The File OPC Server
PRINTED MANUAL

File OPC Server

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: 2/7/2019

Publisher

AGG Software

Production

©2015-2019 AGG Software
http://www.aggsoft.com

Table of Contents

1 Introduction

This program implements a simple OPC DA2 server that polls a file on a local or network drive, reads updated values from that file, and then posts them via the OPC interface.

This way, you can easily create an OPC server and use it in a system that doesn’t support the OPC functionality, if it is inexpedient to add that functionality. You can create and update the input file by using scripts, which allows you to implement support for the OPC interface in a programming language that lacks that capability.

You can keep the input file on a network drive to build a distributed network without having to tackle complicated DCOM settings.

Key features:

- You can run multiple server instances on one computer and collect data from multiple files. Each server instance will have a unique ID.
- The program supports different data types for OPC tags.
- You can assign an OPC tag name for each value.
- You can assign tag names for an OPC server as a tree, which allows you to group variables.
- The program can read arrays of values from a file and create an OPC tag value of the "array" type.
- The program modifies OPC tag quality values. An invalid data flag will be set for an OPC tag if the file couldn’t be found, or the program couldn’t read the data, or the given value is missing.
- After reading a value, the program can scale it by multiplying it by the specified scaling factor.

2 License, Registration and technical support

2.1 License

SOFTWARE LICENSE

Trial Limited Version

The trial limited version of this software may be used for evaluation purposes at the user's own risk for a trial period. At the end of the trial period, the user must either purchase a license to continue using the software, or remove it from his/her system.

The trial limited version may be freely distributed, provided the distribution package is not modified. No person or company may charge a fee for the distribution of File OPC Server without written permission from the copyright holder.

Licensed Version

On payment of the appropriate license fee, the user is granted a non-exclusive license to use File OPC Server on one computer (i.e. a single CPU), for any legal purpose, at a time. The registered software may not be rented or leased, but may be permanently transferred, if the person receiving it agrees to terms of this license. If the software is an update, the transfer must include the update and all previous versions.

Registered customer are entitled to free updates during one year from the date of purchase. It means that during one year you can download and install the latest registered versions of File OPC Server from our site. If you don't want to purchase an updates, you can use the program forever; it will never expire, but you won't be able to use the latest version. If you purchased the software more than one year ago, you are no longer entitled to free upgrade and technical support; however, you can purchase an updates to the latest version at a special, greatly discounted price, and this updates will allow you to have free updates and technical support for another year. The type of update license must match the type of your existing license.

Whilst every care has been taken in the construction and testing of this software, it is supplied subject to the condition that the user undertakes to evaluate the suitability of the control for his/her purposes. AGG Software makes no representation of the software's suitability for any purpose, and the user agrees that AGG Software has no responsibility for any loss or damage occasioned by the use of this software.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" AND AGG SOFTWARE DISCLAIMS ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, CONFORMANCE WITH DESCRIPTION, TITLE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL AGG SOFTWARE BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL OR EXEMPLARY DAMAGES OR LOST PROFITS WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE PRODUCT, EVEN IF AGG SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY CASE, AGG SOFTWARE'S CUMULATIVE AND ENTIRE LIABILITY TO YOU OR ANY OTHER PARTY FOR ANY LOSS OR DAMAGES RESULTING FROM ANY CLAIMS, DEMANDS OR ACTIONS ARISING OUT OF OR RELATING TO THIS AGREEMENT SHALL NOT EXCEED THE PURCHASE PRICE PAID FOR THIS LICENSE.

Should any term of these terms and conditions be declared void or unenforceable by any court of competent jurisdiction, such declaration shall have no effect on the remaining terms hereof.

If you do not agree to these conditions you should not install this software.

2.2 Limitations

The unlicensed program works in the trial mode. The program allows to test all features but it limits the time and/or amount of processed data. The license key removes all limitations from the trial version. You may purchase a license key here.

The trial version of our software has the following limitations:

- The trial period is limited to 21 days. After that time the program stops working.
- The continuous program work time is limited to two hours. After that period the program shows a message and stops working;
- All data export modules can handle the first 100 records only;
- In the spy mode you can receive the first 65535 bytes only.

2.3 How to acquire a license

The unlicensed program works in the trial mode. The license key removes all limitations from the trial version and allows you to use our technical support for one year.

If you want to buy a program through the Internet visit the order page of our site. On this page you can get the newest information about the registration process, and also find an order link. After you've have the form of order registration. Enter your personal information and choose the most convenient payment method for you. Further, you'll get notification and follow the notes in it.

More information about our policies, payment terms and payment methods, frequently asked questions you may find on our web site.

2.4 Support

Technical questions	support@aggsoft.com
Common questions	info@aggsoft.com
Sales questions	sales@aggsoft.com

3 Installation

3.1 System requirements

Windows 2000 Professional - Windows 10, including x64 and x86 OS, Workstation and Server OS.

It is necessary to have at least one free COM port, not busy by any device (mouse, for example) to connect external device.

3.2 Installation process

If any beta-version was installed on your computer, remove it.

Quit of the working File OPC Server on installation time.

Run an installation file.

By default, the installation wizard installs File OPC Server to "C:\Programs Files\File OPC Server" or "C:\Programs Files (x86)\File OPC Server" the directory of your system disk, but you can change this path.

In the standard distributive of File OPC Server are no additional modules files, which you can download from our site.

4 Setup

4.1 Configuration file

The fileopcserver.xml file contains server settings. Put it in the folder containing the fileopcserver.exe file. If you did any changes to the configuration file, relaunch the program.

Example:

```
<Config>
 <Log file="fileopcserver.log" info="1" warning="1" error="1" limit="1000000" />
 <File name="data\ac-net7.raw" readrate="1000" />
  <Tags>
   <Tag name="TA7.COMMON.Time" datatype="time_t" offset="0" />
   <Tag name="TA7.ABS_VIBRO_EXT.01-V.Array16" datatype="short" offset="36;40;44"
arraysize="3" />
  </Tags>
</Config>
```

4.2 Log

The "Log" XML node contains the configuration of the program’s log file. The program can output all messages of a selected type to the specified log file for debugging, security, or process monitoring purposes.

Example:

```
<Log file="opchttpgate.log" info="1" warning="1" error="1" limit="1000000" />
```

file (optional; default: empty value) - The path and name of the log file. If this value is empty, the server will not write anything to the log file.

info (optional; default: 1) - If "1" (true), the program will write all informational messages to the log file.

warning (optional; default: 1) - Same as above for warning messages.

error (optional; default: 1) - Same as above for error messages.

limit (optional; default: 100000) - The log file size limit in bytes.

4.3 Main window

The "DesktopModeSettings" XML node contains the configuration of the main window when the program running in the desktop mode.

Example:

```
<DesktopModeSettings>
  <CustomTitle addversion="1">OPC-HTTP Gateway</CustomTitle>
  <CustomLabel>Uptime</CustomLabel>
  <ExitPassword enabled="1">admin</ExitPassword>
  <StopPassword enabled="1">admin</StopPassword>
</DesktopModeSettings>
```

CustomTitle (optional; default: empty) - The program will place this text in the window caption. If the "addversion" attribute is "1" then the program will append a version number to the custom title.

CustomLabel (optional; default: empty) - If this value is not empty the program will place this text above the uptime label.

ExitPassword (optional; default: empty) - If this value is not empty and the "enabled" attribute is "1" the program will ask for a password when a user close the program. If the "blockuserlogout" attribute is "1" the program will try to block computer reboot or user log out.

StopPassword (optional; default: empty) - If this value is not empty and the "enabled" attribute is "1" the program will ask for a password when a user click the "Stop" button.

4.4 Input file

The "File" node in the configuration XML file contains the description of the input file to be monitored by the program. By default, the program operates with binary data files. The program can also read data from a text file and convert the values to the specified data type.

Example:

```
<File name="data.raw" readrate="1000" />
```

name (mandatory parameter; default value: none) - The input file path and name. You can use a network path like "\\192.168.1.1\Public\Data\data.bin".

readrate (optional parameter; default value: 1000) - The file polling interval in milliseconds. If the file has changed more than once during the specified interval, any intermediate changes will be ignored.

writeretry (optional parameter; default value: 1) - The program may try to write to a data file several times. This is useful if another program writes to the same file too.

OPC tags

This node contains a list of OPC tags, their names, the location of each value in the input file, and additional attributes.

Example:

```
  <Tags>
   <Tag name="TA7.COMMON.Time" datatype="time_t" offset="0" />
  </Tags>
```

name (mandatory parameter; default value: none) - The OPC tag name. It can contain any printed characters. The dot character is used as a path delimiter. For example, the "TA7.COMMON.Time" tag name sets the following data tree hierarchy:

```
TA7
|--- COMMON
       |---- Time
```

datatype (mandatory parameter; default value: short) - The OPC tag data type.

Data type	Size (byte)	Description
time_t	4	Time stamp in the C++ format (in the UTC+0 time zone). It contains the number of seconds since January 1, 1970.
datetime	8	Time stamp in the OleTime format (ref. MSDN OleTimeToSystemTime)
short	2	Signed integer (from -32,768 to 32,767)
ushort	2	Unsigned integer (from 0 to 65,535)
float	4	IEEE 754 floating point number
double	8	IEEE double precision floating point number, 11-bit exponent, and a 52-bit mantissa. (2.23*10^308 to 1.79*10^308)
char	1	ASCII character
byte	1	Unsigned integer (from 0 to 255)
shortint	1	Signed integer (from -128 to 127)
ulong	4	Unsigned integer (from 0 to 4294967295)
integer	4	Signed integer (from -2147483648 to 2147483647)

int64	8	Signed 64-bit (from -2^63 to 2^63-1)
ansi	variable length	ASCII string. The maximum string size in bytes should be defined in the **arraysize** attribute. The string may end with the null byte.
unicode	variable length	Unicode string. The maximum string size in bytes should be defined in the **arraysize** attribute. The string may end with two null bytes.
utf8	variable length	UTF-8 string. The maximum string size in bytes should be defined in the **arraysize** attribute. The string may end with two null bytes.
watchdog	4	This is the special data type. The values is identical with "time_t", but the program handles it internally. See the "timeout" attribute below.
boolean	1	Logical "true" (in binary data it is any non-zero value) or "false" (a zero value)

offset (mandatory parameter; default value: 0) - The offset of the value in the input file. Offsets are counted from zero. A zero value points to the first byte in the file.

Depending on the data type, the program will read one or more bytes starting from the specified position.

The program will work faster if you specify tags in the configuration in succession, that is, from lesser offsets to greater ones.

Make sure that you use the little-endian format in the file, that is, the offset should point to the least significant byte.

If the offset equal "-1" the program makes a virtual OPC tag. This OPC tag is not linked to a file and resides in the memory only. Several OPC clients may exchange data using these tags. If all tags in the configuration are virtual, the program does not read a data file at all.

arraysize (optional parameter; default value: none) - The number of elements in the array. This attribute can be from 2 to 513. If you do not specify it, the OPC tag will contain an array of values.

For a data array, you can specify multiple offsets using the semicolon as a delimiter.

```
<Tag name="TA7.ABS.Array16" datatype="short" offset="0;12;16;20" arraysize="6" />
```

You don’t have to put array elements in sequential order in the input file.

If you specify only one offset, the program will consider all array elements to be in sequential order, starting from the specified offset.

If the number of offsets specified is less than the number of elements in the array, the program will consider the remaining array elements to be in sequential order, starting from the last specified offset.

scale (optional parameter; default value: 1) - The scaling factor. If this attribute is specified and is not equal to 1, the program will multiply each read value by the scaling factor before posting data via OPC.

timeout (optional parameter; default value: 300) - This attribute is used with conjunction with the "watchdog" data type. It defines a watchdog interval in seconds. If a difference between the system time and a read value is more than this interval then the program will set this tag to 2 (stale data). Otherwise the program will set the tag value to 1.

write (optional parameter) - This attribute enables or disables writing to an OPC tag. If the OPC tag is not virtual, the program writes values to the data file. By default, writing is enabled for virtual tags and disable for all other tags. Please note, the program cannot write arrays and strings to a file.

default (optional parameter; default value not defined) - This attribute specifies a default value for an OPC tag. We recommend using this value for virtual OPC tags.

Example:

```
<Tag name="TA7.ABS_VIBRO_RMS.10-O-Vs" datatype="short" offset="964" scale="0.1" />
```

Note that if the scaling factor is a floating-point number, the OPC tag data type will be a floating-point number too, regardless of the input data type.
For an array of values, the scaling factor will be applied to each array element.

4.5 A text file with input data

The program can also read data from a text file and convert the values to the specified data type. Unlike binary files, text files take more disk space but are easy to create and edit. You can use any text editor for doing that.

The program can handle text files of the following types:

An CSV file containing one line in the CSV format. You can use the TAB character, the pipe character (“|”), or the semicolon as a field separator.
A plain-text file. Use a separate line for each value.

An example of a CSV file

```
2017-09-07 00:00:00|1|2|3|99.01|"Test string\\\x03"
```

An example of a simple text file

```
2017-09-07 00:00:00
1
2
3
99.01
Test string\\\x03
```

File parameters

You can configure the text file’s parameters just like in case of a binary file. The **type** attribute, which specifies the file type, is added for text files.

tab – An CSV file; the field separator is the TAB character (its ASCII code is 0x09).
semicolon – An CSV file; the field separator is the semicolon.
pipe – An CSV file; the field separator is the pipe character (its ASCII code is 0x7C).
text – A simple text file.

Example:

```
<File name="data.csv" type="pipe" readrate="1000" />
```

OPC tags

You can configure the tags just like in case of a binary file.

When specifying tags, you must also specify the **offset** attribute, which in this case defines the column in the CSV file or the line number in the simple text file.

Note: The position is counted from zero. An offset attribute equal to zero means the first column in a CSV file or the first line in a simple text file.

The program expects the data in the CSV file or the simple text file to conform to the following format:

Date and time – Use the ISO format YYYY-MM-DD HH:NN:SS. Specify hours in the 24-hour format (for example, 2017-01-01 23:00:00).

Real numbers – Use the dot (period) as a decimal separator (for example, 99.01).

Strings – Any characters whose ASCII code is not within the 0x20–x7F range must be encoded in the form \x00. The backslash character (“\”) is reserved for encoding special characters; if you need to use it per se, write it twice (“\\”). You can find an example of an CSV file and an example of a simple text file above. Please note that each string in a CSV file must be enclosed in double quotes ("some_string").

Boolean - the boolean "True" value can be specified as "1", "Y", "TRUE" (without quotes). Any other value is "False".

5 OPC service tags

Each time the OPC server is launched, the program adds several OPC service tags that describe the server’s operation.

System.RAW.OK - If the last file reading operation was successful, this variable contains "true."
System.RAW.Name - The input file name.
System.RAW.Path - The input file path.
System.RAW.LastRead - The time stamp of the last file polling.
System.Local.Timestamp - The current time at the server.
System.Active.Clients - The number of connected OPC clients.

6 Service mode

The program can work in the Windows Service mode and start automatically with Windows. If the program is already running in the service mode, you cannot start other instances of it in the desktop mode.

Install/Uninstall

To install or uninstall the service, execute one of the following commands in the "cmd.exe" command prompt. If you have Windows Vista or higher, you need to run "cmd.exe" with elevated administrator privileges.

Install: opchttpgate.exe /AI
Uninstall: opchttpgate.exe /R

Start/Stop

To start or stop the service, open the "Services" control panel, find "OPC-HTTP Gateway Service" there, and execute the necessary command.

More commands

To get a list of possible commands, execute the following command in the "cmd.exe" command prompt:

```
opchttpgate.exe /?
```

7 Registering the OPC server in the system

```
fileopcserver.exe /REGSERVER
```

Registering or unregistering the OPC server in Windows Vista or higher requires elevated administrator rights.

Unregistering the OPC server in the system

```
fileopcserver.exe /UNREGSERVER
```

8 Running multiple server instances on one computer

You may need to run multiple instances of the program if you want to post data from multiple input files via OPC.

To run multiple instances of the program, do the following:

1. Copy the "fileopcserver.exe" file to the "fileopcserver1.exe" file; you can use any other unused index number in the new filename (for example, "fileopcserver10.exe").
2. Copy the "fileopcserver.xml" configuration file to the "fileopcserver1.xml" file; make sure that the index is the same as in the executable file’s name.
3. Register the service for the new file as described in the "Service mode - Installation" section.
4. Register the OPC server for the new file in the system as described in the "Registering the OPC server in the system" section.

9 OPC UA

The separate "fileopcserverua.exe" file contains an OPC UA server that you may use at the same time with the OPC DA server. This server support the binary protocol only over TCP/IP.

The setup process of this server is identical with the OPC DA server. The configuration file (fileopcserverua.xml) includes the same settings and the "OpcUa" additional node.

```
<Config>
....
<OpcUa port="10001" login="root" password="root" discovery="opc.tcp://127.0.0.1:4840" encry
</Config>
```

port (optional; default: 10002) – the TCP port of the OPC UA server (add this port to exceptions in your firewall).

login, password (optional; default: empty value) – if you have specified these parameters then any OPC UA client must provide these credentials while connecting to the server. The password may be empty.

discovery (optional; default: empty value) – you may specify the URL address of your LDS (local discovery server) in your local network and our OPC UA will try to register on this LDS. It allows OPC UA clients to find our OPC UA server easily.

Notes:

If this attribute is defined the server will periodically try to connect to LDS and register.

Our server does not use encryption while connecting to LDS. Therefore your LDS server must allow unsecured connections.

The latest version of LDS from OPC Foundation (https://github.com/OPCFoundation/UA-LDS) requires a secure connection in any case. Therefore our server cannot connect to it yet.

encryption (optional; default: 0) – if this attribute is "1" our OPC server will create additional secure endpoints.

In this case you must place a certificate "server_cert.der" file and a private key "server_key.der" file to "program path\CertificateStore\private". If the private key does not exist the server will force disable encryption.

Also you must place public keys of allowed OPC UA clients in the "DER" format to "program path\CertificateStore\trusted".

You may create all necessary key files using openssl. Example commands for Windows is below.

```
openssl genrsa -out ca.key {}

openssl req -x509 -new -nodes -key ca.key -days 3650 -subj "/C=US/O=AGG Software
(www.aggsoft.com)/CN=AGG Software/" -out ca.crt

openssl req -new -newkey rsa:{} -nodes -subj "/C=US/O=AGG Software
(www.aggsoft.com)/CN=FileOpcServer" -keyout localhost.key -out localhost.csr

openssl x509 -req -days 3650 -in localhost.csr -CA ca.crt -CAkey ca.key -
CAcreateserial -out localhost.crt -extfile {} -extensions v3_ca

openssl x509 -in localhost.crt -outform der -out server_cert.der

openssl rsa -inform PEM -in localhost.key -outform DER -out server_key.der
```