Document Name: pbsControl IEC 60870-5-104 Master OPC Server user manual

Version: 1.6.0

Date: Q2 2015

Developer: Kamjoo Bayat – Kamjoo.bayat@pbscontrol.com

Introduction

This document describes IEC870-5-104 Master OPC server configuration and using in SCADA systems.

User should know IEC870-5-104 protocol concepts.

We will talk about IEC870-5-104 protocol and concepts as it is possible for this document.

IEC104 stack is developed by ANSI C.

A simple wrapper module is used for linking IEC104 stack to Dot Net CRL.

IEC870-5-104 Master OPC server is running for 30 Min without License. After that you need to restart OPC server.

License is activated through internet. You need to get license key from supplier and activate it from OPC Server help menu.

Installation

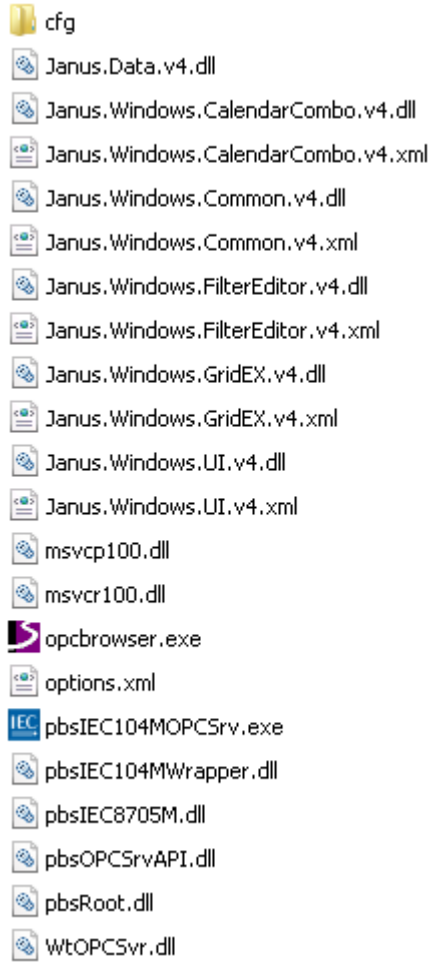
Download IEC870-5-104 Master OPC server from
http://www.pbscontrol.com/pbsProducts/IEC104M_OPCSrv.zip

Unzip IEC104M_OPCsrv.zip at any directory on target system. As an example you will unzip it to c:\pbscontrol\IEC104M_OPCSrv directory.

IEC104 Master OPC server is developed by C# and it needs following components on your system for proper running:

- Dot Net framework Runtime 3.5 SP1
- OPC DA2 .0 Core component distribution
- Visual C 2012 runtime kernel

After unzipping IEC104M_OPCSrv.zip to c:\pbscontrol\IEC104M_OPCSrv directory, you can see following files:



pbsIEC104MOPCSrv.exe: it is main OPC file. For the first time run it as administrator for registering OPC server components in windows registry.

pbsIEC8705M.dll: it is IEC870-5-104 master implementation by ANSI C.

pbsIEC104Wrapper.dll: Wrapper class for using pbsIEC8705M.dll in Dot Net CLR.

WtOPCsvr.dll: OPC DA implementation by C.

pbsOPCSrvAPI.dll: Wrapper class for using WtOPCSvr.dll in Dot net CLR.

OPCBrowser.exe: OPC Browser application for watching OPC server tags.

Options.xml: OPC configuration file

Cfg directory: directory of channels and RTUs configuration files.

All Other files are systematic and must be in OPC root directory.

OPC configuration

IEC870-5-104 Master OPC server is based on channel concept.

You can define many channels, and in each channel you can define many RTUS.

We advise to use one or two RTUs for each Channel.

Each channel will use two CPU threads.

Options.xml file at root directory:

```
<Node>
    <Name>MasterIPAddress</Name>
    <Desc>MasterIPAddress</Desc>
    <Value>192.168.1.102</Value>
</Node>
<Node>
    <Name>TCPIPPort</Name>
    <Desc>TCPIPPort</Desc>
    <Value>2404</Value>
</Node>
<Node>
    <Name>MasterAddress</Name>
    <Desc>MasterAddress</Desc>
    <Value>1</Value>
</Node>
<Node>
    <Name>channels</Name>
    <Desc>channels</Desc>
    <Value>
        <Drv Name="Channel1" Path="\cfg\ch1" Enable="True" />
        <Drv Name="Channel2" Path="\cfg\ch2" Enable="False" />
        <Drv Name="Channel3" Path="\cfg\ch3" Enable="False" />
    </Value>
</Node>
```

MasterIPaddress : IP address of Master OPC server for communication with RTUs .

TCPIPPort : communication port . Default value is 2404.

Masteraddress : IEC870-5-104 originator address .

Channels: definition of communication channels. Each channel has name, Path and Enable attributes.

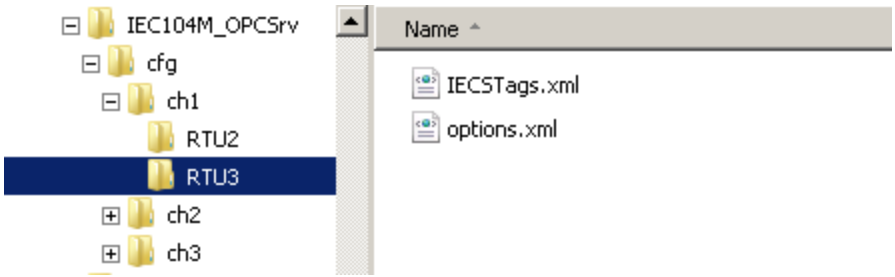
For defining new channel, make a new directory in cfg directory and copy paste RTU files inside new channel directory. Add Channel definition in channels tag in options.xml file.

```
<Node>
    <Name>channels</Name>
    <Desc>channels</Desc>
    <Value>
        <Drv Name="Channel1" Path="\cfg\ch1" Enable="True" />
        <Drv Name="Channel2" Path="\cfg\ch2" Enable="False" />
        <Drv Name="Channel3" Path="\cfg\ch3" Enable="False" />
        <Drv Name="Channel4" Path="\cfg\ch4" Enable="True" />
    </Value>
</Node>
```

Channels with Enable attribute set to True will define in OPC server. You need to restart OPC server when you define or edit channels parameters.

For each channel you should define one directory at cfg directory. Inside channel directory, you should define one directory for each RTU. Inside RTU directory there are two files:

- IECSTags.xml: IEC Tag. It is the same file that is generated by pbsSoftlogic for IEC870-5 slave driver.
- Options.xml: option file for communication with RTU. It is the same file that is generated by pbsSoftLogic IEC870-5 slave driver.

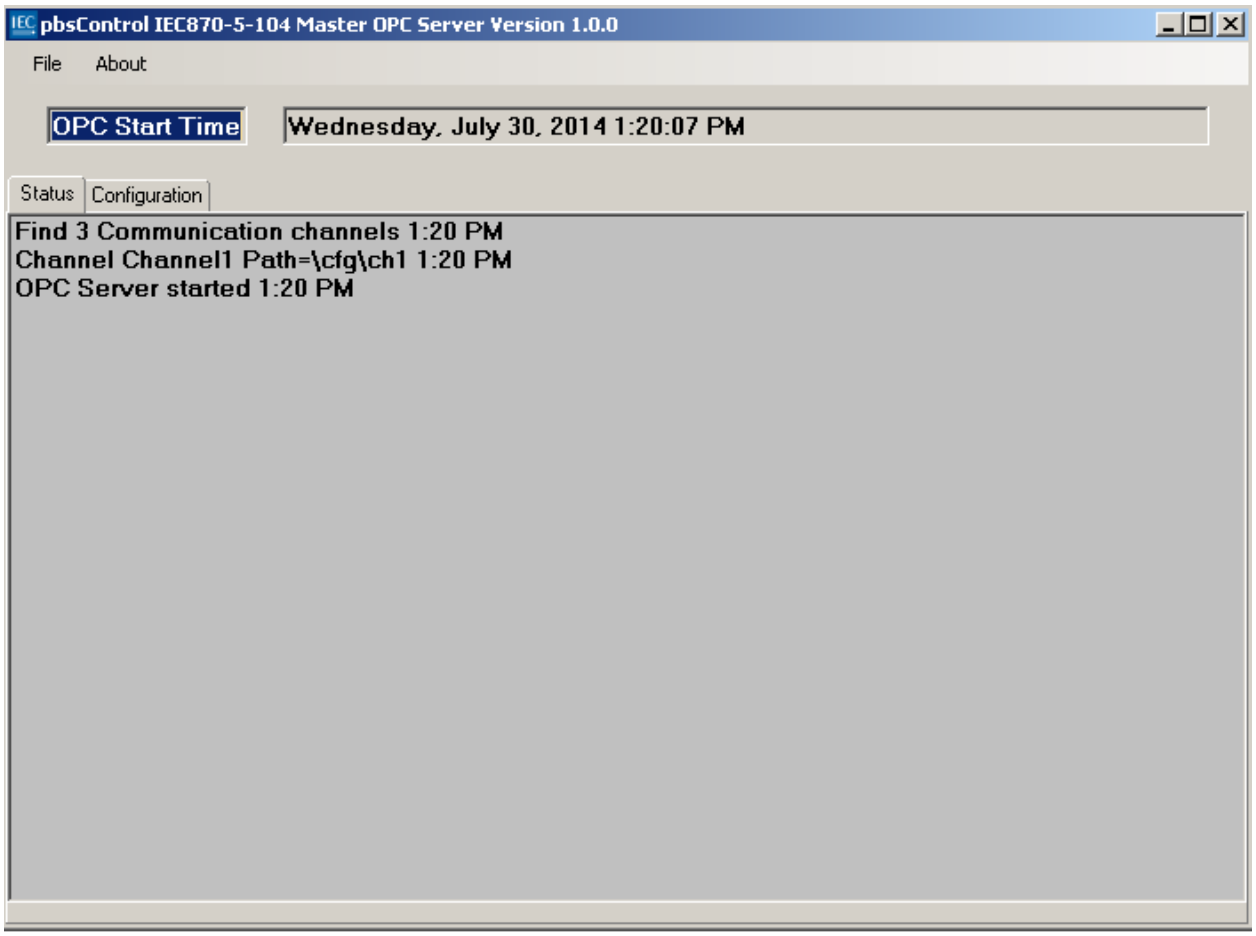


You can set RTU parameters by OPC configuration facility.

For running OPC server, you should run pbsIEC104MOPCsrv.exe application.

Execute pbsIEC104MOPCsrv.exe for first time as administrator for registering OPC components in windows registry.

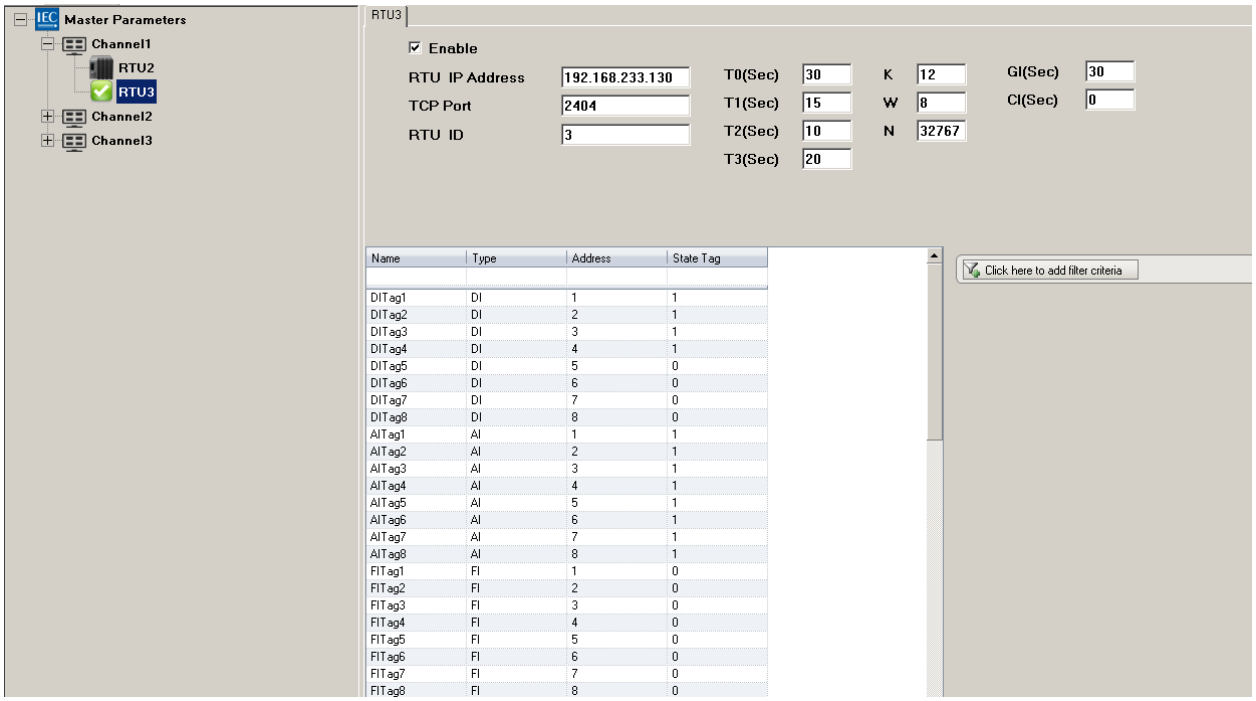
When you run OPC server, you will see following page:



It has two pages:

- Status: shows OPC running status.
- Configuration: editing and saving OPC configuration files.

By clicking on configuration tab, you can see channels and RTU definitions.



RTU configuration setting:

RTU Parameters:

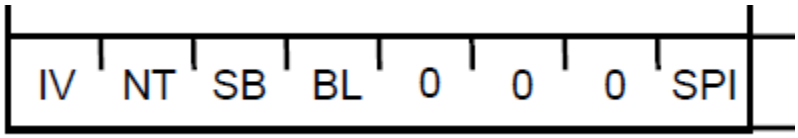
- Enable: When RTU is enable, OPC will try to communicate with RTU, Otherwise just OPC tags will define and OPC server not communicate with RTU.
- RTU IP address : RTU IP address for communication
- TCP Port : communication port default value is 2404
- RTU ID: RTU Common ASDU Address.
- T0 (Sec): Time-out of communication establishment. default value 30 sec
- T1(Sec):Time-Out of send or test APDUs default value 15 sec
- T2(Sec):Time-Out for acknowledges in case of no data messages t2<t1 default value 10 sec
- T3 (Sec):Time-out for sending test frames in case of a long idle state. default value 20 sec
- K: Maximum difference receives sequence number to send state variable. Default value 12.
- W: latest acknowledge after receiving w I format APDUs. default value 8

- N: maximum N(s) Number. Default value is 32767. Max value is 32767.
- GI (Sec): period for sending general interrogation command to RTU. When equal to 0, disabled.
- CI (Sec): period for sending Counter interrogation command to RTU. When equal to 0, disabled.
-

IEC tags:

At bottom part of RTU parameters, you can see IEC Tags grid. Each tag has following fields:

- Name: Tag Name. It must be unique for one RTU.
- Type: IEC Tag Type.
    - DI : Digital input. Single-point information with tag or without tag .
    - DPI : double point information. With tag or without tag.
        - 0 = not permitted.
        - 1 = Off
        - 2 = On
        - 3 = not permitted.
    - AI : analog Input. Measured value, normalized value. With tag or without tag. With state tag or without state tag.
    - FI: measured value, short floating point. With tag or without tag.
    - CNT: Integrated totals. Counters with tag or without tag.
    - DO: single command.
    - DPO: Double command.
    - AO: Set point command, normalized value.
    - FO: Set Point command. Short floating point.
- Address: IEC Tag Address. Start from 1.
- StateTag : if value is 1 , OPC will define state tag .
    - **State tag definition for DI: (1 byte)**



    -

7.2.6.1 Single-point information (IEV 371-02-07) with quality descriptor

SIQ	:=	CP8{SPI,RES,BL,SB,NT,IV}	
SPI	:=	BS1[1]<0..1>	(Type 6)
<0>	:=	OFF	
<1>	:=	ON	
RES = RESERVE	:=	BS3[2..4]<0>	(Type 6)
BL	:=	BS1[5]<0..1>	(Type 6)
<0>	:=	not blocked	
<1>	:=	blocked	
SB	:=	BS1[6]<0..1>	(Type 6)
<0>	:=	not substituted	
<1>	:=	substituted	
NT	:=	BS1[7]<0..1>	(Type 6)
<0>	:=	topical	
<1>	:=	not topical	
IV	:=	BS1[8]<0..1>	(Type 6)
<0>	:=	valid	
<1>	:=	invalid	

○

OV = OVERFLOW/NO OVERFLOW

The value of the INFORMATION OBJECT is beyond a predefined range of value (mainly applicable

to analog values).

BL = BLOCKED/NOT BLOCKED

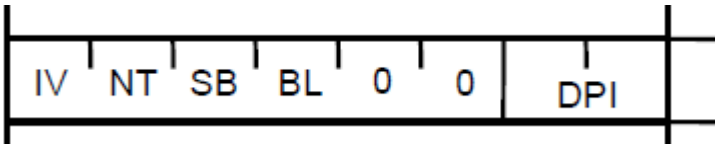
The value of the INFORMATION OBJECT is blocked for transmission; the value remains in the state that was acquired before it was blocked. Blocking and deblocking may be initiated for example by a local lock or a local automatic cause.
SB = SUBSTITUTED/NOT SUBSTITUTED
The value of the INFORMATION OBJECT is provided by the input of an operator (dispatcher) or by an automatic source.
NT = NOT TOPICAL/TOPICAL
A value is topical if the most recent update was successful. It is not topical if it was not updated successfully during a specified time interval or if it is unavailable.
IV = INVALID/VALID
A value is valid if it was correctly acquired. After the acquisition function recognizes abnormal conditions of the information source (missing or non-operating updating devices) the value is then marked invalid. The value of the INFORMATION OBJECT is not defined under this condition. The mark INVALID is used to indicate to the destination that the value may be incorrect and cannot be used.

Intermediate devices may modify the quality descriptors BL, SB, NT and IV.
BL: if an intermediate device blocks the transmission of an information object, it shall assert the quality descriptor BL. Otherwise it shall report the quality descriptor BL as reported from the lower level device.
SB: if an intermediate device substitutes the value of an information object, it shall assert the quality descriptor SB. Otherwise it shall report the quality descriptor SB as reported from the lower level device.
NT: if an intermediate device cannot obtain the value of an information object, it shall assert the quality descriptor NT. Otherwise it shall report the quality descriptor NT as reported from

the lower level device.
IV: if an intermediate device identifies that an information object is not valid, it shall assert the quality descriptor IV. Otherwise it shall report the quality descriptor IV as reported from the lower level device.

Example 1
Suppose that the monitored status of a circuit-breaker is blocked because the field interface is in test mode. In this case, the quality descriptor (BL = 1 “blocked”) will be transferred unchanged through all system levels from the field interface to the controlling station.
Example 2
A substituted value may be assigned automatically or manually to a measured value, for example when the data acquisition is disturbed. This substituted measured value is transmitted to the controlling station with the quality bit SB = 1 substituted.
If the value of an information object is automatically marked with a new quality descriptor due to specific conditions, the quality descriptor may be reset manually or automatically when the conditions change.
If a given information object is normally only reported spontaneously, every change of the quality descriptor initiates a spontaneous transmission. Information objects with a time tag are transmitted with the point of time at which the change of the quality descriptor occurred

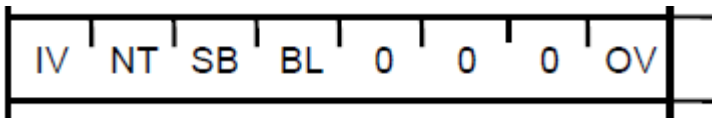
- **State tag definition for DPI: (1 byte)**



7.2.6.2 Double-point information (IEV 371-02-08) with quality descriptor

DIQ		:=	CP8{DPI,RES,BL,SB,NT,IV}	
DPI		:=	UI2[1..2]<0..3>	(Type 1.1)
	<0>	:=	indeterminate or intermediate state	
	<1>	:=	determined state OFF	
	<2>	:=	determined state ON	
	<3>	:=	indeterminate state	
RES = RESERVE		:=	BS2[3..4]<0>	(Type 6)
BL		:=	BS1[5]<0..1>	(Type 6)
	<0>	:=	not blocked	
	<1>	:=	blocked	
SB		:=	BS1[6]<0..1>	(Type 6)
	<0>	:=	not substituted	
	<1>	:=	substituted	
NT		:=	BS1[7]<0..1>	(Type 6)
	<0>	:=	topical	
	<1>	:=	not topical	
IV		:=	BS1[8]<0..1>	(Type 6)
	<0>	:=	valid	
	<1>	:=	invalid	

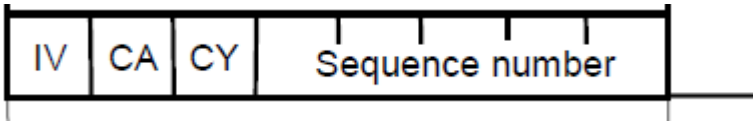
State tag definition for AI ,FI: (1 byte)



QDS		:=	CP8{OV,RES,BL,SB,NT,IV}	
OV		:=	BS1[1]<0..1>	(Type 6)
	<0>	:=	no overflow	
	<1>	:=	overflow	
RES = RESERVE		:=	BS3[2..4]<0>	(Type 6)
BL		:=	BS1[5]<0..1>	(Type 6)
	<0>	:=	not blocked	
	<1>	:=	blocked	
SB		:=	BS1[6]<0..1>	(Type 6)
	<0>	:=	not substituted	
	<1>	:=	substituted	
NT		:=	BS1[7]<0..1>	(Type 6)
	<0>	:=	topical	
	<1>	:=	not topical	
IV		:=	BS1[8]<0..1>	(Type 6)
	<0>	:=	valid	
	<1>	:=	invalid	

OV = OVERFLOW/NO OVERFLOW

State tag definition for CNT: (1 byte)



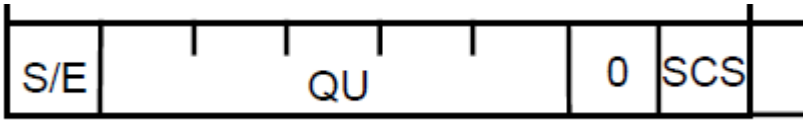
7.2.6.9 Binary counter reading

BCR	:=	CP40{Counter reading, Sequence notation}	
Counter reading	:=	I32[1..32]<$-2^{31}..+2^{31}-1$>	(Type 2.1)
Sequence notation	:=	CP8{SQ,CY,CA,IV}	
SQ	:=	UI5[33..37]<0..31>	(Type 1.1)
CY	:=	BS1[38]<0..1>	(Type 6)
<0>	:=	no counter overflow occurred in the corresponding integration period	
<1>	:=	counter overflow occurred in the corresponding integration period	
CA	:=	BS1[39]<0..1>	(Type 6)
<0>	:=	counter was not adjusted since last reading	
<1>	:=	counter was adjusted since last reading	
IV	:=	BS1[40]<0..1>	(Type 6)
<0>	:=	counter reading is valid	
<1>	:=	counter reading is invalid	

SQ = sequence number
CY = carry
(Counter overflow occurs when the value increments from $+2^{31}-1$ to zero or from -2^{31} to zero)
CA = counter was adjusted
(The counter is considered to have been adjusted if a counter is initialized to some value, for example set to zero or another value at startup).
IV = invalid

Note that CA, CY and IV are only modified when the value is determined. This may be in response to a counter interrogation command or in response to an automatic internal function that performs the counter freeze or freeze and reset command.

State tag definition for single command: (1 byte)

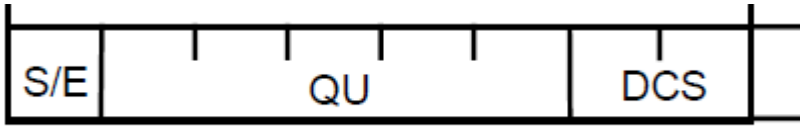


SCO	:=	CP8{SCS,BS1,QOC}	
SCS=Single command state	:=	BS1[1]<0..1>	(Type 6)
<0>	:=	OFF	
<1>	:=	ON	
RES= RESERVE	:=	BS1[2]<0>	(Type 6)
QOC	:=	CP6[3..8]{QU,S/E}	see 7.2.6.26 QOC

7.2.6.26 Qualifier of command

QOC	:=	CP6{QU, S/E}	
QU	:=	UI5[3..7]<0..31>	(Type 1.1)
<0>	:=	no additional definition [5]	
<1>	:=	short pulse duration (circuit-breaker), duration determined by a system parameter in the outstation	
<2>	:=	long pulse duration, duration determined by a system parameter in the outstation	
<3>	:=	persistent output	
<4..8>	:=	reserved for standard definitions of this companion standard (compatible range)	
<9..15>	:=	reserved for the selection of other predefined functions [6]	
<16..31>	:=	reserved for special use (private range)	
S/E	:=	BS1[8]<0..1>	(Type 6)
<0>	:=	execute	
<1>	:=	select	

State tag definition for Double command: (1 byte)



7.2.6.16 Double command (IEV 371-03-03)

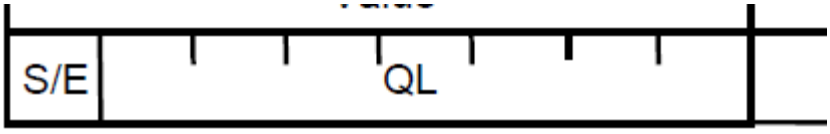
DCO	:=	CP8{DCS,QOC}
DCS=Double command state	:=	UI2[1..2]<0..3>
<0>	:=	not permitted
<1>	:=	OFF
<2>	:=	ON
<3>	:=	not permitted
QOC	:=	CP6[3..8]{QU,S/E}

7.2.6.26 Qualifier of command

QOC := CP6{QU, S/E}

QU	:= UI5[3..7]<0..31>	(Type 1.1)
<0>	:= no additional definition [5]	
<1>	:= short pulse duration (circuit-breaker), duration determined by a system parameter in the outstation	
<2>	:= long pulse duration, duration determined by a system parameter in the outstation	
<3>	:= persistent output	
<4..8>	:= reserved for standard definitions of this companion standard (compatible range)	
<9..15>	:= reserved for the selection of other predefined functions [6]	
<16..31>	:= reserved for special use (private range)	
S/E	:= BS1[8]<0..1>	(Type 6)
<0>	:= execute	
<1>	:= select	

State tag definition for Set Point command, normalized value , scaled value , short float : (1 byte)



QL = 0

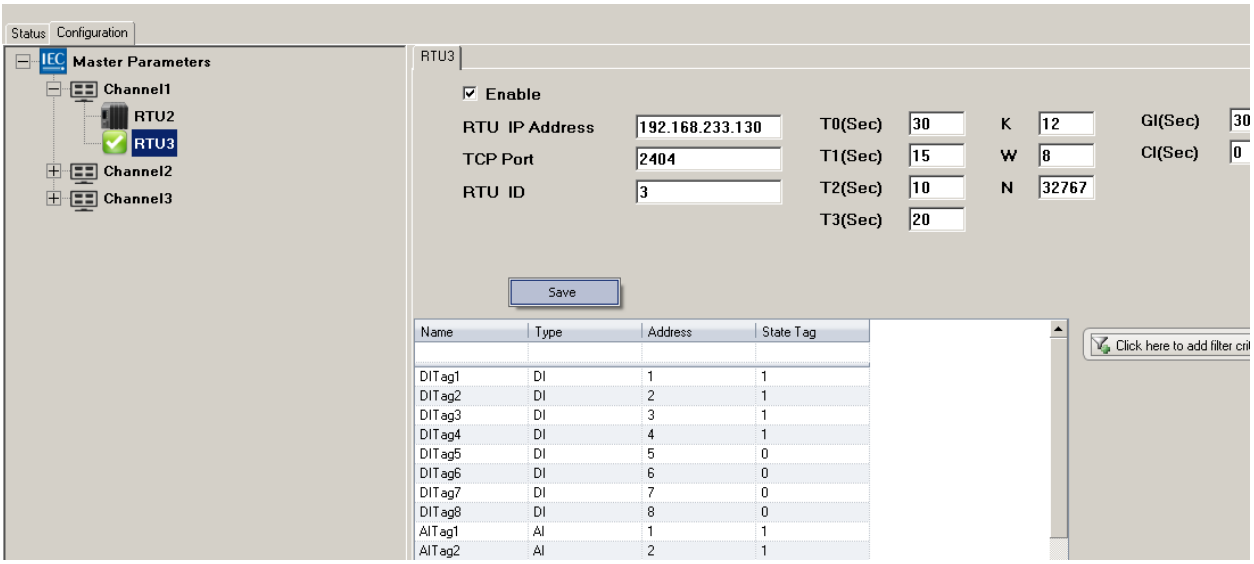
S/E:

0 = Execute

1 = Select

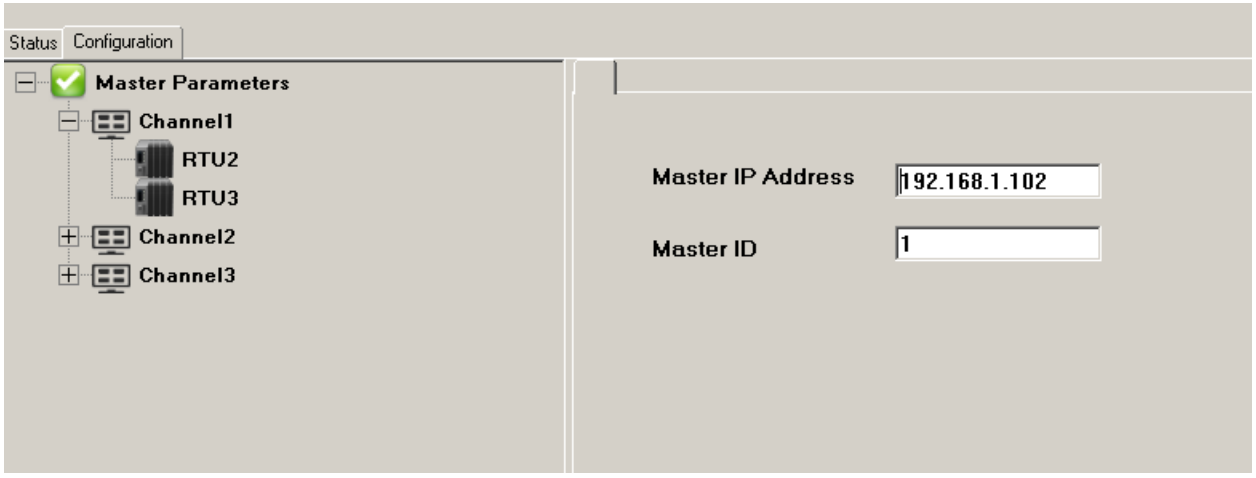
Saving RTU Configuration

Right click on RTU page, then you can save configuration by “Save” menu.



Master Parameter configuration:

Click on “Master parameter” item , you will see following page :



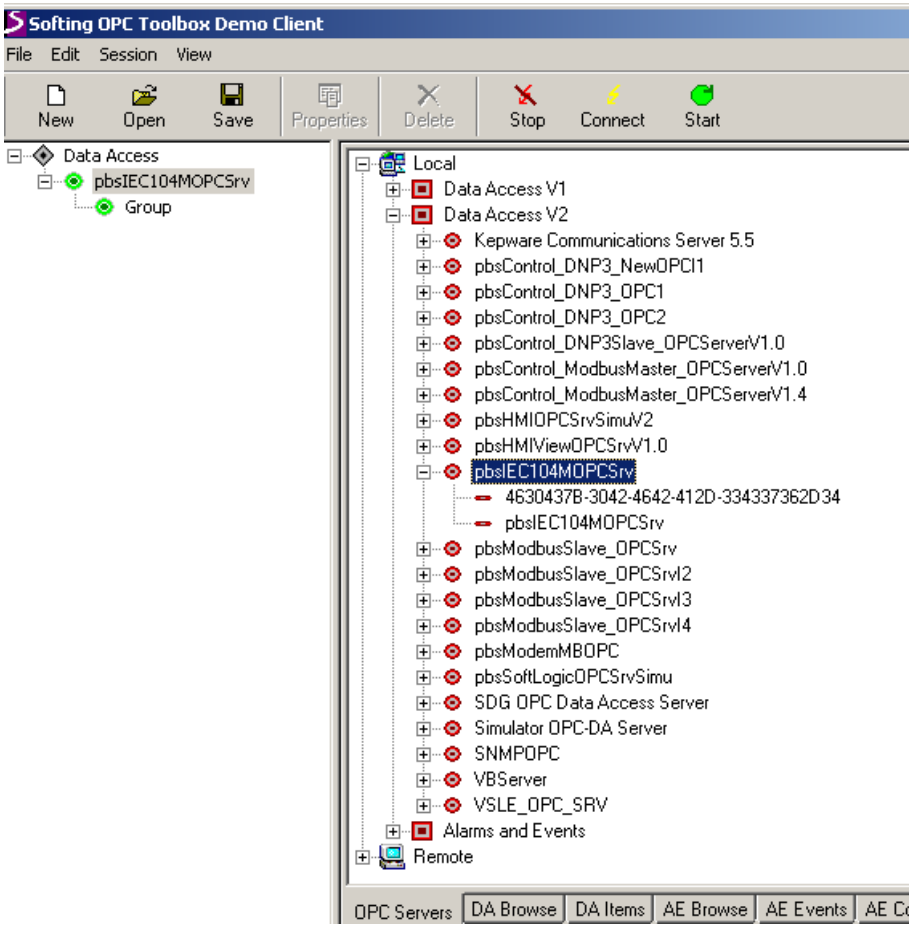
You can change Master IP address and Master ID. For saving values, use right click menu and select “Save” item.

OPC Runtime

You can connect to OPC server by any OPC client or OPC browser.

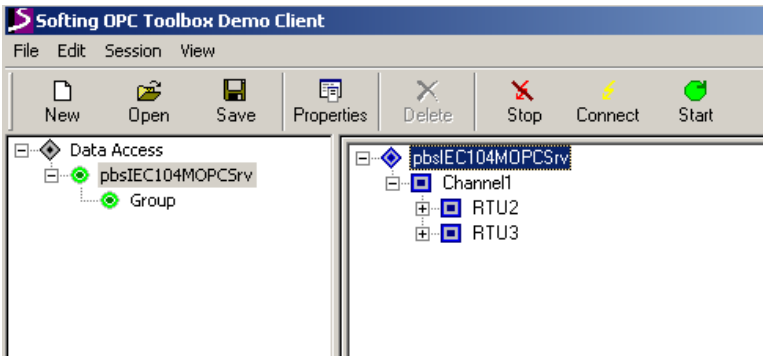
Softing OPC Browser is included to IEC 104 master OPC package.

Run OPCBrowser.exe utility and connect to IEC104 Master OPC server:



IEC 104 Master OPC Server name: pbsIEC104MOPCSrv

Open DA Browse Page. You can see following tag structure in OPC server :

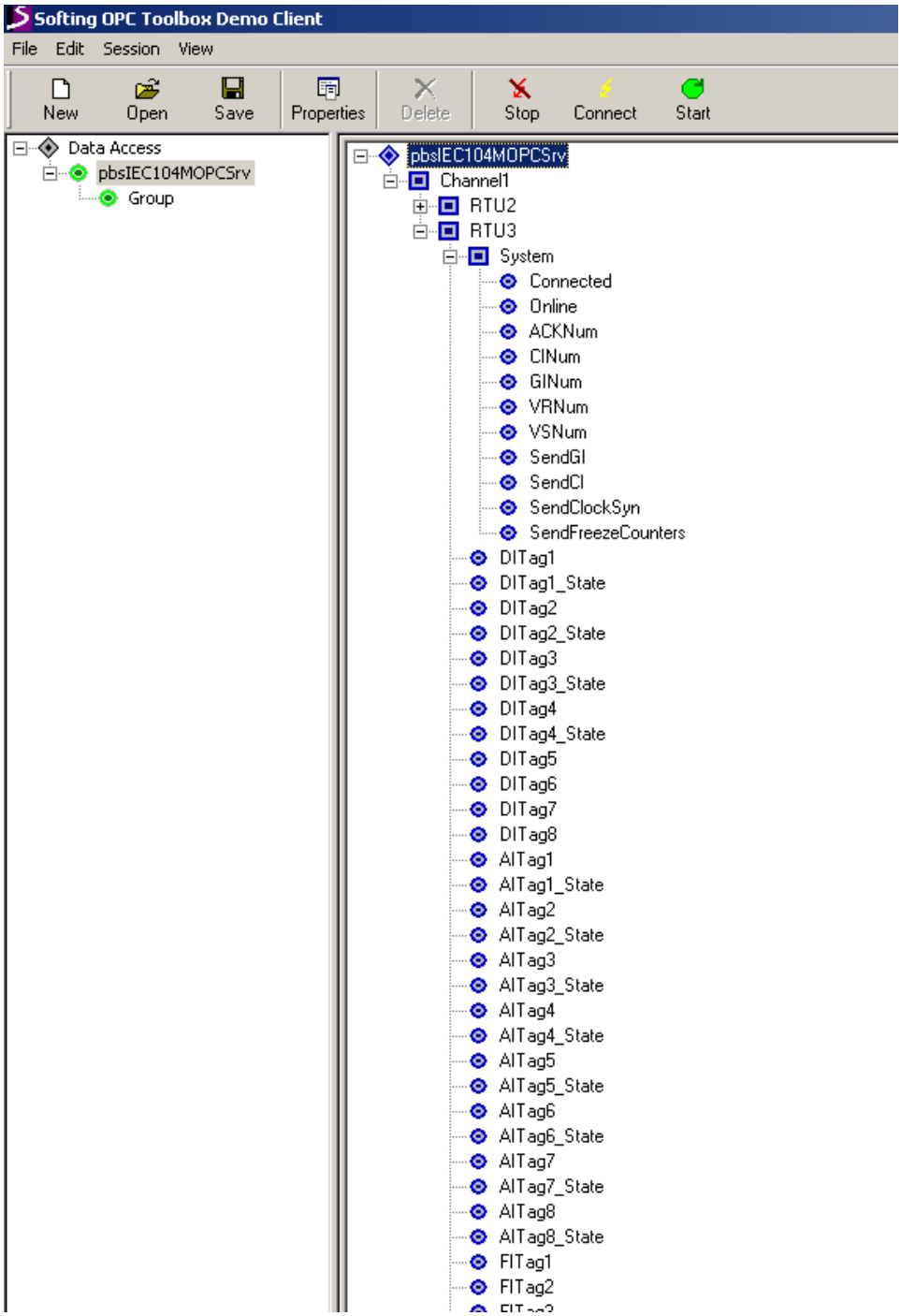


All Enabled channels (Just Enabled channels) are defined in OPC server.

But All RTUs in an Enabled channel are defined in OPC Space.

Disabled RTUs are not scheduled by OPC server.

Open RTU OPC Tag space, you can see following items:



For each RTU we have following system Tags:

Item	Value	Quality
Channel1.RTU3.System.Connected	TRUE	GOOD
Channel1.RTU3.System.Online	TRUE	GOOD
Channel1.RTU3.System.ACKNum	9	GOOD
Channel1.RTU3.System.CINum	0	GOOD
Channel1.RTU3.System.GINum	9	GOOD
Channel1.RTU3.System.VRNum	19	GOOD
Channel1.RTU3.System.VSNum	9	GOOD
Channel1.RTU3.System.SendGI	0	GOOD
Channel1.RTU3.System.SendCI	0	GOOD
Channel1.RTU3.System.SendClockSyn	0	GOOD
Channel1.RTU3.System.SendFreezeCount...	0	GOOD

Connected: When OPC server is connected to RTU, “Connected” Tag will change to True.

Online: When RTU answered to StartDT command of OPC Server, “Online” Tag will change to True.

ACKNum : Shows Number of Acknowledged APDUs by RTU .

CINum : Shows number of Counter Interrogation command that sent to RTU by OPC server.

GINum: Shows number of General Interrogation command that sent to RTU by OPC server.

VRNum : shows Number of Received ASDUs to OPC server .

VSNum : Shows Number of Sent ASDU to RTU .

SendGI: When value is changed from 0 to 1 , OPC server will send General Interrogation command to RTU .

SendCI: When value is changed from 0 to 1 , OPC server will send Counter Interrogation command to RTU .

SendClocksyn: When value is changed from 0 to 1 , OPC server will send clock synchronization command to RTU .

SendFreezeCount : When value is changed from 0 to 1 , OPC server will send counter Freeze command to RTU .

OPC Server will read IECSTags.xml file and define OPC tags based on this file.

For Input tags with StateTag =”True”, OPC server will define one extra tag with ended “_State“signal:

For output Tags with StateTag=”True”, OPC Server will define one extra tag with ended “_StateW”Signal:

State Tag for all signals is integer type.

For all IEC tags (with State tag or without state tag) there is following rule:

- When IV Bit of State tag is 0, OPC tag Quality is set to GOOD.
- When IV Bit of State tag is 1, OPC tag Quality is set to BAD.

Example 1:

DI tag (Single Point Information) with address 1 in RTU has following status:

- IV = 0
- BL = 1
- SPI = 1

OPC state signal is 00010001 = 17

Example 2:

DPI tag (Double Point Information) with address 10 in RTU has following status:

- IV = 1
- BL = 0
- DPI = 2

OPC state signal is 10000010 = 130

Writing Output signals

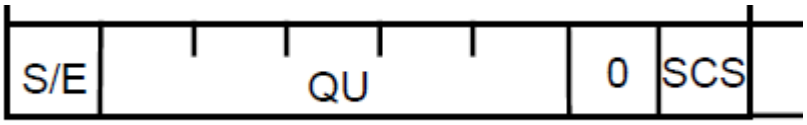
If you define state tag for output signals then you can set tag state value as following:

For all output signals, MSB bit of state tag is S/E Bit.

- S/E = 0 , Execute or direct operation
- S/E = 1 , Select

For single command:

MSB LSB



- QU = 1 short Pulse duration. Short pulse duration is a parameter in RTU .
- QU = 2 long Pulse duration. Long pulse duration is a parameter in RTU .
- QU = 3 persistent Output signal
- SCS = 0 OFF
- SCS = 1 ON

Example1:

DO tag (Single Command) with address 1, Set output to ON (direct Operation - persistent):

Suppose OPC tag name is DOTag1

- Step 1 : State Tag = 00001100 = 12. DOTag1_StateW = 12
- Step2 : DOTag1 = 1

Example2:

DO tag (Single Command) with address 1, Set output to ON (direct Operation – short Pulse):

- Setp1: State Tag = 00000100 = 4. DOTag1_StateW=4
- Step2: DoTag1 = 1

Example3:

DO tag (Single Command) with address 1, Select Tag for persistent output to ON:

- Step1: State Tag = 10001100 = 140. DOTag1_StateW=140
- Step2:DOTag1 = 1

Example4:

DO tag (Single Command) with address 1, Execute Tag for persistent output to ON:

- Step1: State Tag = 00001100 = 12. DOTag1_StateW=12
- Step2:DOTag1 = 1

Example5:

DO tag (Single Command) with address 1, Select Tag for persistent output to OFF:

- Step1: State Tag = 10001100 = 140. DOTag1_StateW=140
- Step2:DOTag1 = 0

Example6:

DO tag (Single Command) with address 1, Execute Tag for persistent output to OFF:

- Step1: State Tag = 00001100 = 12. DOTag1_StateW=12
- Step2:DOTag1 = 0

If you define state tag for Output Signals , then sending command is two steps process .

- First write {TagName}_StateW signal to state value
- Second Write {TagName} to output value

If you did not define State tag , commands are sending as direct operation to RTU and it is one step process.

Above rule are applied for DO, DPO, AO and FO data types.

OPC Server specification

- OPC DA 2.0
- Number of Channels : No Limitation
- Number of RTU per channel: Use only One RTU per channel
- Number of CPU Threads for each channel : 2
- Controlling station definition (Master)
- **Transmission mode for application data**
- Mode 1 (Least significant octet first), as defined in 4.10 of IEC 60870-5-4, is used exclusively in this companion standard.
- **Common address of ASDU :** Two octets
- **Information object address :** Three octets
- **Cause of transmission:** Two octets (with originator address). Originator address is set to zero if not used
- Length of APDU :253
- **Selection of standard ASDUs**
  - <1> := Single-point information M_SP_NA_1
  - <3> := Double-point information M_DP_NA_1
  - <9> := Measured value, normalized value M_ME_NA_1
  - <11> := Measured value, scaled value M_ME_NB_1
  - <13> := Measured value, short floating point value M_ME_NC_1
  - <15> := Integrated totals M_IT_NA_1
  - <21> := Measured value, normalized value without quality descriptor M_ME_ND_1
  - <30> := Single-point information with time tag CP56Time2a M_SP_TB_1
  - <31> := Double-point information with time tag CP56Time2a M_DP_TB_1
  - <34> := Measured value, normalized value with time tag CP56Time2a M_ME_TD_1
  - <35> := Measured value, scaled value with time tag CP56Time2a M_ME_TE_1
  - <36> := Measured value, short floating point value with time tag CP56Time2a M_ME_TF_1
  - <37> := Integrated totals with time tag CP56Time2a M_IT_TB_1
  - <45> := Single command C_SC_NA_1
  - <46> := Double command C_DC_NA_1
  - <48> := Set point command, normalized value C_SE_NA_1
  - <49> := Set point command, scaled value C_SE_NB_1
  - <50> := Set point command, short floating point value C_SE_NC_1
  - <70> := End of initialization M_EI_NA_1
  - <100>:= Interrogation command C_IC_NA_1
  - <101>:= Counter interrogation command C_CI_NA_1
  - <103>:= Clock synchronization command (option see 7.6) C_CS_NA_1
- **Transmission of integrated totals :**
  - Mode C: Freeze and transmit by counter-interrogation commands
  - Counter read
  - Counter freeze without reset
  - General request counter

Definition of time outs

Parameter	Default value	Remarks	Selected value
t_0	30 s	Time-out of connection establishment	
t_1	15 s	Time-out of send or test APDUs	
t_2	10 s	Time-out for acknowledges in case of no data messages $t_2 < t_1$	
t_3	20 s	Time-out for sending test frames in case of a long idle state	

Maximum range of values for all time-outs: 1 to 255 s, accuracy 1 s.

Maximum number of outstanding I format APDUs *k* and latest acknowledge APDUs (*w*)

Parameter	Default value	Remarks	Selected value
k	12 APDUs	Maximum difference receive sequence number to send state variable	
w	8 APDUs	Latest acknowledge after receiving *w* I format APDUs	

Maximum range of values *k*: 1 to 32767 (2^{15}–1) APDUs, accuracy 1 APDU

Maximum range of values *w*: 1 to 32767 APDUs, accuracy 1 APDU (Recommendation: *w* should not exceed two-thirds of *k*).

Portnumber

Parameter	Value	Remarks
Portnumber	2404	In all cases