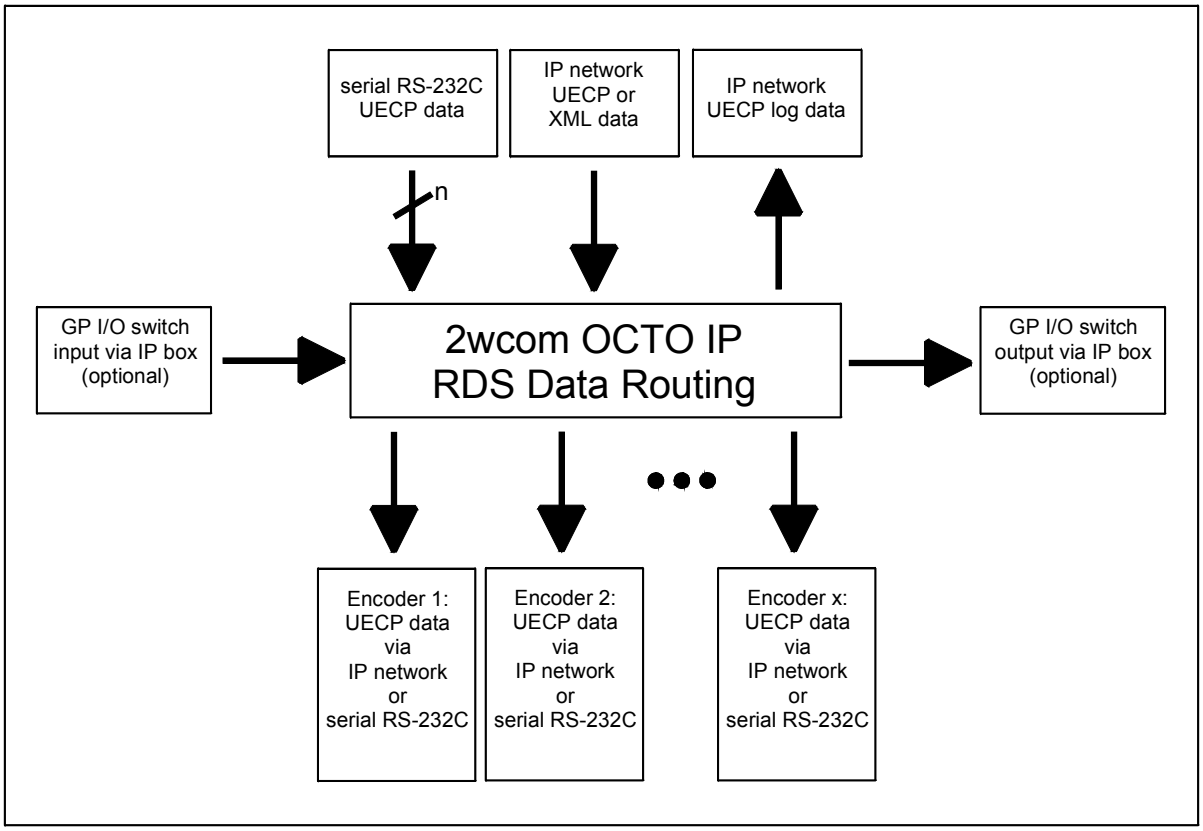
The OCTO IP software has a user interface tool that can be used as a manually operated application. Alternatively it is possible to use the user interface tool to configure and start the OCTO IP service. Once this service is configured, it is started automatically with the windows operating system and works without manual control.

The OCTO IP software supports multiplexed serial RS-232C UECP data inputs (or data from any Windows COM port) and the input of XML data via the IP-interface, as well as serial RS-232C outputs with the routed UECP data for the individual RDS encoders and the output of UECP data streams to the individual RDS encoders via the IP-interface (one stream is only for logging purposes).

Additionally general-purpose switch inputs and general-purpose switch outputs are supported via optional IP-network controlled I/O boxes.

The max. number of handled inputs/outputs is limited by the available serial RS-232C COM ports and the performance of the system.

The incoming data can be routed to the outputs by the PSN number, the defined group name, or the UECP address respectively.



Basic data connection overview

Important

Please read the entire manual before you start to use the software. Keep this manual for future use. It contains important operating instructions. This manual describes the OCTO IP software and parts of the hardware and does not replace the manual of the “2wcom C0x RDS Encoder”. For details of 2wcom C0x functions, please consult the corresponding manual.

Explicit definitions

This software is developed for the use with the MS Windows operating system. Its usage is similar to MS Windows. Please see your MS Windows manual for details of windows operations. For necessary mouse operations please place the mouse pointer on the specified display area and execute one of the following corresponding actions.

Double-click	Quickly push left mouse button twice
Mouse-click	Single push on left mouse button
Right-click	Single push on right mouse button

Additional notes:

The figures in the manual may differ from the actual screen display.

Configurations, functions and specifications may change for further development without notice.

2. Manufacturer

2wcom GmbH • Lise-Meitner-Str. 4 • 24941 Flensburg • Germany
Phone (+49) 461 – 97892-0 • Fax (+49) 461 – 97892-11
contact@2wcom.com • www.2wcom.com

© 2008 • 2wcom and the 2wcom logo are registered trademarks of 2wcom in Germany and/or other countries.

3. Installation

This section gives you basic information for using the software package on the Windows 2003 Server.

System Requirements

For running the OCTO IP software on a system, a PC with Windows 2000, XP, Vista or a Windows 2003 Server is assumed. Under this server the package runs as a service under operating system control without human intervention, i.e. it can be configured to start automatically at launch time. This way of operation is suited for a clustered server farm as well. The disk space needed for the OCTO IP software with the optional software “Encoder-Check” is less than. 5 MBytes. The software package itself does not need any special privileges, except the access to the different communication ports, i.e. serial RS-232C lines and/or TCP/IP must be ensured, and write access must be granted for logging. When running the software, several log files can be generated; the occupied size of the log files depends on the number of open connections, message intervals, and the specified lifetime of the log-data. For a more detailed description of the log files please refer to the logging section in this document in the appendix on page 37.

For operating the OCTO IP server software, a normal directory structure is assumed, i.e. the operating system is installed on drive C: , and e.g. data and log files are stored to drive D:.

Software Overview

The software that is operating the RDS encoders consists of several packages. Two packages are located on the Win2k3 server and run as a service:

• 2wcom OCTO IP Server
• 2wcom Encoder-Check (optional)

Two additional packages exist for each encoder and are for configuration purposes only. These are:

• 2wcom Arcos Basic (only for 2wcom C0x RDS encoder)
• 2wcom C04 Set-Up (only for 2wcom C0x RDS encoder)

These packages can be used to assign actions on pins, to set Level & Phase, or set the IP-address of one encoder, etc. Please refer to the delivered encoder documentation for more detailed information.

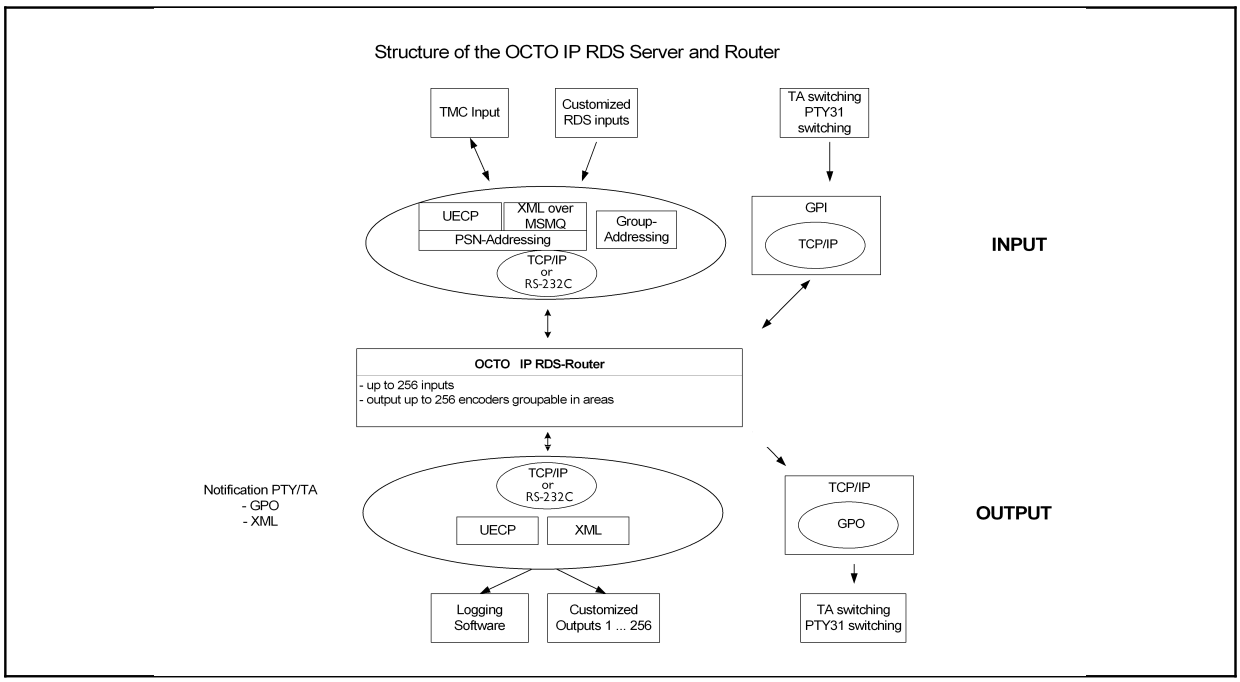
Besides the already named packages, 2wcom Arcos Network (optional) can be provided as a central tool for configuring all 2wcom C0x RDS encoders in the network. This software package builds a database for the whole encoder environment and sets up data (log/config-files) for the Encoder-Check with UECP commands and a different database with a complete RDS set-up for each encoder.

The last package is the Studio Control Software 2wcom Arcos Studio (optional). All the software packages communicate over TCP/IP with various protocols.

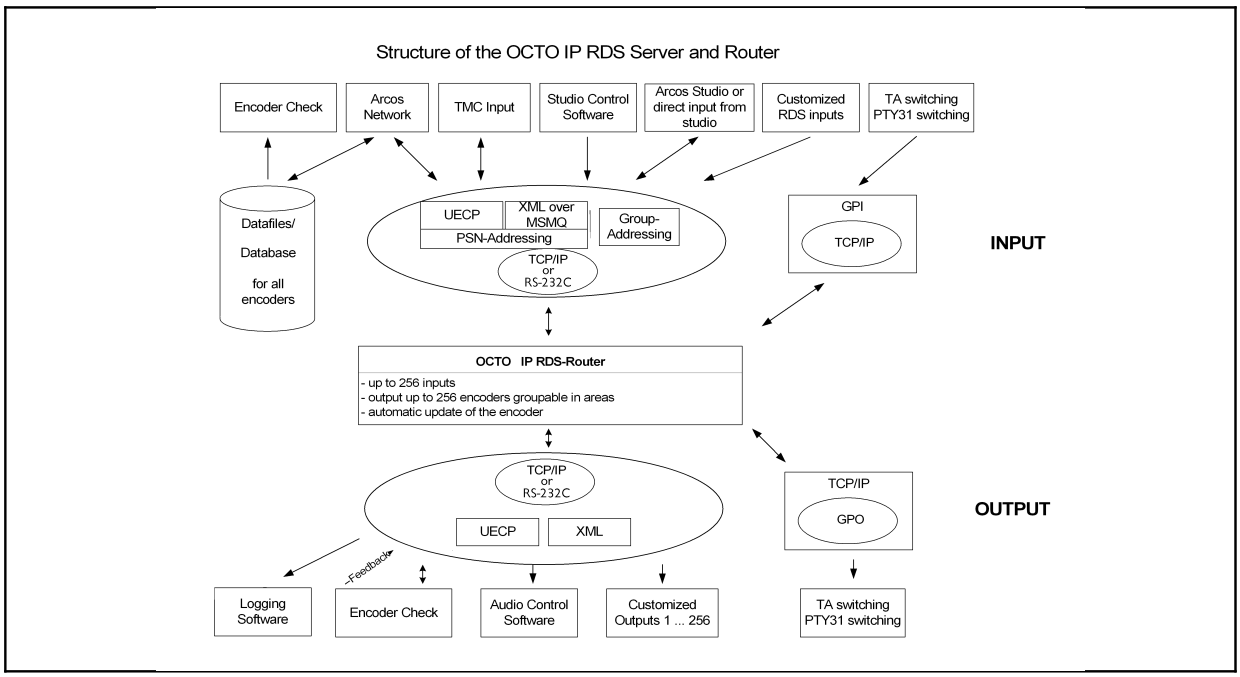
4. General operation

Principle operation of OCTO IP-Server

The OCTO IP software is the central tool and can be configured to start automatically through the operating system when launched. This should be done, if every configuration is set up with 2wcom Arcos Network (optional) properly and tested. The generic approach of operation with the OCTO IP management software is depicted in the following diagrams.



Structure without optional 2wcom software tools



Structure including optional 2wcom software tools

The complete communication and routing takes place under TCP/IP or serial RS-232C (or data from any Windows COM port). It is possible to address each RDS encoder with a PSN, the group name, or the UECP address. The address scheme provided is pure physical addressing of boxes.

For TCP/IP, a failsafe server structure has to be ensured with unique IP addressing and port addresses, e.g. a cluster under Microsoft Windows 2003 Servers.

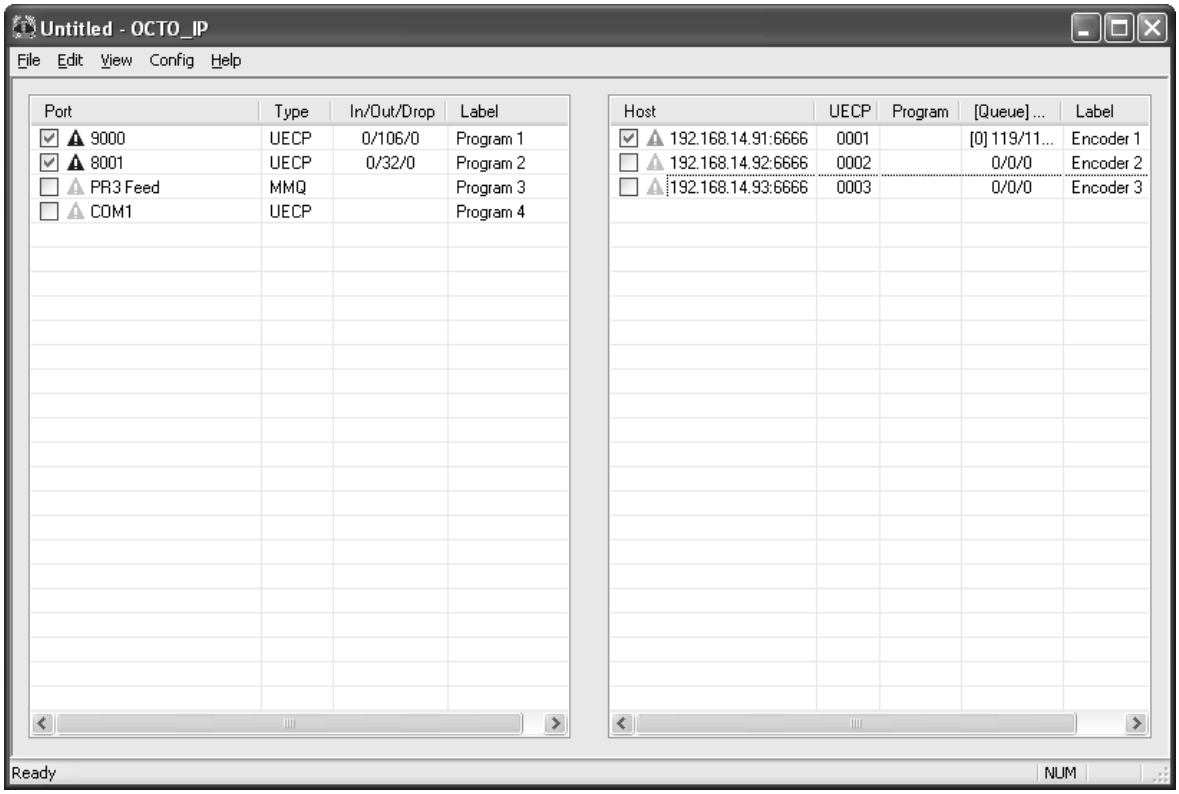
The GPI/O is handled over TCP/IP too, so no necessity for serial RS-232C lines exists for this purpose.

In total it is possible to operate 256 ports/connections, sufficient hardware support assumed. The actual plan foresees 128 outputs (max. number of ports/connections depends on available hardware and system performance).

The protocol is based on UECP 6.0, and for special task data via IP network is encapsulated in XML frames.

OCTO-IP I/O Overview

If OCTO-IP is started with the graphical user interface, a routing overview is shown. On the left, the configured data inputs are listed. On the right, the configured outputs are listed.



New input / output:	Right click in the left list for generating a new input entry, or in the right list to generate a new output entry. Then click on "New Input" or "New Output" respectively. The functions of the appearing input / output properties window are shown on the next pages.
Edit input / output:	Right-click the specific entry in the left list for editing an existing input entry, or in the right list for editing an existing output entry. Then click on "Edit Input" or "Edit Output" respectively. Alternatively just double-click the existing entry. The functions of the appearing input / output configuration window are shown on the next pages.
Port Log for output:	Right click the specific output entry in the right list, you want to see a Port Log for. Then click on "Port Log". Then a Port Log window shows the UECP hex data of this output.
Activate/Deactivate input or output.	Activate the checkbox of the specific input or output entry. The color of the corresponding exclamation mark symbol shows the state of the connection: blue: Tool tries to connect. green: Tool established connection successfully. red: Tool cannot establish connection. TA tag on output symbol: TA flag of the corresponding RDS encoder is activated.
Port / Type / In/Out/Drop / Label	"Port" shows the selected input connection. (IP-port, COM-port, MMQ Label, GP I/O IP->Pin). "Type" shows the selected input type (UECP, MMQ, GP I/O). "In/Out/Drop" shows a frame count of "Received Frames, Sent Frames, Dropped Frames", "Label" shows the defined input label.
Host / UECP / Program / [Queue] … / Label	"Host" shows the selected connection for output (IP address+IP port or COM port), "UECP" shows the selected UECP encoder address, "Program" shows the selected Group(s), "[Queue]…" shows a frame count of "[Frames in wait loop]/Sent Frames/ Received Frames/Dropped Frames", "Label" shows the defined output label.

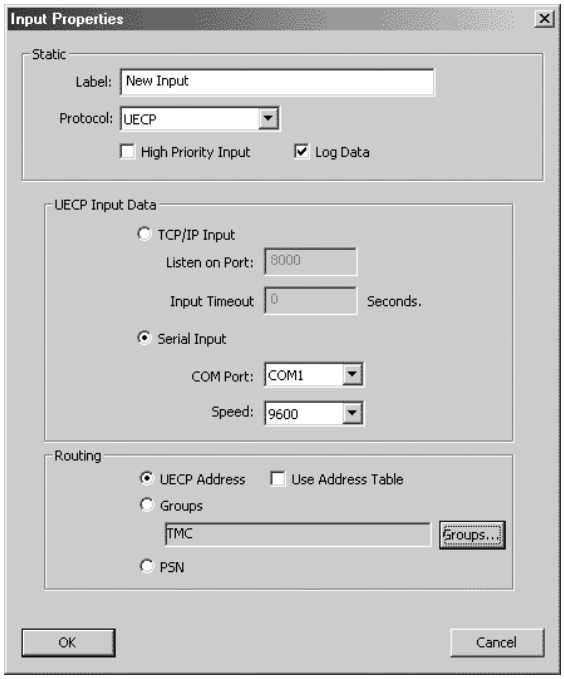
OCTO IP-Input Settings

The OCTO IP server can have several connections (queues) simultaneously to your system management server. For connections via IP network, distinguished by different port numbers; for connections via serial RS-232C distinguished by the COM port numbers. There can be one queue for every port. The queues of multiple ports can be multiplexed and routed to a single output. The advantage of this concept is a faster processing, because the input queues are running in parallel, the commands/actions in one queue are serialized. XML input is converted into UECP automatically before processed; the input of XML is based on Microsoft Message Queues (MMQ).

The “Input Properties” screen content depends on the selected protocol type, except for the area “Static”. The different screens are explained below and on the next page.

Screen “UECP“:

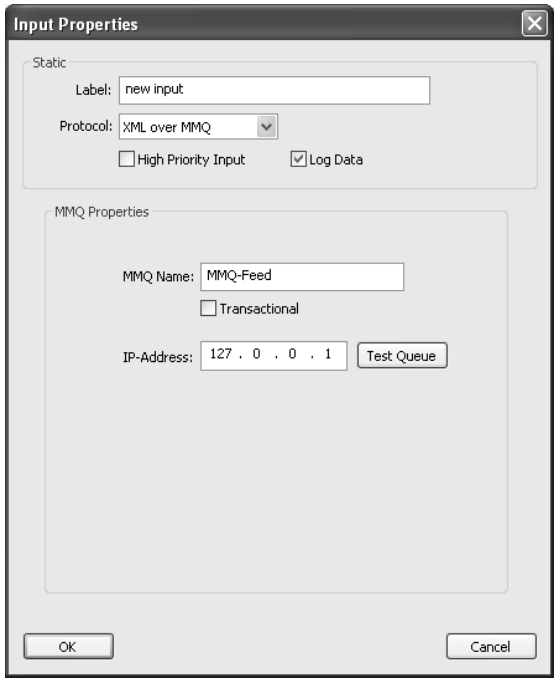
Label:	Enter a short name for the input queue to have a better readability.
Protocol:	Select the data protocol to be used (UECP, XML over MMQ, GP I/O, TEC)
High Priority Input:	Activate checkbox to have a high priority on the data of this input. This is to ensure that the data of this input makes it to the output, even in case of data output overload.
Log Data:	Log the data, received on this input. The log is hourly, locally stored as a non size-limited hex-dump file.
TCP/IP Input	Select to use the IP-network input.
Listen on Port:	Enter the IP-port to be used.
Input Timeout	Enter the IP-port timeout.
Serial Input	Select to use the serial RS-232C input.
COM Port:	Select the serial RS-232C COM port number to be used.
Speed:	Select the serial RS-232C COM port baud rate to be used.
UECP Address	Select to route the data by UECP address.
Use Address Table	This check box indicates if any protocol conversion should be processed (UECP to TEC or TEC to UECP). See ‘Address Table’ for details. If this checkbox is unchecked, no data conversion is done,
Groups	Select to route the data by specified groups. Click on button [Groups…] and activate the required checkboxes. These groups can be initially setup in the file “\config\OCTO_IP.ini”, contained in the installation folder.
PSN	Select to route the data by the encoder PSN.



	Please note, that for PSN routing, it is necessary to setup the PSNs of your encoders (format: PSN_0*hexvalue*) as groups in the OCTO_IP.ini file.

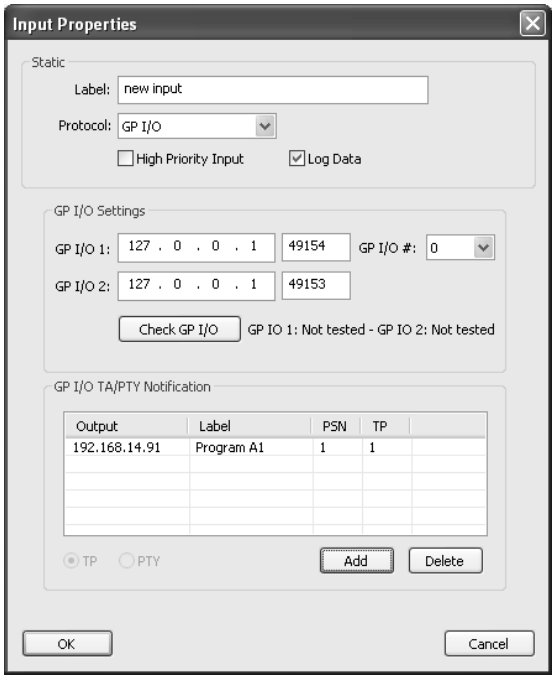
Screen “XML over MMQ”:

Static	See page 10 for the details of the “Static” area.
MMQ Name:	Enter a short name for the XML over MMQ input.
Transactional	Activate checkbox if the MMQ service requires transactional data.
IP-Address:	Enter the IP-address of the server with the MMQ service.
Test Queue	Click to send test data to the MMQ server.



Screen “GP I/O”:

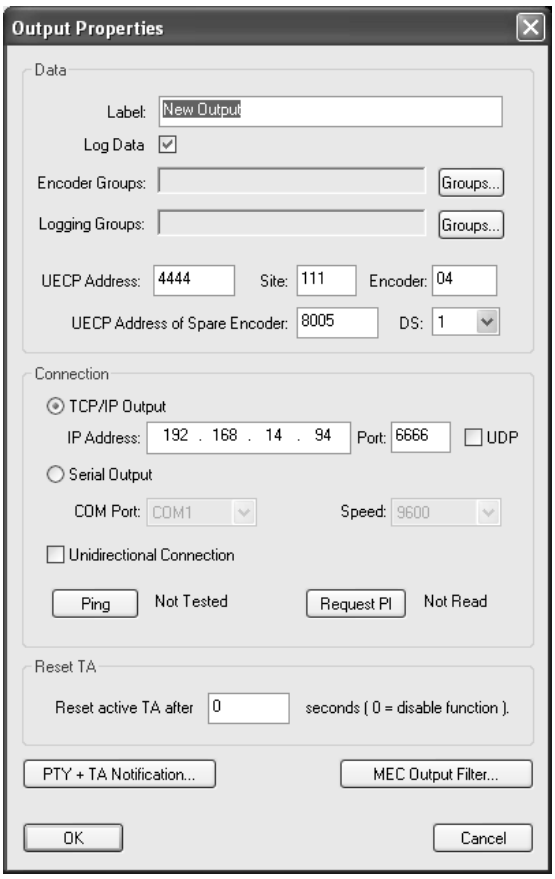
Static	See page 10 for the details of the “Static” area.
GP I/O 1:	Enter the IP-address of the first 2wcom GP I/O box and the corresponding IP-port.
GP I/O #:	Select the GP I/O pin to be used.
GP I/O 2:	Enter the IP-address of the backup 2wcom GP I/O box (uses the same I/O pin #).
Check GP I/O	Click to check the IP network connection to the 2wcom GP I/O boxes.
TP	Activate radio button if you want the configured GP I/O pin to trigger a TP notification. The destination of the notification can then be configured via the [Add] button.
PTY	Activate radio button if you want the configured GP I/O pin to trigger a PTY notification. The destination of the notification can then be configured via the [Add] button.
Add	Click to select the destination with the affected PSN and PTY number of the selected TP/PTY notification.
Delete	Click to delete the selected TP/PTY notification entry.



OCTO IP-Output Settings

Up to 128 output / communication ports can be configured to operate over TCP/IP. The max. number of RS-232C ports depends on your system. Each IP output has a different IP-address with an assigned port number. For every type of operation it has to be ensured that the software has privileges and rights to use the communication interfaces. The specification for the principle data for an output definition is:

Label:	Enter a short name for the output queue to have a better readability.
Log Data	Activate checkbox to log the data of this output.
Encoder Groups: Groups…	Display of the routing groups. Select the routing groups. For PSN routing, select the PSN groups that correspond to the PSN of your encoders. OCTO_IP then scans the incoming UECP data for a PSN that matches a configured PSN group name and routes it via the configured IP address or COM port to the corresponding encoder. These group names need to be configured in the OCTO_IP.ini file (see also page 39).
Logging Groups: Groups…	Display of the logging groups. Select the logging groups. The logging groups (e.g. LogData1) can be used to forward the UECP data to separately configured outputs.
UECP Address	UECP address of the encoder (in hex).
Site: / Encoder:	UECP address, split up in site and encoder number (in hex). Alternative input method to "UECP address".



UECP Address of Spare Encoder: / DS:	UECP address of the spare encoder and active data set for proper rerouting of dynamic data to the spare encoder.
TCP/IP Output	Select to use the IP-network output.
IP Address	IP-address of the output destination.
Port:	IP-port number of the output destination.
UDP	Activate checkbox if UDP instead of TCP should be used for the IP connection (then checkbox "Unidirectional Connection" is activated automatically).
Serial Output	Select to use serial RS-232C input.
COM Port:	Select the COM port to be used.
Speed:	Select the data baud rate to be used.
Unidirectional Connection	If this checkbox is activated, there is only data-flow from an input to the output (no responding traffic). The only exception are acknowledge frames, sent by OCTO IP to the sending application on the input, if PSN/Group routing is used while one input is routed to more than one output.
Ping	Check the configured IP network connection to the encoder.
Request PI	Check the PI of the connected encoder.
Reset active TA after	Enter delay time to activate the TA reset.

PTY + TA Notification…	See page 14 for details.
MEC Output Filter…	See page 15 for details.

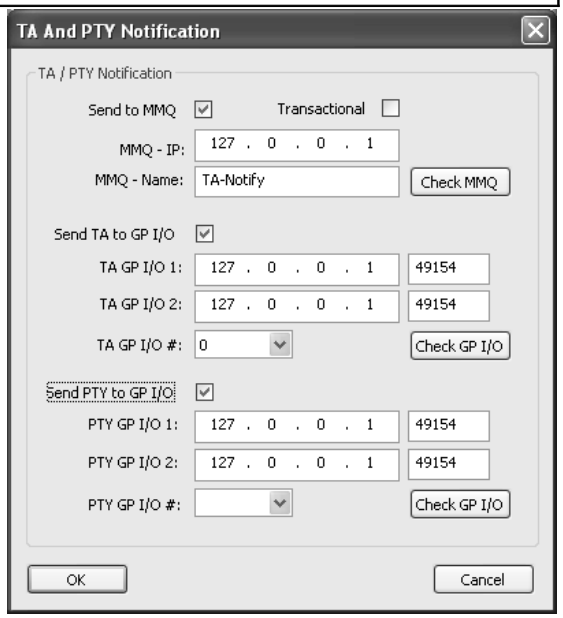
PTY + TA Notification Configuration

The TA flag and the PTY31 can be signaled into your system management server and defined in XML, with the specification given in the appendix. As mentioned in the input section, the XML is also realized to work with Microsoft Message Queues when operating as an output, because the only connection is into your system management server.

For GPO two GPI/O modules can be specified for setting the defined output pin signaling TA or PTY31.

The general-purpose outputs (GPOs) are handled with an optional hardware from 2wcom over TCP/IP.

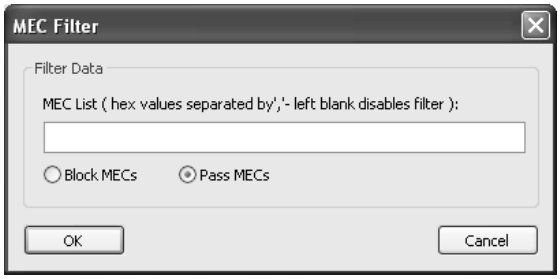
Note: The OCTO IP server can send TAs to the encoders and signal this into your system management server. The encoders will instantly put TA on air (they have a socket connection, i.e. TCP) and the XML packages into your system management server may processed asynchrony at any time on the server (and are queued when the connection is down).



Send to MMQ / Transactional / MMQ-IP: / MMQ-Name:	For Microsoft Message Queues with outgoing TAs and PTY31, an IP-address and queue name is required to send the appropriate XML message, transactional or non-transactional.
Check MMQ	Check the Microsoft Message Queue connection.
Send TA to GP I/O TA GP I/O 1: TA GP I/O 2: TA GP I/O #:	For GPO two GPI/O modules can be used (activate checkbox) and specified (enter the IP address at "TA GP I/O 1:" and/or "TA GP I/O2:") for setting the defined output pin (TA GP I/O #) signaling TA. In case TA is treated, the appropriate encoder for signaling is the one holding the Traffic Program (TP), which is always exactly one per TX station (it is not possible to define more than one encoder/output-queue switching the same output-pin).
Check GP I/O	Check the connection to the specific GP I/O module(s).
Send PTY to GP I/O PTY GP I/O 1: PTY GP I/O 2: PTY GP I/O #:	For GPO two GPI/O modules can be used (activate checkbox) and specified (enter the IP address at "PTY GP I/O 1:" and/or "PTY GP I/O2:") for setting the defined output pin (PTY GP I/O #) signaling PTY31.
Check GP I/O	Check the connection to the specific GP I/O module(s).

MEC Output Filter

The MEC filter can be used to allow only specific MEC commands to be routed through (“Pass MECs”), or to inhibit specific MEC commands to be routed through (“Block MECs”).



Example: 01,02,0A (MEC for PI, PS, RT)

See the UECP (Universal Encoder Communication Protocol) specification for all possible MEC commands.

If the MEC filter has blocked a MEC command, this is logged in the corresponding error log.

Note: Please use the filter function only if required for your application and make sure to configure the MEC filter carefully. To avoid operating error, please keep in mind that the MEC filter for each individual output is activated until the MEC list input field is completely blanked and set by a click on “OK”.

Protocol Conversion

The OCTO IP is able to convert data between TEC and UECP protocol and vice versa. To make the conversion possible some extra information about the Encoder addresses and PSN/TA settings are necessary. The followings commands are converted:

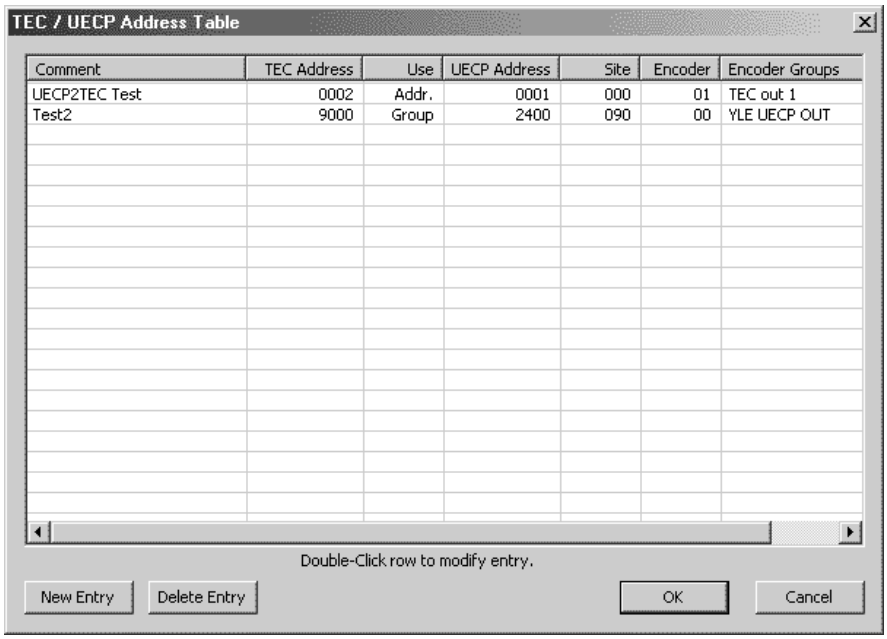
UECP to TEC conversion:

Command	UECP MEC	TEC MEC	Comment
PI	0x01	0x80 / 0x01	TEC EON Frame
PS	0x02	0x80 / 0x02	TEC EON Frame
TA/TP	0x03	0x80 / 0x03 +0x04	TEC EON Frame
DI	0x04	0x80 / 0x08	TEC EON Frame
MS	0x05	0x80 / 0x07	TEC EON Frame
PIN	0x06	0x80 / 0x05	TEC EON Frame
PTY	0x07	0x80 / 0x06	TEC EON Frame
RT	0x0a	0x80 / 0x09	TEC EON Frame
FFG	0x24	0x45	
IH	0x25	0x44	only RDS group type A is convertable
TDC	0x26	0x43	only RDS group type A is convertable
TMC	0x30	0x51	

TEC to UECP conversion:

Command	TEC MEC	UECP MEC	Comment
PI	0x80 / 0x01	0x01	
PS	0x80 / 0x02	0x02	
TP/TA	0x80 / 0x03 + 0x04	0x03	Uses the PSN-TP Tabel
PIN	0x80 / 0x05	0x06	
PTY	0x80 / 0x06	0x07	
MS	0x80 / 0x07	0x05	
DI	0x80 / 0x08	0x04	
RT	0x80 / 0x09	0x0a	
TDC	0x43	0x26	only RDS group type A is convertable
IH	0x44	0x25	only RDS group type A is convertable
FFG	0x45	0x24	
TMC	0x51	0x30	

Address Table



The Address Table is used to convert Encoder addresses (TEC address to UECP address or vice versa). In case you selected the 'Use Address Table' checkbox in the Input Property dialog, the address field of the incoming data is inspected. If the routing mechanism finds an entry for the specific address, the routing information from the Address Table is used to proceed. The Address Table can be used to influence the routing in two different ways:

1.) Substitute the address field with the corresponding address and send the data to the output which matches the substituted address
2.) Substitute the address field with the corresponding address and send the data to an defined output group.

Example 1:
TEC address=1205, UECP address=3421, use address selected

In case of TEC input, the address '1205' is looked up in the 'TEC Address' field of the table. If found, the data is converted to UECP and the address is substituted with the UECP address '3421'. The data is send as UECP to the output with the address '3421'.

Example 2:
TEC address=1205, UECP address=3421, use address NOT selected and the Output Group is set to 'Test Transmitters'

In case of TEC input, the address '1205' is looked up in the 'TEC Address' field of the table. If found, the data is converted to UECP and the address is substituted with the UECP address '3421'. The data is send as UECP to all outputs that belong to the 'Test Transmitters' group.

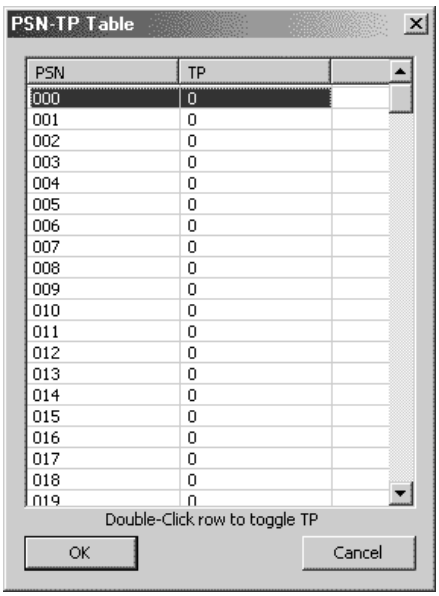
In case the input protocol is UECP the conversion is the same, but the UECP address is looked up and substituted with the TEC address.

Note: With an improper configuration it is possible that one output is sending mixed protocols (TEC and UECP)!

In case of an conversion error, the error log file is updated.

PSN-TP Table

In UECP it is not possible to send a single TA or a single TP information. The TA and TP informations are both needed. The TEC protocol allows to set the TA and TP information individual. To make the TEC to UECP conversion possible, the PSN-TP table is used. When ever a TEC TA is send without the TP information (a complete TEC Frame 80 is parsed to find both information) the missing information is looked up in the PSN-TP table.

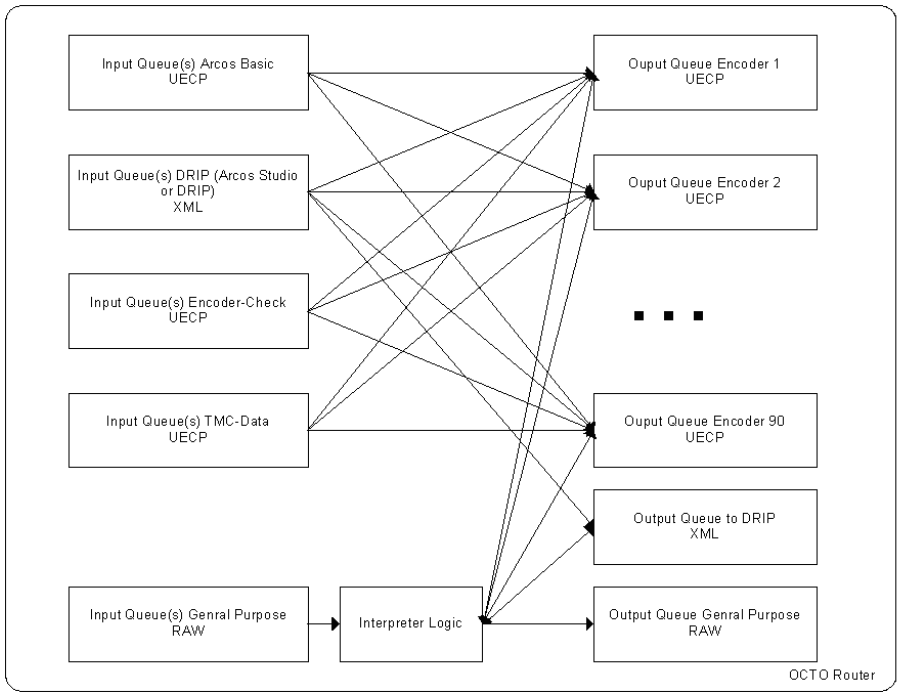


RDS Data Routing

The routing of RDS information is handled with respect to three address types, i.e. addressing by group names, UECP-addressing, and PSN addressing. Additionally it is possible to connect inputs for switching different information via GPI like TA or PTY (optional).

The supported input formats are UECP and XML (XML only via IP-network) formats for PI, PS, PTY, RT and TA from the Studio via e.g. Arcos Studio (optional) that is routed through your system management server. On the output side, the formats UECP and XML for other applications are supported too. The XML format is summarized in the appendix.

For every ingoing feed a queue is implemented, working as a buffer to handle all requests from various applications. Depending on the content and routing information given in the routing table, the data packets are routed to the appropriate outgoing feed into a queue. A scheme of the data routing can be seen below.



What can be seen is the possibility to reach every encoder from every input. The optional general-purpose inputs and outputs are handled with additional interpreter logic.

The provided services for routing are the following:

- Automatic insertion of the UECP address according to the used input (routing table)
- Recalculation of the sequence counter and CRC code
- TA switching, single GPI TA inputs can be used for national or regional TA's. TA will be answered by the encoder and the OCTO IP server will check the answer and send a signal back to the source of the TA (relay output)
- PTY switching will be handled like TA
- Priority handling of PTY31
- Diagnostic of the traffic
- Status function of all traffic of the server

TA/PTY31 GPI-Routing (optional):

TA/PTY31-Input over GPI:
If a TA is activated/deactivated or is recognized over GPI, the appropriate UECP frame is set-up and transmitted into the routing logic described before. An UECP frame with encoder number zero is internally set-up to reach every encoder in the TX station. On successful completion of the UECP command a XML frame is set-up and sent into your system management server, signaling TA active/inactive in the appropriate TX-Site. Simultaneously the corresponding TA active/inactive is set on the GPOs. The same workflow takes place when handling PTY31. The appropriate UECP frame is set-up (this addresses all sites and all encoders), and send through the routing logic.

TA/PTY31-Input over UECP:
The UECP frame is analyzed if it is TA and x-checked if the encoder address is the one that has TP on, i.e. the encoder is the one “owning” the Traffic Program. For fulfilling this workflow, the OCTO IP configuration has to make sure that the encoder holding the TP has the GPO set accordingly. It is even possible to signal this into a defined Microsoft Message Queue for notification with a XML frame. If it is PTY31, it is similar to the handling of TA.

OCTO IP to your system management server

Beside the basic functionality, the OCTO IP server can send an acknowledgement to your system management server with the defined XML frame given in the appendix for:

- TA on
- TA off
- PTY31 on
- PTY31 off

into a Message Queue to your system management server. This is the only functionality going from the server into your system management server. This function can be activated in the Output Properties – TA And PTY Notification window.

GPI/O (optional)

The GPI/Os are handled directly over the OCTO IP server and are based on a TCP/IP socket connection. The supported hardware are the GPI/O modules and are optionally provided by 2wcom. These modules are to be wired to a panel of the broadcast service.

Please note that it is necessary to enter the IP-addresses of all 2wcom GPI/O boxes in the OCTO_IP.ini file. See page 39 for details.

The GP I/O function can be configured in the Input Properties and Output Properties window respectively.

Encoder-Check (optional)

The 2wcom Encoder Check is software, that checks all the encoders via IP-network if they are alive and runs as a service. For this task the software needs as a requirement produced data from Arcos Network (optional). Three different scenarios are possible:

A polled encoder does not respond in a defined time frame (Communication Fault):
In this case a SNMP trap is generated and send to a defined IP-address and IP port number. Additionally a *Test Failure* is signaled over a GPO.

Active data set of the spare encoder does not match (Spare Encoder Active):
All the spare encoders are checked if the active data set is set to the spare one, in case of a mismatch a single SNMP trap is generated.

A polled encoder replies, but the data sent back is not consistent (Encoder Initialize):
In this case the complete RDS data, provided by the configuration of the whole encoder with Arcos Network, is taken and sent to the encoder. After successful transmission a SNMP trap is generated, different to the ones above, to a defined IP-address and port number. If the initialization fails, a SNMP trap for the communication fault is generated.

Besides this, the encoder needs a direct connection over TCP/IP to the OCTO IP server, because the OCTO IP server is the only software being connected directly to each encoder. Nevertheless, the SNMP traps are sent by the Encoder-Check itself. For accomplishing this task, an IP-address and port number needs to be provided in a configuration file, EncoderCheck.ini, that is read by the Encoder-Check service at start-up in a sub-folder named config. Furthermore up to two IP-addresses with port number and an output-pin number for the GPO can be found in the ini file as well. Besides this, the Encoder-Check needs configuration data provided by the OCTO IP server (EC_OCTO IP.cfg) that can be found in the appendix EC Exchange Format on page 36. If any changes were made in the configuration, the Encoder-Check needs to be stopped and started again for reading the configuration data.

To install the Encoder-Check simply copy the package in your desired location. Before installation you should copy the (fresh) configuration folder *exports* from *ArcosNetwork* into the config sub-folder where the EncoderCheck stays and rename it to EncoderInit. These data is used in case of an encoder switchover. Afterwards invoke the executable by putting *–install* behind the name. Other parameters are *start/stop/uninstall* respectively. The package starts automatically after installation, so usually nothing is to do. To make sure everything is set, you can look under services if the 2wcom service shows up.

Arcos Studio (optional)

Arcos Studio is an application that can be installed in every studio on a computer. This software builds a simple Studio Control Software to handle the following RDS information:

• PS, SPS, RT,TA

It is possible to directly address an encoder, or to address a group of encoders. The handling of grouping, as selecting a single encoder, will be done in the OCTO IP server itself.

For this task some configuration data (the grouping) provided in the OCTO IP server must be known and distributed to the studios. The program names are held in a cfg-file, and can be altered for customizing each studio.

The user of Arcos Studio chose one program name (group/encoder) to initiate one of the given actions above, with additional data if necessary.

The software communicates over TCP/IP and uses the XML protocol in the appendix for all the tasks. The communication of Arcos Studio into your system management server takes place over MMQs (Microsoft Message Queues). The configuration of the Message Queue can be done with the application and is held in the registry of the operating system.

In more detail the provided software has the following single window:

The required server name of the host where the Microsoft Message Queue is located to send to, followed by the required name of the MMQ itself is hidden in the *arcos_studio.cfg*. The pop-up for the *Program* is based on data in *arcos_studio.cfg* file too, with program names separated by CRLF, for popping up. The file itself is located in the same directory as the executable and is usually generated by the OCTO IP server. As mentioned before, the OCTO IP server will resolve the name and can address a single decoder or group of encoders.

The pop-up for *Transmissions* has fifteen choices, the numbers from *1* till *15*. The remaining fields are of free choice, with respect to the UECP 6.0 standard, e.g. *PS* will have 8 characters maximal. Pushing the *Send* button will transmit every single item checked in one message into the specified queue to the provided host, which is a machine in your system management server. The last item TA, generates a TA frame with DSN 0 and a PSN defined in the *arcos_studio.cfg* and send this to the input queue. This will be most probably number 4, which should always be the encoder holding the traffic program, i.e. where TP is set.

TA Overview

Initiating a TA is generally possible in two different ways.

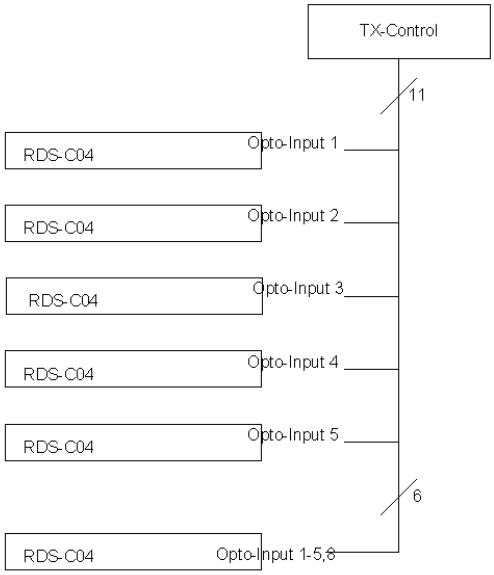
TA over TCP/IP with a XML frame:
This is the normal way of operation and can be initiated from every location e.g. with the Arcos Studio software (optional) distributed over the studios, and an OCTO IP server in-between. It is also possible to use any other initiator, e.g. a TA from Broadcast Service, as long as the TA is encapsulated in the specified XML frame send to the OCTO IP server.

TA wired at the back of the 2wcom RDS C0x encoder:
TA setting and resetting is build with a single line/input connected to the 2wcom RDS C0x encoder, a so called General Purpose Input, GPI, configured as “TA Toggle”. This way of initiating a TA is only possible in a TX site, and has higher priority as the TA over TCP/IP. This means if a TA is set physically over an input line and a TA UECP frame arrives resetting the TA, the TA remains active. This behavior is level sensitive. If the TA over GPI is released, the TA is inactivated only once, this behavior is edge-triggered, and afterwards normal operation takes place.

Note: There is no automatic reset if the TA stays active. The user has to ensure proper operations setting and resetting TA.

Failsafe Hardware Set-Up for a TX-Station (optional)

This function works with the optional 2wcom EncoderCheck. Every station is assumed to have up to five RDS-C04 (the maximum could be seven), plus one spare RDS-C0x, as depicted in the following diagram.



As a general set-up the data transmission to a TX station is organized in a way, that every data transferred to a dedicated encoder is transmitted to the spare encoder. This identical data is put into the encoder corresponding data set number of the spare encoder. Only the IP/UECP-address will be different to address the spare one. As an example sending RT to encoder number three, this RT is rerouted and sent as data set number three to the spare encoder. Working this way ensures that the spare encoder has all the data of the active encoder. The activation of the spare encoder is recognized by the EncoderCheck and signaled into the OCTO IP server for rerouting.

The following failures are possible:

The TX-Control/PLC detects an error, and switches the complete chain:	In this case the PLC set pin eight at the spare encoder inactive, and activate the pin of the failed chain. Due to the fact that the spare encoder has all data sets available, the correct RDS data is instantly on air. This situation is detected afterwards with the requested DSN by the Encoder-Check, and a SNMP trap is generated if the requested DSN differs with the spare one.
The RDS encoder has a DSP communication failure, but is still reachable:	This situation is detected by the PLC, and the PLC switches to the spare encoder. This in turn is recognized by the Encoder-Check, as can be seen in the first scenario.
The RDS encoder has a TCP/IP communication failure:	This failure is detected by the Encoder-Check too. Due to the fact that the Encoder-Check permanently checks for the availability of all the encoders, a not responding encoder will be detected in a short amount of time. In this situation a SNMP trap is signaled by the Encoder-Check. No other action takes place, because it is not sure why the encoder is not reachable. It could be a switch/router or something else, not necessarily the encoder itself. Human intervention is needed.
One additional scenario is worth to be considered.	This is not a failure, but most probably the action taken afterwards. In case of a maintenance, and an encoder is replaced by a new one from of the shelf, this will be detected by the requested PI (what is 0xFFFF in this case) with the Encoder-Check. For the follow-up actions taken see section Encoder-Check on page 21, scenario three, “A polled encoder replies, but the data sent back is not consistent”.

Appendix XML Interchange format

The XML interchange format between the OCTO IP server and your system management server has the following specification automatically generated with XMLSpy.

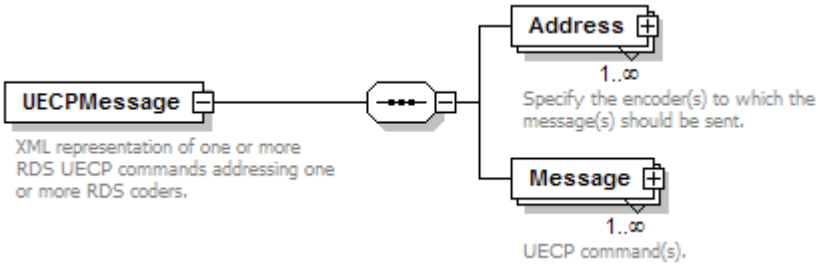
Please note that you can enter the URLs of your own XML definitions in the OCTO_IP.ini file. See page 39 for details.

Scheme **UECPMessage.xsd**

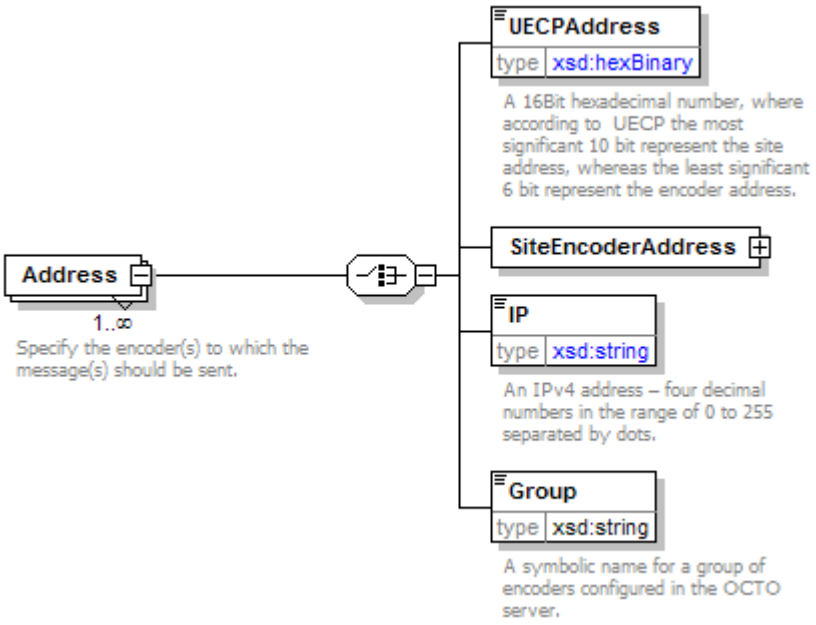
scheme location: **D:\Temp\UECPMessage.xsd**
attribute form default: **unqualified**
element form default: **unqualified**

Elements
UECPMessage

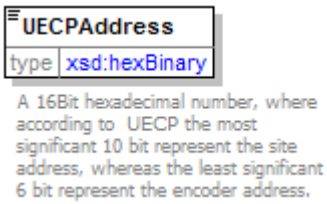
element UECPMessage

diagram	
properties	content complex
children	**Address Message**
annotation	documentation XML representation of one or more RDS UECP commands addressing one or more RDS coders.

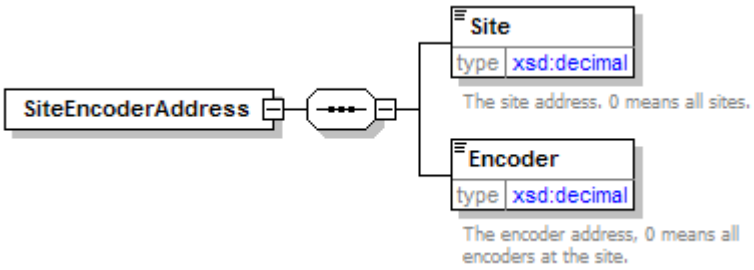
element UECPMessage/Address

diagram	
properties	isRef 0 minOcc 1 maxOcc unbounded content complex
children	**UECPAddress SiteEncoderAddress IP Group**
annotation	documentation Specify the encoder(s) to which the message(s) should be sent.

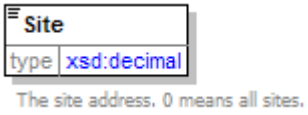
element UECPMessage/Address/UECPAddress

diagram	
type	restriction of **xsd:hexBinary**
properties	isRef 0 content simple
facets	length 2
annotation	documentation A 16Bit hexadecimal number, where according to UECP the most significant 10 bit represent the site address, whereas the least significant 6 bit represent the encoder address.

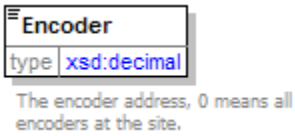
element UECPMessage/Address/SiteEncoderAddress

diagram	
properties	isRef 0 content complex
children	**Site Encoder**

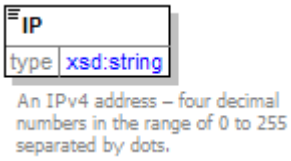
element UECPMessage/Address/SiteEncoderAddress/Site

diagram	
type	restriction of **xsd:decimal**
properties	isRef 0 content simple
facets	minInclusive 0 maxInclusive 1023
annotation	documentation The site address. 0 means all sites.

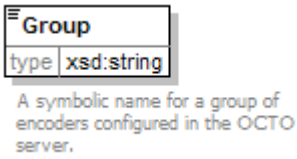
element UECPMessage/Address/SiteEncoderAddress/Encoder

diagram	
type	restriction of **xsd:decimal**
properties	isRef 0 content simple
facets	minInclusive 1 maxInclusive 63
annotation	documentation The encoder address, 0 means all encoders at the site.

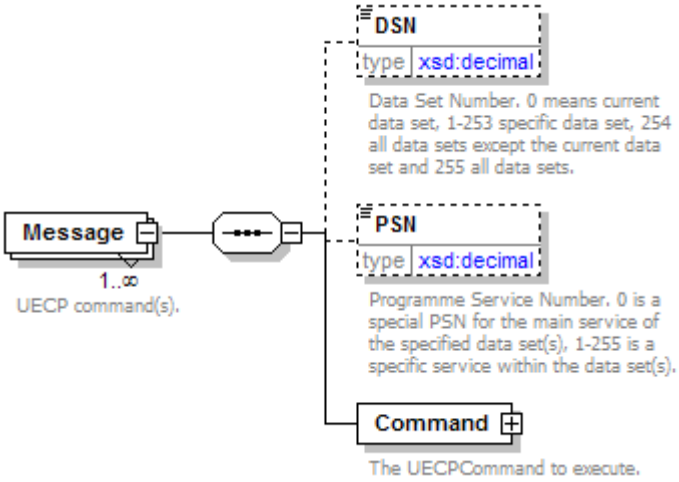
element UECPMessage/Address/IP

diagram	
type	restriction of **xsd:string**
properties	isRef 0 content simple
facets	pattern ((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])\\.){3}(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])
annotation	documentation An IPv4 address – four decimal numbers in the range of 0 to 255 separated by dots.

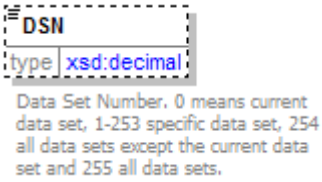
element UECPMessage/Address/Group

diagram	
type	**xsd:string**
properties	isRef 0 content simple
annotation	documentation A symbolic name for a group of encoders configured in the OCTO IP server.

element UECPMessage/Message

diagram	
properties	isRef 0 minOcc 1 maxOcc unbounded content complex
children	**DSN PSN Command**
annotation	documentation UECP command(s).

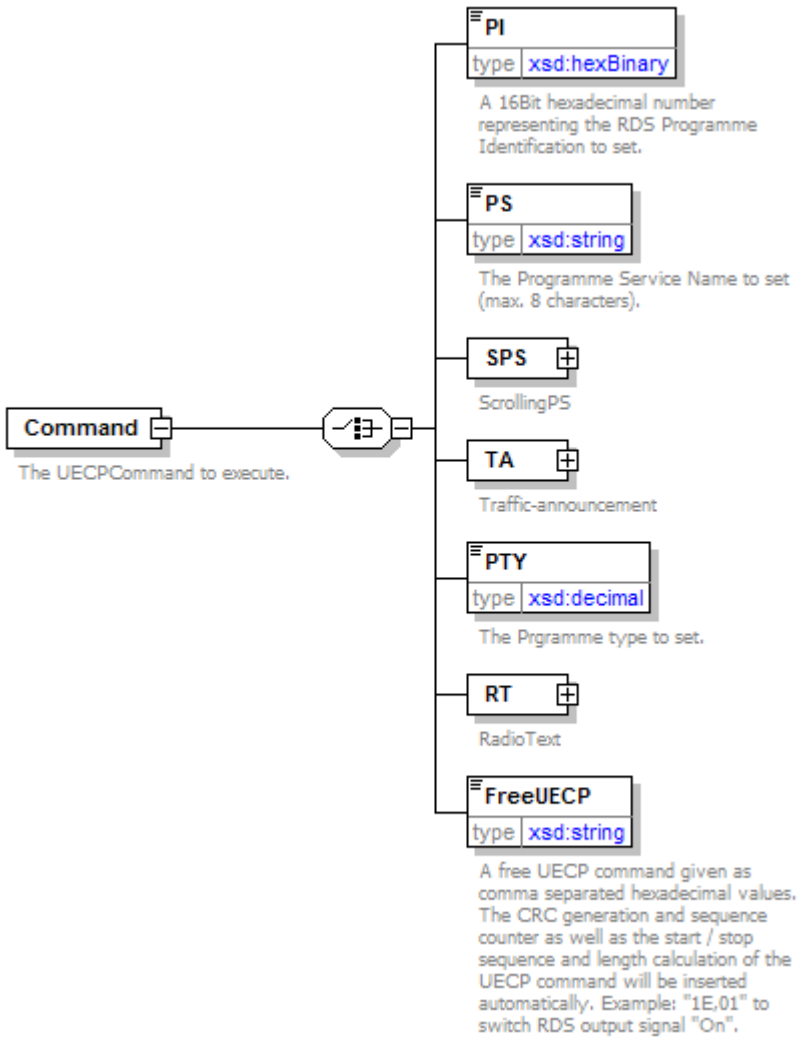
element UECPMessage/Message/DSN

diagram	
type	restriction of **xsd:decimal**
properties	isRef 0 minOcc 0 maxOcc 1 content simple default 0
facets	minInclusive 0 maxInclusive 255
annotation	documentation Data Set Number. 0 means current data set, 1-253 specific data set, 254 all data sets except the current data set and 255 all data sets.

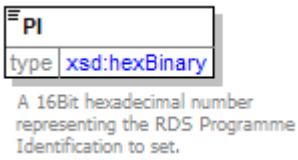
element UECPMessage/Message/PSN

diagram	PSN type xsd:decimal Programme Service Number. 0 is a special PSN for the main service of the specified data set(s), 1-255 is a specific service within the data set(s).
type	restriction of **xsd:decimal**
properties	isRef 0 minOcc 0 maxOcc 1 content simple default 0
facets	minInclusive 0 maxInclusive 255
annotation	documentation Programme Service Number. 0 is a special PSN for the main service of the specified data set(s), 1-255 is a specific service within the data set(s).

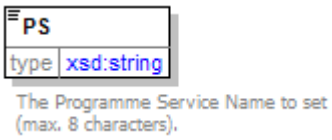
element UECPMessage/Message/Command

diagram	
properties	isRef 0 content complex
children	**PI, PS, SPS, TA, PTY, RT, FreeUECP**
annotation	documentation The UECPCommand to execute.

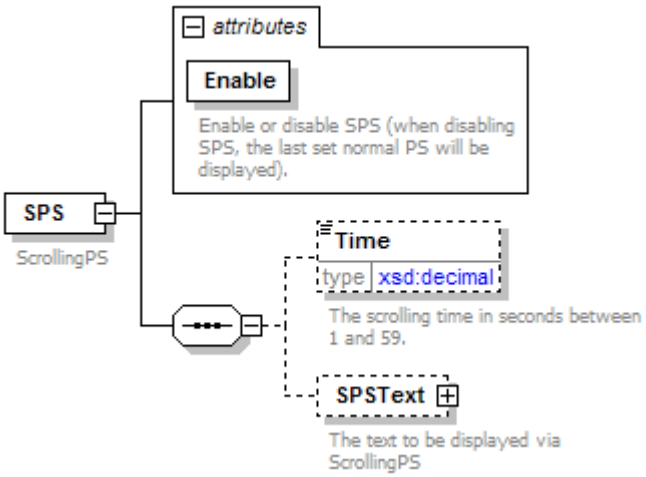
element UECPMessage/Message/Command/PI

diagram	
type	restriction of **xsd:hexBinary**
properties	isRef 0 content simple
facets	length 2
annotation	documentation A 16Bit hexadecimal number representing the RDS Programme Identification to set.

element UECPMessage/Message/Command/PS

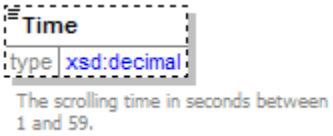
diagram	
type	restriction of **xsd:string**
properties	isRef 0 content simple
facets	maxLength 8
annotation	documentation The Programme Service Name to set (max. 8 characters).

element UECPMessage/Message/Command/SPS

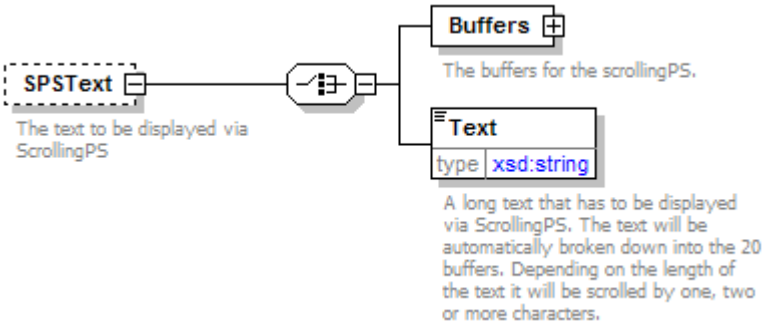
diagram	
properties	isRef 0 content complex
children	**Time, SPSText**
attributes	Name: Enable; Type: **xsd:boolean**; Use: required; Default; Fixed; Annotation documentation: Enable or disable SPS (when disabling SPS, the last set normal PS will be displayed).

annotation	documentation ScrollingPS

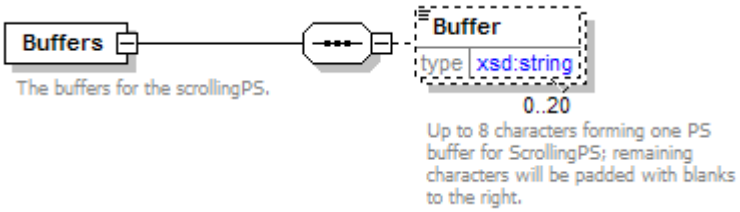
element UECPMessage/Message/Command/SPS/Time

diagram	
type	restriction of **xsd:decimal**
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minInclusive 1 maxInclusive 59
annotation	documentation The scrolling time in seconds between 1 and 59.

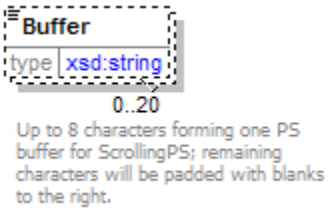
element UECPMessage/Message/Command/SPS/SPSText

diagram	
properties	isRef 0 minOcc 0 maxOcc 1 content complex
children	**Buffers, Text**
annotation	documentation The text to be displayed via ScrollingPS

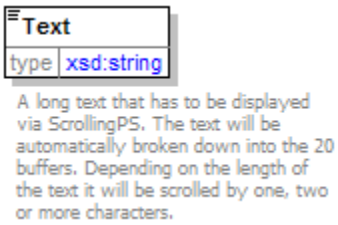
element UECPMessage/Message/Command/SPS/SPSText/Buffers

diagram	
properties	isRef 0 content complex
children	**Buffer**
annotation	documentation The buffers for the scrollingPS.

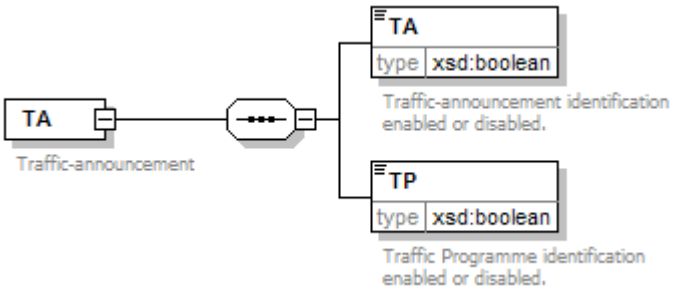
element UECPMessage/Message/Command/SPS/SPSText/Buffers/Buffer

diagram	
type	restriction of **xsd:string**
properties	isRef 0 minOcc 0 maxOcc 20 content simple
facets	maxLength 8
annotation	documentation Up to 8 characters forming one PS buffer for ScrollingPS; remaining characters will be padded with blanks to the right.

element UECPMessage/Message/Command/SPS/SPSText/Text

diagram	
type	restriction of **xsd:string**
properties	isRef 0 content simple
facets	maxLength 160
annotation	documentation A long text that has to be displayed via ScrollingPS. The text will be automatically broken down into the 20 buffers. Depending on the length of the text it will be scrolled by one, two or more characters.

element UECPMessage/Message/Command/TA

diagram	
properties	isRef 0 content complex
children	**TA, TP**
annotation	documentation Traffic-announcement

element UECPMessage/Message/Command/TA/TA

diagram	TA type xsd:boolean Traffic-announcement identification enabled or disabled.
type	**xsd:boolean**
properties	isRef 0 content simple
annotation	documentation Traffic-announcement identification enabled or disabled.

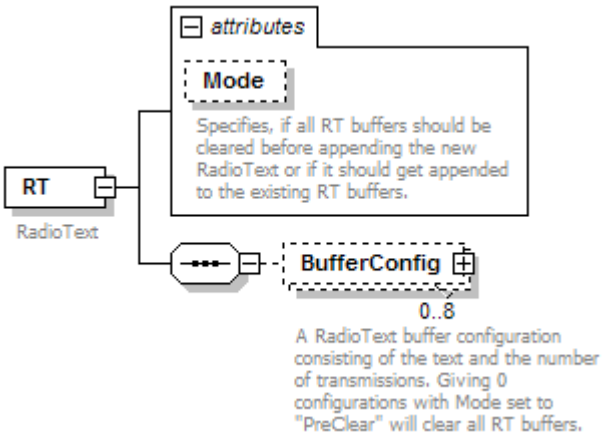
element UECPMessage/Message/Command/TA/TP

diagram	TP type xsd:boolean Traffic Programme identification enabled or disabled.
type	**xsd:boolean**
properties	isRef 0 content simple
annotation	documentation Traffic Programme identification enabled or disabled.

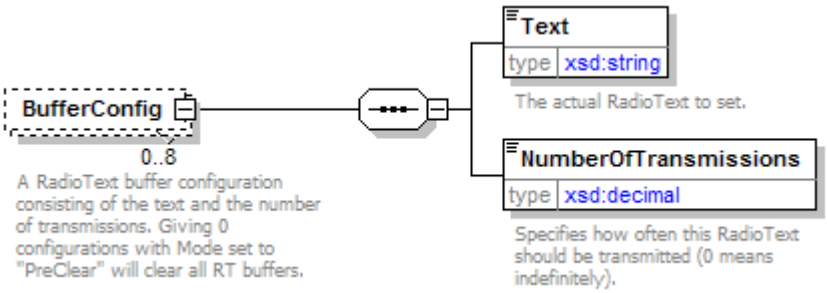
element UECPMessage/Message/Command/PTY

diagram	PTY type xsd:decimal The Prgramme type to set.
type	restriction of **xsd:decimal**
properties	isRef 0 content simple
facets	minInclusive 0 maxInclusive 31
annotation	documentation The Programme type to set.

element UECPMessage/Message/Command/RT

diagram	
properties	isRef 0 content complex
children	**BufferConfig**
attributes	Name: Mode; Type: **derived by: xsd:string**; Use: optional; Default: PreClear; Fixed: ; Annotation documentation: Specifies, if all RT buffers should be cleared before appending the new RadioText or if it should get appended to the existing RT buffers.
annotation	documentation RadioText

element UECPMessage/Message/Command/RT/BufferConfig

diagram	
properties	isRef 0 minOcc 0 maxOcc 8 content complex
children	**Text, NumberOfTransmissions**
annotation	documentation A RadioText buffer configuration consisting of the text and the number of transmissions. Giving 0 configurations with Mode set to "PreClear" will clear all RT buffers.

element UECPMessage/Message/Command/RT/BufferConfig/Text

diagram	**Text** type xsd:string The actual RadioText to set.
type	restriction of **xsd:string**
properties	isRef 0 content simple
facets	maxLength 64
annotation	documentation The actual RadioText to set.

element UECPMessage/Message/Command/RT/BufferConfig/NumberOfTransmissions

diagram	**NumberOfTransmissions** type xsd:decimal Specifies how often this RadioText should be transmitted (0 means indefinitely).
type	restriction of **xsd:decimal**
properties	isRef 0 content simple
facets	minInclusive 0 maxInclusive 15
annotation	documentation Specifies how often this RadioText should be transmitted (0 means indefinitely).

element UECPMessage/Message/Command/FreeUECP

diagram	**FreeUECP** type xsd:string A free UECP command given as comma separated hexadecimal values. The CRC generation and sequence counter as well as the start / stop sequence and length calculation of the UECP command will be inserted automatically. Example: "1E,01" to switch RDS output signal "On".
type	restriction of **xsd:string**
properties	isRef 0 content simple
facets	minLength 2 maxLength 764 pattern [0-9A-Fa-f]{2}(,[0-9A-Fa-f]{2})*
annotation	documentation A free UECP command given as comma separated hexadecimal values. The CRC generation and sequence counter as well as the start / stop sequence and length calculation of the UECP command will be inserted automatically. Example: "1E,01" to switch RDS output signal "On".

Appendix EC Exchange Format

This specification describes the interchange format between data generated by the OCTO IP server and the Encoder-Check (optional). The Encoder-Check only reads the content and uses it for addressing the encoders. The list is a comma-separated list in ASCII. Comments are written with a leading number sign (#) in the first position for a single line.

File:
<*UECP-Address*>,[<IP-Address>],[<Symbolic-Name>],<*PI-Number*>,<*Spare-Encoder*>

Where:
<*UECP-Address*> is a 16bit hexadecimal number and represents the UECP-Address
<IP-Address> is the optional IP-Address of the encoder for error reporting purposes only
<Symbolic-Name> is the optional name of the encoder for error reporting purposes only
<*PI-Number*> is a 16bit hexadecimal number and represents the Program-Identification
<*Spare-Encoder*> is 1 for spare, and 0 for a regular encoder

Example:

```
  1,,,1,0
  2,,,2,0
  5,192.168.237.5,,3,0
#
# spare encoder is for test purposes only, comment this out if not active
#
#3,,,4,1
3,,,4,0
01C3,,city_station1,ABCD,1
```

Appendix Logging

Several different log files can be generated when operating the OCTO IP server software, e.g. for Message Queues, GPI/Os or socket connections. These event logs can be activated for the inputs and the outputs. Error logs are always activated. All the generated log and error files will be written the way *open, append, close*. The files do have the following format and content: (Note: Variable values are *ITALIC).*

Input Log

*.log

File Type	*.log	directory 'log\input\'
File name format	'*dd.mm.yyyy_hh_type_port*.log'	'*dd.mm.yyyy*" is 'day/month/year', 'hh' is full hour, '*type'* is e.g. UECP, '*port'* is the IP or COM port number
Description	Logs the data frames of an input when enabled in the configuration.	
Possible Messages	Some MECs (e.g. TA/PTY/REQ) are placed in readable form, others in hex. At the first position a data direction symbol and a time stamp is placed. "<" means incoming data, ">" means outgoing data.	
Example <14:00:01 REQ 192.1.1.1 'Coder1' FE,00,41,5A,05,17,03,01,00,00,C3,34,FF		
Description Example	At 14:00:01 a REQ was received from IP address 192.1.1.1, configured with the destination label (or group) Coder1 and hex data.	

*.err

File Type	*.err	directory 'log\input\'
File name format	'*dd.mm.yyyy_hh_type_port*.err.log'	'*dd.mm.yyyy*" is 'day/month/year', 'hh' is full hour, '*type'* is e.g. UECP, '*port'* is the IP or COM port number
Description	Logs the error frames (no option, always enabled)	
Possible Messages	e.g. 'No Target with UECP-address'	
Example 14:00:02 REQ - 'No Target with UECP-address:0048' FE,00,48,60,05,17,03,01,00,00,30,FD,01,FF		
Description Example	At 14 :00 :02 a REQ was sent, but the appropriate encoder with UECP address 0048 was not available, the sent data is following.	

Output Log

*.log

File Type	*.log	directory 'log\output\'
File name format	'*dd.mm.yyyy_hh_dest*.log'	'*dd.mm.yyyy*' is 'day/month/year', 'hh' is full hour, '*type'* is e.g. UECP, '*dest'* is the destination IP address or COM port number. The omitted data type is always UECP.
Description	Logs the outgoing data frames of an output when enabled in the configuration.	
Possible Messages	Some MECs (e.g. TA/PTY/REQ) are placed in readable form, others in hex. At the first position a data direction symbol and a time stamp is placed. "<" means incoming data, ">" means outgoing data.	
Example >12:00:00 PI UECP '6667' FE,00,00,18,05,01,00,0A,3E,E8,37,76,FF		
Description Example	At 12:00:00 the data type PI was received from source port 6667 for forwarding to the destination (see log file name). The corresponding data is following.	

*.err

File Type	*.err	directory 'log\output\'
File name format	'*dd.mm.yyyy_hh _dest*.err.log'	'*dd.mm.yyyy*' is 'day/month/year', 'hh' is full hour, '*dest'* is the destination IP address or COM port number.
Description	Logs the error frames (no option, always enabled).	
Possible Messages	e.g. 'Data connection lost'	
Example 12:00:42 Connection lost !		
Description Example	At 12:00:42 the data connection was lost.	

Appendix OCTO_IP.ini

OCTO_IP uses the configuration file “OCTO_IP.ini”, for several settings. The file has the path “\config\OCTO_IP.ini” in the installation folder. This file can be edited by the user with care and contains the following sections:

XML Output	Enter here the paths to your own specific XML definitions.
Groups	Here you can change the default names for the routing groups to your own definitions. For group addressing you can use short group names that represent the RDS encoders and the logging outputs. For PSN addressing you need to use special group names to correspond to the PSN of your RDS encoders. The format is: PSN_0*hexvalue* (e.g. PSN_0A). Use upper case characters for the hex values.
WebIO	Enter here the IP addresses of all possible 2wcom GP I/O boxes. Only then you are able to use them with OCTO-IP.