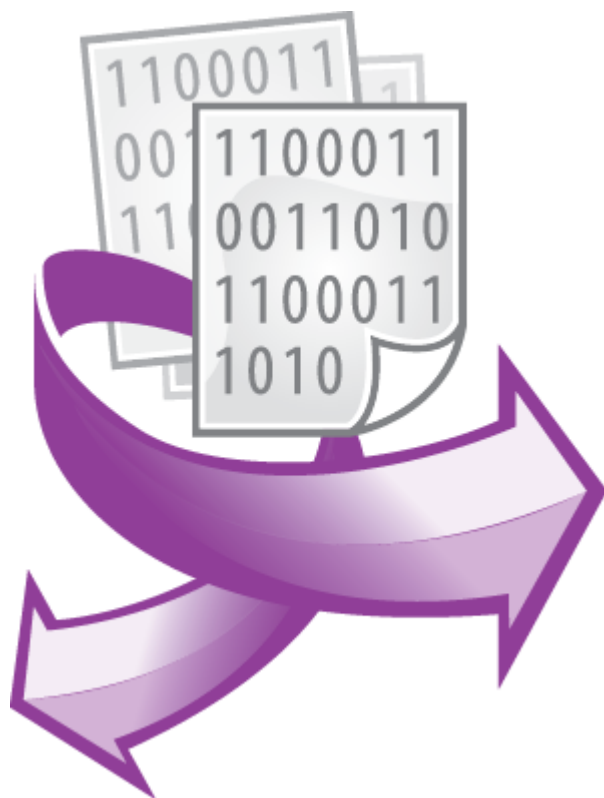




**The RS232 logger ActiveX
PRINTED MANUAL**

RS232 logger ActiveX

© 1999-2025 AGG Software

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

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: 28.03.2025

Publisher

AGG Software

Production

© 1999-2025 AGG Software

<http://www.aggsoft.com>

Table of Contents

Part 1 Введение

1

1 О компоненте

1

2 Регистрация

2

3 Техническая поддержка

2

4 Лицензионное соглашение

2

5 Системные требования

3

6 Процесс установки

4

Part 2 Справочная информация

4

1 Интерфейсы

4

ITLoggerControlAX

5

ITLoggerControlAX::RegisterDataNotify

6

ITLoggerControlAX::GetConfigurations

6

ITLoggerControlAX::GetConfigID

7

ITLoggerControlAX::Send

8

ITLoggerControlAX::StartPause

9

ITLoggerControlAX::SetLicenseKey

9

ITLoggerControlAX::SetOption

10

ITLoggerControlAX::GetOption

11

ITLoggerControlAX::SaveApplyOptions

12

ITLoggerControlAXEvents

13

ITLoggerControlAXEvents::OnData

13

ITLoggerControlAXEvents::OnDataPacket

14

ITLoggerControlAXEvents::OnEvent

15

1 Введение

1.1 О компоненте

RS232 logger ActiveX это ActiveX component, который использует программу Advanced Serial Data Logger для получения данных из RS232 или COM порта или уже обработанных и преобразованных данных, помимо стандартных средств экспорта Advanced Serial Data Logger.

Ключевые особенности RS232 logger ActiveX:

32-bit ActiveX компонент для Visual Basic и других ActiveX совместимых средств разработки (Visual C++, Delphi, Microsoft Office и множество других). Полностью поддерживается среда разработки .NET.

Улучшенные средства адаптации - поддержка ActiveX Automation обеспечивает тесную связь с другими приложениями Windows и позволяют вести адаптацию привычными инструментами типа Visual Basic.

Отлично оптимизирован для быстрой работы и минимального потребления памяти. Данный компонент может довольно просто принимать и пересылать данные от нескольких источников данных и конфигураций Advanced Serial Data Logger, при этом гарантируя минимальное потребление оперативной памяти.

Позволяет использовать модули парсинга данных Advanced Serial Data Logger для предварительной обработки данных и получать для последующего использования уже подготовленные, отформатированные и преобразованные в надлежащий тип данных значения.

Несколько удобных функций и процедур, не требующих особенных знаний, позволяют вам в течении часа получить данные, которые вы хотели бы иметь. Благодаря связке с Advanced Serial Data Logger и модулями для Advanced Serial Data Logger данный компонент представляет собой мощный инструмент для получения и обработки данных из RS232 или COM порта.

Возможность работы с несколькими конфигурациями одновременно.

Возможность принимать уведомления о внутренних событиях программы.

Возможность эмуляции отправки и приема данных.

Пример для Microsoft Excel с небольшим количеством кода Visual Basic.

Сайт компании: <https://www.aggsoft.ru/>

Страница программы: <https://www.aggsoft.ru/data-logger-plugins/activex.htm>

1.2 Регистрация

Программа распространяется на условиях shareware-программ. Это означает, что закрытые возможности программы становятся доступными только после ее регистрации.

Если вы хотите быть зарегистрированным пользователем программы, получать информацию о выпуске ее новых версий, использовать техническую поддержку и, наконец, получить доступ к заблокированным функциям программы, зарегистрируйте вашу версию программы. Условия регистрации указаны в [лицензионном соглашении](#)².

Чтобы купить программу через Internet, посетите [страницу регистрации](#) нашего сайта. На этой странице мы сможете найти последнюю информацию о процессе регистрации, а также найти ссылку на форму оформления заказа регистрации. В форме оформления заказа необходимо ввести Ваши подробные личные сведения и выбрать наиболее удобную для Вас форму оплаты. После чего вы получите уведомление с указаниями на дальнейшие действия. После чего вы получите письмо с регистрационными данными.

На [странице регистрации](#) нашего сайта вы можете получить подробную информацию об услугах, регистрационных документах, способах оплаты программы.

Зарегистрированные пользователи имеют право на БЕСПЛАТНЫЕ обновления в течение одного года после регистрации. После этого пользователи могут приобрести обновление со скидкой.

1.3 Техническая поддержка

Вопросы по RS232 logger ActiveX Общие вопросы Вопросы по покупке ПО	https://www.aggsoft.ru/support/
---	---

1.4 Лицензионное соглашение

Copyright © 1999-2025 AGG Software.
All Rights Reserved

Лицензия на использование и распространение

Эта лицензия действительна только для граждан стран бывшего СССР.

1. Все авторские права на "RS232 logger ActiveX" принадлежат только автору - Артему Герасимову.
2. Граждане стран бывшего СССР имеют право зарегистрировать данный продукт по льготной цене.

3. Это подразумевает оплату регистрации "RS232 logger ActiveX" в рублях, в пределах бывшего СССР.

4. Если условия данной лицензии не устраивают попадающих под ее действие лиц, они имеют право воспользоваться на общих основаниях лицензией, приведенной на нашем сайте в Интернете.

5. Для коммерческого использования "RS232 logger ActiveX" необходимо зарегистрировать. Уточнить процедуру регистрации можно по адресу sales@aggsoft.ru

6. "RS232 logger ActiveX" распространяется свободно, при условии того, что настоящий дистрибутив не изменен. Ни одно частное лицо или организация не может брать плату за распространение "RS232 logger ActiveX" без разрешения автора.

7. "RS232 logger ActiveX" ПОСТАВЛЯЕТСЯ ПО ПРИНЦИПУ "AS IS". НИКАКИХ ГАРАНТИЙ НЕ ПРИЛАГАЕТСЯ И НЕ ПРЕДУСМАТРИВАЕТСЯ. ВЫ ИСПОЛЬЗУЕТЕ ЭТО ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ НА СВОЙ РИСК. АВТОР НЕ БУДЕТ ОТВЕЧАТЬ НИ ЗА КАКИЕ ПОТЕРИ ИЛИ ИСКАЖЕНИЯ ДАННЫХ, ЛЮБУЮ УПУЩЕННУЮ ВЫГОДУ В ПРОЦЕССЕ ИСПОЛЬЗОВАНИЯ ИЛИ НЕПРАВИЛЬНОГО ИСПОЛЬЗОВАНИЯ ЭТОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.

8. Вы не можете использовать, копировать, эмулировать, создавать новые версии, сдавать в наем или аренду, продавать, изменять, декомпилировать, дизассемблировать, изучать код программы другими способами, передавать зарегистрированную программу или любые из ее составляющих, иначе, чем определено настоящим лицензионным соглашением. Любое такое нелегальное использование означает автоматическое и немедленное прекращение действия настоящего соглашения и может преследоваться по закону.

Все права, не предоставленные здесь явно, сохраняются за Артемом Герасимовым.

9. Установка и использование "RS232 logger ActiveX" означает, что вы понимаете положения настоящего лицензионного соглашения и согласны с ними.

10. Если почему-либо вы не согласны ни с этим лицензионным соглашением, ни с соглашением, приведенным на нашем сайте в Интернете, вам необходимо удалить файлы дистрибутива "RS232 logger ActiveX" с ваших устройств хранения информации и прекратить использование "RS232 logger ActiveX".

1.5 Системные требования

Для запуска программы необходима одна из следующих ОС:

- Windows 95 (потребуется установить последнее обновление comctl.dll, пожалуйста, посетите наш сайт и скачайте обновление);
- Windows 95 OSR2;
- Windows 98;
- Windows Me.

Если вы использует наше ПО на следующих операционных системах, то вам необходимы права администратора для инсталляции и регистрации:

- Windows NT4 (SP4 и старше).

- Windows 2000 Professional;
- Windows 2000 Server;
- Windows XP x86 и x64;
- Windows 2003 Server x86 и x64;
- Windows Vista (все версии) x86 и x64.

Вам также необходимо установить Internet Explorer 5.0 или более поздней версии для того, чтобы просматривать справку по программе.

Необходимо установить и настроить Advanced Serial Data Logger перед использованием данного ActiveX компонента.

1.6 Процесс установки

Выйдите на время установки из работающей программы Advanced Serial Data Logger.

Запустите файл установки.

По умолчанию, "RS232 logger ActiveX" будет установлен в папку "/Programs Files/Advanced Serial Data Logger" вашего системного диска, но вы можете указать свой путь.

Во время процесса установки мастер инсталляции спросит вашего разрешения на регистрацию "RS232 logger ActiveX" в системе. Если вы откажитесь от этой процедуры, то вам необходимо будет запустить !register.bat в папке с программой.

2 Справочная информация

Эта секция файла справки содержит подробную информацию о функциях API содержащимся в **asdlog**.

Справочная информация по **asdlog** разделена на следующие категории.

- [Интерфейсы](#) ⁴

2.1 Интерфейсы

Эта секция содержит справочную информацию о COM интерфейсах предлагаемых **RS232 logger ActiveX**. Следующие интерфейсы можно использовать при работе с **RS232 logger ActiveX**.

- [ITLoggerControlAX](#) ⁵
- [ITLoggerControlAXEvents](#) ¹³

2.1.1 ITLoggerControlAX

RS232 logger ActiveX Object.

Методы интерфейса **ITLoggerControlAX** представлены в таблице ниже

Список методов		
Метод	RegisterDataNotify ⁶	Функция регистрирует ваше приложение для получения событий от программы. Без вызова этой процедуры ваша программа не будет получать никаких уведомлений о событиях. Если вы не хотите использовать события, то не вызывайте этот метод и вы сэкономите ресурсы компьютера.
Метод	GetConfigurations ⁶	Функция возвращает строку с перечнем всех конфигурация программы. Строка содержит имена из выпадающего списка, в главном окне программы. Каждое имя в строке разделено символами CR+LF. Пример: COM1+CRLF+COM2+CRLF
Метод	GetConfigID ⁷	Получает идентификатор (ID) конфигурации по ее имени. Позднее вы должны использовать этот идентификатор для последующих вызовов.
Метод	Send ⁸	Отправляет данные через логгер данных. Эта функция позволяет как отослать данные через порт, так и выполнить симуляцию этого.
Метод	StartPause ⁹	Останавливает/запускает прием данных из порта для указанной конфигурации. Функция закрывает и открывает источник данных.
Метод	SetLicenseKey ⁹	Метод позволяет указать ваш лицензионный ключ, который снимет все ограничения с компонента.
Метод	SetOption ¹⁰	Устанавливает значение указанного свойства у конфигурации. Некоторые свойства можно изменять во время работы, без сохранения.
Метод	GetOption ¹¹	Возвращает значение указанного свойства у конфигурации. Некоторые значения необходимо декодировать.
Метод	SaveApplyOptions ¹²	Позволяет сохранить и применить все свойства для указанной конфигурации. Программа перезагрузит все используемые модули и переинициализирует их.

Требования:

Заголовочный файл: объявлено в asdlog.h

Импорт библиотеки: используйте asdlog.lib

2.1.1.1 ITLoggerControlAX::RegisterDataNotify

Функция регистрирует ваше приложение для получения событий от программы. Без вызова этой процедуры ваша программа не будет получать никаких уведомлений о событиях. Если вы не хотите использовать события, то не вызывайте этот метод и вы сэкономите ресурсы компьютера.

[C++]

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

[Visual Basic]

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

Параметры

ConfigID

[in] Идентификатор (ID) конфигурации, который вы должны получить перед этим с помощью [GetConfigID](#)⁷.

NotifyType

[in] Идентификатор типа события, для которого вы хотите зарегистрироваться: 0 - события получения необработанных (RAW) данных; 1 - события получения и обработки пакета данных; 2 - уведомления о событиях программы. Если вы хотите использовать все события, то вы должны вызвать функцию несколько раз.

Требования:

Заголовочный файл: объявлено в asdlog.h

Импорт библиотеки: используйте asdlog.lib

Смотрите также:

[Интерфейс ITLoggerControlAX](#)⁵

2.1.1.2 ITLoggerControlAX::GetConfigurations

Функция возвращает строку с перечнем всех конфигурация программы. Строка содержит имена из выпадающего списка, в главном окне программы. Каждое имя в строке разделено символами CR+LF. Пример: COM1+CRLF+COM2+CRLF

[C++]

```
BSTR GetConfigurations();
```

[Visual Basic]

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

Требования:

Заголовочный файл: объявлено в asdlog.h

Импорт библиотеки: используйте asdlog.lib

Смотрите также:

[Интерфейс ITLoggerControlAX](#) ⁵

2.1.1.3 ITLoggerControlAX::GetConfigID

Получает идентификатор (ID) конфигурации по ее имени. Позднее вы должны использовать этот идентификатор для последующих вызовов.

[C++]

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

[Visual Basic]

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

Параметры

ConfigName

[in] Имя конфигурации, которое вы можете получить с помощью функции [GetConfigurations](#) ⁶.

Требования:

Заголовочный файл: объявлено в asdlog.h

Импорт библиотеки: используйте asdlog.lib

Смотрите также:

[Интерфейс ITLoggerControlAX](#) ⁵

2.1.1.4 ITLoggerControlAX::Send

Отправляет данные через логгер данных. Эта функция позволяет как отослать данные через порт, так и выполнить симуляцию этого.

[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
```

Параметры

ConfigID

[in] Идентификатор (ID) конфигурации, который вы должны получить перед этим с помощью [GetConfigID](#) ⁷.

SendType

[in] Идентификатор типа отправки данных: 0 - симуляция приема данных; 1 - симуляция отправки данных; 2 - отправка данных через порт.

DataType

[in] Тип данных для отправки: 0 - двоичные данные; 1 - текстовая строка, которая будет преобразована и все символы, соответствующие маске #AA будут заменены на символ с соответствующим шестнадцатеричным кодом.

Data

[in] Данные для отправки.

Требования:

Заголовочный файл: объявлено в asdlog.h

Импорт библиотеки: используйте asdlog.lib

Смотрите также:

[Интерфейс ITLoggerControlAX](#) ⁵

2.1.1.5 ITLoggerControlAX::StartPause

Останавливает/запускает прием данных из порта для указанной конфигурации. Функция закрывает и открывает источник данных.

[C++]

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

[Visual Basic]

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

Параметры

ConfigID

[in] Идентификатор (ID) конфигурации, который вы должны получить перед этим с помощью [GetConfigID](#)⁷.

State

[in] Значение больше 0 означает, что источник данных будет открыт (если порт не был открыт ранее). Если State=0, то источник данных будет закрыт.

Требования:

Заголовочный файл: объявлено в asdlog.h

Импорт библиотеки: используйте asdlog.lib

Смотрите также:

[Интерфейс ITLoggerControlAX](#)⁵

2.1.1.6 ITLoggerControlAX::SetLicenseKey

Метод позволяет указать ваш лицензионный ключ, который снимет все ограничения с компонента.

[C++]

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

[Visual Basic]

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

Параметры

Key

[in] Ваш лицензионный ключ вида: AAAA-BBBB-CCCC-DDDD

Требования:

Заголовочный файл: объявлено в asdlog.h

Импорт библиотеки: используйте asdlog.lib

Смотрите также:

[Интерфейс ITLoggerControlAX](#)⁵

2.1.1.7 ITLoggerControlAX::SetOption

Устанавливает значение указанного свойства у конфигурации. Некоторые свойства можно изменять во время работы, без сохранения.

[C++]

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

[Visual Basic]

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

Параметры

ConfigID

[in] Идентификатор (ID) конфигурации, который вы должны получить перед этим с помощью [GetConfigID](#)⁷.

Option

[in] Имя параметра/свойства. Это имя вы можете найти в реестре Window (смотри раздел ниже [SaveApplyOptions](#))^[12].

Value

[in] Значение параметра вида True или False для значений с типом данных boolean.

Требования:

Заголовочный файл: объявлено в asdlog.h

Импорт библиотеки: используйте asdlog.lib

Смотрите также:

[Интерфейс ITLoggerControlAX](#)^[5]

2.1.1.8 ITLoggerControlAX::GetOption

Возвращает значение указанного свойства у конфигурации. Некоторые значения необходимо декодировать.

[C++]

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

[Visual Basic]

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

Параметры

ConfigID

[in] Идентификатор (ID) конфигурации, который вы должны получить перед этим с помощью [GetConfigID](#)^[7].

Option

[in] Имя параметра/свойства. Это имя вы можете найти в реестре Window (смотри раздел ниже [SaveApplyOptions](#))^[12].

Value

[out] Значение параметра вида True или False для значений с типом данных boolean.

Требования:

Заголовочный файл: объявлено в asdlog.h
Импорт библиотеки: используйте asdlog.lib

Смотрите также:

[Интерфейс ITLoggerControlAX](#)⁵⁾

2.1.1.9 ITLoggerControlAX::SaveApplyOptions

Позволяет сохранить и применить все свойства для указанной конфигурации. Программа перезагрузит все используемые модули и переинициализирует их. Программа сохранит все свойства у указанной ветви реестра:

HKEY_LOCAL_MACHINE\SOFTWARE\AGG Software\Advanced Serial Data Logger\ {8BB712F0-490F-499F-B616-3E809FAE4A22}.

Если данная ветвь не доступна для записи текущему пользователю, то программа будет использовать

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

Если вы используете INI файл (вы должны выбрать этот режим в программе), тогда, вместо реестра, программа сохранит настройки в соответствующий INI файл.

Замечание: идентификатор "{8BB712F0-490F-499F-B616-3E809FAE4A22}" может быть различным на разных компьютерах.

[C++]

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

[Visual Basic]

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

Параметры

ConfigID

[in] Идентификатор (ID) конфигурации, который вы должны получить перед этим с помощью [GetConfigID](#)⁷⁾.

DoSave

[in] Если этот параметр > 0, то программа применит и сохранит все настройки, иначе программа только применит их без сохранения и после перезапуска будут считаны настройки из реестра или INI файла.

Требования:

Заголовочный файл: объявлено в asdlog.h

Импорт библиотеки: используйте asdlog.lib

Смотрите также:

[Интерфейс ITLoggerControlAX](#)^[5]

2.1.2 ITLoggerControlAXEvents

Интерфейс событий RS232 logger ActiveX. События интерфейса ITLoggerControlAXEvents представлены в таблице ниже.

Список событий		
Метод	OnData ^[13]	Объект получает это событие когда Advanced Serial Data Logger получает новые блок необработанных данных.
Метод	OnDataPacket ^[14]	Объект получает это событие когда новый пакет был принят и обработан парсером Advanced Serial Data Logger.
Метод	OnEvent ^[15]	Объект получает это событие когда возникает внутреннее событие логгера.

Требования:

Заголовочный файл: объявлено в asdlog.h

Импорт библиотеки: используйте asdlog.lib

2.1.2.1 ITLoggerControlAXEvents::OnData

Объект получает это событие когда Advanced Serial Data Logger получает новые блок необработанных данных.

[C++]

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

[Visual Basic]

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

Параметры*DataSent*

[in] Если DataSent>0 то данные (Data) были отправлены, иначе данные (Data) были приняты.

Data

[in] Буфер данных.

Требования:

Заголовочный файл: объявлено в asdlog.h

Импорт библиотеки: используйте asdlog.lib

Смотрите также:

[Интерфейс ITLoggerControlAXEvents](#) ¹³

2.1.2.2 ITLoggerControlAXEvents::OnDataPacket

Объект получает это событие когда новый пакет был принят и обработан парсером Advanced Serial Data Logger.

[C++]

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

[Visual Basic]

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

Параметры*DataSent*

[in] Если DataSent>0 то данные (Data) были отправлены, иначе данные (Data) были приняты.

VarNames

[in] Одномерный массив, которые содержит имена переменных парсера. Эти переменные заданы в модуле-парсере логгера.

VarValues

[in] Одномерный массив, которые содержит значения переменных парсера. Количество элементов в обоих массивах одинаково.

Требования:

Заголовочный файл: объявлено в asdlog.h

Импорт библиотеки: используйте asdlog.lib

Смотрите также:

[Интерфейс IILoggerControlAXEvents](#) 

2.1.2.3 IILoggerControlAXEvents::OnEvent

Объект получает это событие когда возникает внутреннее событие логгера.

[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
)
```

Параметры

Event

[in] Внутренний идентификатор события.

EventName

[in] Имя события вида PORT-OPEN.

ArgNames

[in] Одномерный массив, которые содержит имена дополнительных атрибутов, которые используются в этом событии (например имя лог-файла).

ArgValues

[in] Одномерный массив, которые содержит значения дополнительных атрибутов, которые используются в этом событии . Количество элементов в обоих массивах одинаково.

Требования:

Заголовочный файл: объявлено в asdlog.h

Импорт библиотеки: используйте asdlog.lib

Смотрите также:

[Интерфейс IILoggerControlAXEvents](#)  ¹³