OSHMI

Open Substation HMI

OPC UA/DA Client Driver

Configuration Manual

Version 0.2

Introduction

This OPC UA/DA driver is based on Hayasoft’s OPC library stack from:

https://github.com/hylasoft-usa/h-opc

The h-opc library was forked to solve problems and customize as needed to:

https://github.com/riclolsen/h-opc

This driver requires the Microsoft .NET Framework 4.6 or later.

Using this driver it is possible to connect OSHMI to “n” OPC UA or DA servers over TCP/IP. Only the TCP/IP binary protocol is supported. Client certificates are now supported.

This driver is not a gateway, it only directs data to OSHMI.

Configuration

OSHMI must be configured to use the JSON UDP protocol driver interface on port 9100 (this is already configured by default installation, see OSHMI Configuration Manual).

The driver config file is c:\oshmi\conf\opc_client.conf. This file has the following format:

```
# OPC_UA_URL                    ,READ_INTERVAL_IN_SECONDS ,APP_NAME ,APP_CERTIFICATE_FILENAME ,CERTIFICATE_PASSWORD
opc.tcp://opcuaserver.com:48010 ,30                       ,OSHMI    ,certfile.pfx             ,OSHMI

#OPC_TAG_PATH                                  ,TYPE          ,SUBSCRIBE ,OSHMI_TAG            ,OSHMI_COMMAND_TAG
ns=2;s=Demo.Dynamic.Scalar.Double              ,Double        ,N         ,Demo.Dynamic.Scalar.Double        ,
ns=2;s=Demo.Dynamic.Scalar.Int16               ,Int16         ,N         ,Demo.Dynamic.Scalar.Int16         ,
ns=2;s=Demo.Dynamic.Scalar.Boolean             ,Boolean       ,N         ,Demo.Dynamic.Scalar.Boolean       ,
ns=2;s=Demo.Dynamic.Scalar.Byte                ,Byte          ,N         ,Demo.Dynamic.Scalar.Byte          ,
ns=2;s=Demo.Dynamic.Scalar.SByte               ,SByte         ,N         ,Demo.Dynamic.Scalar.SByte         ,
ns=3;s=AirConditioner_1.Temperature            ,Double        ,Y         ,AirConditioner_1.Temperature      ,

opc.tcp://opcuaserver.com:48484, 10
ns=1;s=Countries.US.Queens.Latitude            ,Double        ,Y         ,US.Queens.Latitude                ,
ns=1;s=Countries.US.Queens.Longitude           ,Double        ,N         ,US.Queens.Longitude               ,
ns=1;s=Countries.US.Queens.Temperature         ,Double        ,N         ,US.Queens.Temperature             ,
ns=1;s=Countries.US.Queens.Pressure            ,Double        ,N         ,US.Queens.Pressure                ,
ns=1;s=Countries.US.Queens.WindBearing         ,Double        ,N         ,US.Queens.WindBearing             ,
ns=1;s=Countries.US.Queens.WindSpeed           ,Double        ,N         ,US.Queens.WindSpeed               ,
ns=1;s=Countries.US.Queens.ApparentTemperature ,Double        ,N         ,US.Queens.ApparentTemperature     ,

opc.tcp://opcua.demo-this.com:51210/UA/SampleServer, 15
ns=2;i=10851                                   ,Int64         ,N         ,Data.Dynamic.Scalar.Int64Value    ,
ns=2;i=10856                                   ,DateTime      ,N         ,Data.Dynamic.Scalar.DateTimeValue ,
ns=2;i=10844                                   ,Boolean       ,Y         ,Data.Dynamic.Scalar.BooleanValue  ,WRITETAG
...
```

Fields are comma separated.

To comment a line add a “#” symbol to its beginning.

Empty lines are allowed.

Server parameters

OPC_URL: the standard OPC URL for the server. OPC UA URLs are like "opc.tcp://..." and OPC DA are like "opcda://...". User and password can be specified like in "opc.tcp://user:password@servername.com:port/PATH". This parameter is mandatory.

READ_INTERVAL_IN_SECONDS: this is the time interval in seconds to requests reading from all tags. This parameter is mandatory.

APP_NAME: application name. It is recommended as it is specified as the same application name that was used to create the certificate. This parameter is optional, if not specified the "OSHMI" name is assumed.

APP_CERTIFICATE_FILENAME: path and name to the certificate file. This parameter is optional, if not specified the connection is only possible without encryption. Only the .pfx file format is known to work.

CERTIFICATE_PASSWORD: password to use the certificate. This parameter is optional, if not specified an empty password is assumed.

Use the batch file mentioned below as example on how to create a certificate. The used certificate must be trusted by the server, this may require manual intervention in the server.

C:\oshmi\Opc.Ua.CertificateGenerator\create_oshmi_cert.bat

Tag parameters

OPC TAG PATH: path for the OPC tag. For UA it must be included the namespace number as and type of addressing in "ns=2;s=Random1".

TYPE: standard OPC data type. This parameters is optional, leave empty to let the driver detect the tag types. Boolean values are converted to digital values in OSHMI, all numeric values are converted to OSHMI analog values (double precision: 8 bytes floating point).

Incorrect type specification will trigger exceptions in the code, slowing down the communications. Please pay attention to logs produced when established communications.

String/text types and arrays are not supported.

See tables below for supported types.

OPC UA type	***Aliases***	***OSHMI Type***	***Precision Bytes***
Boolean	Bool	Digital	1
Byte, SByte		Analog (double)	1
Decimal		Analog (double)	16→8
Float	Single	Analog (double)	4
Double		Analog (double)	8
Int16, UInt16		Analog (double)	2
Int32	Integer	Analog (double)	4

UInt32		Analog (double)	4
Int64,UInt64		Analog (double)	8
DateTime		Analog (double)	8
Time		Analog (double)	8

OPC DA type	***Aliases***	***OSHMI Type***	***Precision Bytes***
vt_bool	Bool	Digital	1
vt_i1	Byte	Analog (double)	1
vt_i2	Int16	Analog (double)	2
vt_i4	Int32, State, Integer	Analog (double)	4
vt_r4	System.Single	Analog (double)	4
vt_r8	Float, System.Double	Analog (double)	8
vt_date	Date, DateTime	Analog (double)	8

SUBSCRIBE: “Y” or “N” (default is “N”). Allows to monitor a tag by exception. Servers may have limits in the number of monitored items, exceeding that limit can trigger exceptions in the code.

OSHMI TAG: a tag from OSHMI that will store the OPC tag values.

OSHMI COMMAND TAG: a tag from OSHMI that will send control values to the OPC tag for writing. Do not repeat OSHMI command tags in the OPC configuration file.

Remarks

OSHMI tags must be present at *point_list.txt* or they will be created at run time when first received (at the end of the address space).

To write to OPC tags it must be configured a command tag in *point_list.txt*.

Multiple connections to the same server can be used to establish different read intervals for each group of tags.