The OPC logger ActiveX
PRINTED MANUAL

OPC logger ActiveX

© 1999-2008 AGG Software

Printed: 09.09.2008

Table of Contents

1   Introduction

1.1   About

OPC logger ActiveX is an ActiveX component that uses Advanced OPC Data Logger to get either data from an OPC server or already processed and converted data besides the standard export tools in Advanced OPC Data Logger.

OPC logger ActiveX key features:

It is a 32-bit ActiveX component for Visual Basic and other ActiveX-compatible development tools (Visual C++, Delphi, Microsoft Office and a lot more). The .NET development environment is fully supported.

It is perfectly optimized for fast performance and minimum RAM usage. This component can quite easily receive and send data from several data sources and Advanced OPC Data Logger configurations. At the same time, it ensures minimum RAM usage.

It allows you to use Advanced OPC Data Logger data parsing modules without previously processing data and get already prepared, formatted and converted into the necessary data type values for future use.

Several convenient functions and procedures that do not require special knowledge allow you to get the data you would like to have within an hour. Due to working together with Advanced OPC Data Logger and modules for Advanced OPC Data Logger, this component is a powerful tool for getting and processing data from an OPC server.

It is possible to work with several configurations simultaneously.

It is possible to get notifications about the internal program events.

It is possible to emulate sending and receiving data.

Company site: http://www.aggsoft.com/
Program page: http://www.aggsoft.com/opc-logger-activex.htm.

1.2   How to register

The program is distributed on shareware terms. This signifies limited or unavailable several features of the program, getting of full value or available after program registration.

If you'd like to be a registered user, to get information about the release of new versions, to use technical support and, at last, to get access to disabled functions of the program, register your copy. For registration, please, read license agreement 2.

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 services, registration documents, payment means you can get on our order page of our site.

Registered users are entitled to FREE upgrades for the major version, they purchased. That means, that in case you bought registration key for version 2.00, you will receive upgrades for versions 2.xx for FREE . If a new authorization code is required it will be issued upon request at no charge to users who have registered for the current major version.

1.3   Support

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

1.4   License

OPC logger ActiveX
Copyright ©1999-2008 AGG Software.
All Rights Reserved

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 OPC logger ActiveX 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 OPC logger ActiveX on one computer (i.e. a single CPU), with one instance of a host application, 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. The registered software may be used in the development of an end-user application, as described below. This license is for a single developer and not for an entire company. If additional programmers wish to use OPC logger ActiveX, additional copies must be licensed.

An 'end-user application' is a specific application program that is licensed to a person or firm for business or personal use. Furthermore, the end-user must not be in a position to be able to neither modify the program, nor to create OPC logger ActiveX based programs. Likewise, the end-user must not be given the OPC logger ActiveX serial number.

This license for a single developer may be used in the 'end-user application' program, if the developer does not use the OPC logger ActiveX component in the development.

OPC logger ActiveX contains an additional command line registration tools for installing OPC logger ActiveX on a PC. These files are not redistributable.

Registered customers 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 OPC logger ActiveX 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.
```

1.5   System requirements

One of the following operation systems are necessary to run the program:

- Windows 95 (needed to install the last comctrl.dll update, please, visit download section of our site);
- Windows 95 OSR2;
- Windows 98;
- Windows Me.

If you'll use our software on the following OS, then you need administrator privileges for registration and installation:

- Windows NT4 (SP4 and above).
- Windows 2000 Professional;
- Windows 2000 Server;
- Windows XP x86;
- Windows 2003 Server x86;
- Windows Vista (all editions) x86.

Please, note, that the x64 architecture isn't supported.

You'll also need installed Internet Explorer 5.0 or later to look help on the program.
It is necessary to install and configure the "Advanced OPC Data Logger" software.

1.6   Install

Exit the running instance of Advanced OPC Data Logger before the installation.

Run the installation file.

By default, "OPC logger ActiveX" will be installed to the "/Programs Files/Advanced OPC Data Logger" folder of your system disk, but you can specify a custom path.

During the installation, the installation wizard will ask your confirmation for registering "OPC logger ActiveX" in the system. If you refuse this procedure, you will have to run !register.bat located in the program folder.

2   Reference

This section of help provides reference information for the API elements provided by **OPC logger ActiveX**.

The **OPC logger ActiveX** reference information can be divided into following categories.
- Interfaces [5]

2.1   Interfaces

This section contains reference information for the COM interfaces provided by **OPC logger ActiveX**. The following interfaces are used with **OPC logger ActiveX**.

- ITLoggerControlAX [5]
- ITLoggerControlAXEvents [12]

2.1.1   ITLoggerControlAX

OPC logger ActiveX Object

All methods of the **ITLoggerControlAX** interface are listed in the following table

Method List		
method	RegisterDataNotify [6]	The function registers your application for events receiving. Without this registration the data logger will not send you any events. If you don't need to use events, then don't call this function and you save computer resources.
method	GetConfigurations [7]	The function returns a string with all configurations in the data logger software. The string contains names from the drop-down box from the main window. Each configuration name is separated by CR+LF characters. Example: COM1+CRLF+COM2+CRLF
method	GetConfigID [7]	Retrieves a configuration ID for a specified configuration name. Later, you should use this ID in subsequent calls.
method	Send [8]	Send data through the data logger. This function allows you send data to an OPC server or simulate it.
method	StartPause [9]	Stops/starts data receiving for a specified data source, closes/opens an OPC server.
method	SetLicenseKey [9]	Allows you to specify the license key, which will remove all demo limitations.
method	SetOption [10]	Sets a specified property/option for the configuration.

		Some properties can be changes at run-time without saving.
method	GetOption [11]	Retrieves a specified property/option for the configuration. Some options should be decoded after reading.
method	SaveApplyOptions [11]	Allows you to save and apply all settings for the specified configuration. The program will reload and reinitialize all plug-ins.

Requirements

Header: Declared in opclogger.h
Import Library: Use opclogger.lib

2.1.1.1 ITLoggerControlAX::RegisterDataNotify

The function registers your application for events receiving. Without this registration the data logger will not send you any events. If you don't need to use events, then don't call this function and you save computer resources.

[C++]

```
long   RegisterDataNotify(
    long ConfigID
    long NotifyType
);
```

[Visual Basic]

```
object.RegisterDataNotify(
    ConfigID   As Long
    NotifyType As Long
) As Long
```

Parameters
ConfigID

    [in] Configuration ID, which you should retrieve with help of the GetConfigID [7] function.

NotifyType

    [in] The event type identifier: 0 - RAW data events; 1 - data packet events; 2 - program events. If you want to utilize all events, then call this function several times.

Requirements

Header: Declared in opclogger.h
Import Library: Use opclogger.lib

See Also
ITLoggerControlAX interface [5]

2.1.1.2 ITLoggerControlAX::GetConfigurations

The function returns a string with all configurations in the data logger software. The string contains names from the drop-down box from the main window. Each configuration name is separated by CR+LF characters. Example: COM1+CRLF+COM2+CRLF

[C++]

```
BSTR   GetConfigurations();
```

[Visual Basic]

```
object.GetConfigurations() As String
```

Requirements

Header: Declared in opclogger.h
Import Library: Use opclogger.lib

See Also
ITLoggerControlAX interface 5

2.1.1.3 ITLoggerControlAX::GetConfigID

Retrieves a configuration ID for a specified configuration name. Later, you should use this ID in subsequent calls.

[C++]

```
long   GetConfigID(
    BSTR ConfigName
);
```

[Visual Basic]

```
object.GetConfigID(
    ConfigName As String
) As Long
```

Parameters

ConfigName

[in] The configuration name, which you can receive with help of the GetConfigurations 7 function.

Requirements

Header: Declared in opclogger.h
Import Library: Use opclogger.lib

2.1.1.4 ITLoggerControlAX::Send

Send data through the data logger. This function allows you send data to an OPC server or simulate it.

[C++]

```
long   Send(
    long ConfigID
    long SendType
    long DataType
    BSTR Data
);
```

[Visual Basic]

```
object.Send(
    ConfigID As Long
    SendType As Long
    DataType As Long
    Data      As String
) As Long
```

Parameters

ConfigID

[in] Configuration ID, which you should retrieve from the GetConfigID 7 .

SendType

[in] The type of the sending operation: 0 - simulation, receiving; 1 - simulation, sending; 2 - send data to an OPC server.

DataType

[in] The type of data: 0 - binary data; 1 - text string, that will be parsed and all patterns like #AA will be replaced with a character with a corresponding hexadecimal code.

Data

[in] Data to send.

Requirements

Header: Declared in opclogger.h
Import Library: Use opclogger.lib

2.1.1.5   ITLoggerControlAX::StartPause

Stops/starts data receiving for a specified data source, closes/opens an OPC server.

[C++]

```
long    StartPause(
    long ConfigID
    long State
);
```

[Visual Basic]

```
object.StartPause(
    ConfigID As Long
    State    As Long
) As Long
```

Parameters

ConfigID

[in] Configuration ID, which you should retrieve from the GetConfigID⌐ 7⌐.

State

[in] If the value greater than zero, then port will be opened (if the port isn't opened before). If State = 0, then the port will be closed.

Requirements

Header: Declared in opclogger.h
Import Library: Use opclogger.lib

See Also
ITLoggerControlAX interface⌐ 5⌐

2.1.1.6   ITLoggerControlAX::SetLicenseKey

Allows you to specify the license key, which will remove all demo limitations.

[C++]

```
long    SetLicenseKey(
    BSTR Key
);
```

[Visual Basic]

```
object.SetLicenseKey(
    Key As String
) As Long
```

Parameters

Key

    [in] Your license key like: AAAA-BBBB-CCCC-DDDD

Requirements

  Header: Declared in opclogger.h
  Import Library: Use opclogger.lib

See Also
ITLoggerControlAX interface ⌐5⌐

2.1.1.7 ITLoggerControlAX::SetOption

Sets a specified property/option for the configuration. Some properties can be changes at run-time without saving.

[C++]

```
long    SetOption(
    long     ConfigID
    BSTR     Option
    VARIANT* Value
);
```

[Visual Basic]

```
object.SetOption(
    ConfigID As Long
    Option   As String
    Value    As Variant
) As Long
```

Parameters

ConfigID

    [in] Configuration ID, which you should retrieve from the GetConfigID ⌐7⌐.

Option

    [in] The option name. This name you can find the Window registry (see the section below SaveApplyOptions) ⌐11⌐

Value

    [in] The option value like True or False for the boolean data type.

Requirements

  Header: Declared in opclogger.h
  Import Library: Use opclogger.lib

See Also
ITLoggerControlAX interface ⌐5⌐

2.1.1.8   ITLoggerControlAX::GetOption

Retrieves a specified property/option for the configuration. Some options should be decoded after reading.

[C++]

```
long   GetOption(
    long     ConfigID
    BSTR     Option
    VARIANT** Value
);
```

[Visual Basic]

```
object.GetOption(
    ConfigID As Long
    Option   As String
    Value    As Variant
) As Long
```

Parameters

ConfigID

[in] Configuration ID, which you should retrieve from the GetConfigID ⌐7⌐.

Option

[in] The option name. This name you can find the Window registry (see the section below SaveApplyOptions) ⌐11⌐

Value

[out] The option value like True or False for the boolean data type.

Requirements

Header: Declared in opclogger.h
Import Library: Use opclogger.lib

See Also
ITLoggerControlAX interface ⌐5⌐

2.1.1.9   ITLoggerControlAX::SaveApplyOptions

Allows you to save and apply all settings for the specified configuration. The program will reload and reinitialize all plug-ins. The program will store all settings in the following registry node:

HKEY_LOCAL_MACHINE\SOFTWARE\AGG Software\Advanced OPC Data Logger\

{8BB712F0-490F-499F-B616-3E809FAE4A22}. If this

registry node isn't available for writing for the current user, then the program will use

HKEY_CURRENT_USER\SOFTWARE\AGG Software\Advanced OPC Data Logger\
{8BB712F0-490F-499F-B616-3E809FAE4A22}

If you are using an INI file (you specified this mode in the program), then the program will store the settings in the corresponding INI file instead.

Please, note that the "{8BB712F0-490F-499F-B616-3E809FAE4A22}" identifier, may be different on your PC.

[C++]

```
long   SaveApplyOptions(
    long ConfigID
    long DoSave
);
```

[Visual Basic]

```
object.SaveApplyOptions(
    ConfigID As Long
    DoSave   As Long
) As Long
```

Parameters

ConfigID
　　[in] Configuration ID, which you should retrieve from the GetConfigID 7 .
DoSave
　　[in] If this parameter is > 0, then the program will apply and save all settings, otherwise the program will apply the settings only.

Requirements

Header: Declared in opclogger.h
Import Library: Use opclogger.lib

See Also
ITLoggerControlAX interface 5

2.1.2 ITLoggerControlAXEvents

Events interface for OPC logger ActiveX.

All methods of the **ITLoggerControlAXEvents** interface are listed in the following table

Method List

method	OnData [13]	Container receives this event when a new RAW data is received by the data logger software.
method	OnDataPacket [14]	Container receives this event when a new data packet is received and parsed by the data logger software.
method	OnEvent [14]	Container receives this event when a new internal event of the data logger software is fired.

Requirements

Header: Declared in opclogger.h
Import Library: Use opclogger.lib

2.1.2.1 ITLoggerControlAXEvents::OnData

Container receives this event when a new RAW data is received by the data logger software.

[C++]

```
HRESULT   OnData(
    BYTE DataSent
    BSTR Data
);
```

[Visual Basic]

```
object.OnData(
    DataSent As Byte
    Data     As String
)
```

Parameters

DataSent
   [in] If DataSent>0 then the Data is sent, otherwise the Data is received.
Data
   [in] Data buffer.

Requirements

Header: Declared in opclogger.h
Import Library: Use opclogger.lib

See Also
ITLoggerControlAXEvents interface [12]

2.1.2.2 ITLoggerControlAXEvents::OnDataPacket

Container receives this event when a new data packet is received and parser by the data logger software.

[C++]

```
HRESULT   OnDataPacket(
    BYTE    DataSent
    VARIANT* VarNames
    VARIANT* VarValues
);
```

[Visual Basic]

```
object.OnDataPacket(
    DataSent  As Byte
    VarNames  As Variant
    VarValues As Variant
)
```

Parameters

DataSent

[in] If DataSent>0 then the Data is sent, otherwise the Data is received.

VarNames

[in] The one dimensional array, which contains parsed variables name. These names are defined in a parser plug-in in the data logger software.

VarValues

[in] The one dimensional array, which contains parsed variables value. The number of elements in both arrays are identical.

Requirements

Header: Declared in opclogger.h
Import Library: Use opclogger.lib

See Also
ITLoggerControlAXEvents interface 12

2.1.2.3 ITLoggerControlAXEvents::OnEvent

Container receives this event when a new internal event of the data logger software is fired.

[C++]

```
HRESULT   OnEvent(
    long     Event
    BSTR     EventName
```

```
    VARIANT* ArgNames
    VARIANT* ArgValues
);
```

[Visual Basic]

```
object.OnEvent(
    Event     As Long
    EventName As String
    ArgNames  As Variant
    ArgValues As Variant
)
```

Parameters

Event

   [in] The internal event ID.

EventName

   [in] The event name like PORT-OPEN.

ArgNames

   [in] One dimensional array, which contains names of additional attributes, which are used with this event.

ArgValues

   [in] One dimensional array, which contains values of additional attributes, which are used with this event. The number of elements in both arrays are identical.

Requirements

 Header: Declared in opclogger.h
 Import Library: Use opclogger.lib

See Also
ITLoggerControlAXEvents interface 12