OMRON

Power PMAC PDK User Manual

User Manual

O033-E-03

Copyright Information

To report errors or inconsistencies, email:

Delta Tau Data Systems, Inc. Technical Support

Email: **odt-support@omron.com**

Website: **https://www.omron.com/global/en/business/industrial_automation/**

Operating Conditions

All Delta Tau Data Systems, Inc. motion controller products, accessories, and amplifiers contain static sensitive components that can be damaged by incorrect handling. When installing or handling Delta Tau Data Systems, Inc. products, avoid contact with highly insulated materials. Only qualified personnel should be allowed to handle this equipment.

In the case of industrial applications, we expect our products to be protected from hazardous or conductive materials and/or environments that could cause harm to the controller by damaging components or causing electrical shorts. When our products are used in an industrial environment, install them into an industrial electrical cabinet or industrial PC to protect them from excessive or corrosive moisture, abnormal ambient temperatures, and conductive materials. If Delta Tau Data Systems, Inc. products are directly exposed to hazardous or conductive materials and/or environments, we cannot guarantee their operation.

Security Measures

To prevent computer viruses, install antivirus software on a computer where you use this software.

Make sure to keep the antivirus software updated.

Keep your computer's OS updated to avoid security risks caused by vulnerability in the OS.

Always use the highest version of this software to add new features, increase operability, and enhance security.

Manage usernames and passwords for this software carefully to protect them from unauthorized uses.

Set up a firewall (E.g., disabling unused communication ports, limiting communication hosts, etc.) on a network for a control system and devices to separate them from other IT networks.

Make sure to connect to the control system inside the firewall.

Use a virtual private network (VPN) for remote access to a control system and devices from this software.

Warning

A Warning identifies hazards that could result in personal injury or death. It precedes the discussion of interest.

Caution

A Caution identifies hazards that could result in equipment damage. It precedes the discussion of interest.

Note

A Note identifies information critical to the user's understanding or use of the equipment. It follows the discussion of interest.

REVISION HISTORY				
REV.	**DESCRIPTION**	**DATE**	**CHG**	**APPVD**
1	Description of sample project.	9/11/2017	DG	AG
2	Upgrading of build and download to support all PMAC types and match the Power PMAC IDE build and download.	11/21/2023	DG	AH
3	Dart socket communication samples in managed and unmanaged C++.	10/28/2024	SCP	DG

Table of Contents

Introduction

The ODT.PowerPmacComLib namespace is reserved for the communication library of the Power Programmable Multi Axis Controller (PMAC), which is a product of Delta Tau Data Systems, Inc. This library provides motion (GpAscii), host operating system (Terminal), and FTP communications (FTP Client) for the Power PMAC.

Namespace	ODT.PowerPmacComLib
Assembly	ODT.PowerPmacComLib (in *ODT.PowerPmacComLib.dll)* Version: 3.0.0.0 (3.0.0.0)

To start programming with this library, use the factory class Connect to display a communication interface.

Version updates
PDK 4.6.3.x

In this version of the PDK, build and download methods of ODT.PowerPmacBuildAndDownload assembly have been modified to match the build and download process of the IDE. In addition, the following major changes have been added to this revision:

Generating Download's Dependent Files

All the dependent files required by the download are generated from the PDK and no manual addition and copying of those dependent files are necessary. Exclude.txt, rsync-filter.txt and pp_proj.ini are generated from the PDK.

Compilers Installation

In the previous versions of the PDK, a small subset of the compiler and its tools were distributed to enable the building and downloading. In this version the full compiler set is distributed and installed with the PDK installation. Downloadtools.exe has been removed and replaced by the compilers.msi file. If the compilers have been already installed on a machine, the PDK installation will not install the twice.

To distribute a customer's developed HMI application to their end users, the compiles.msi file can be distributed to the end users and included in any installation. There is no need to put the downloadtools.exe in the bin directory of the PDK developed application. By migrating to compilers installation and distribution, the PDK behaviour matches the Power PMAC IDE.

Project Build and Download Matches the Build and Download Functionality of IDE 4.6.3.x

Download method of the PDK is updated to match the functionality of the Download All Files of the Power PMAC IDE.

Encrypted Projects can be Downloaded in PDK 4.6.3.x

EcatConfig and systemsetup config are downloaded during the build and download process.

PDK 4.6.4.x

In this version of the PDK, TCP communication class for dart application is added in ODT.PowerPmacComLib assembly. In addition, the following major changes have been added to this revision:

Added VC++ samples

Following samples are added for VC++,

VC++ Managed Samples	• Communication Library Demo • Single File Downloader • Dart Socket Demo
VC++ Unmanaged Samples	• Build and download • Gather Control • Dart Socket Demo

PDK Compatibility Chart

FW / PDK	PDK Compatibility					
	2.6.0.0	2.6.1.0	2.7.0.0	2.6.7.1	2.8.0.0	2.8.0.1
PDK 4.6.2.x or Older	OK	OK	OK	OK	Not Supported	Not Supported
PDK 4.6.3.x	OK	OK	OK	OK	OK	OK
PDK 4.6.4.x	OK	OK	OK	OK	OK	OK

SSH Configuration

Depending on the Power PMAC SSH configuration, the user should take caution that if the default SSH settings is kept then there is a possibility that the SSH connection can timeout if the connection stays inactive for a period. These settings can be changed via Power PMAC IDE from the Tools->Options->Power PMAC->Communications Settings page. “Client Alive Counter Max” and “Client Alive interval” control the SSH connection timeout.

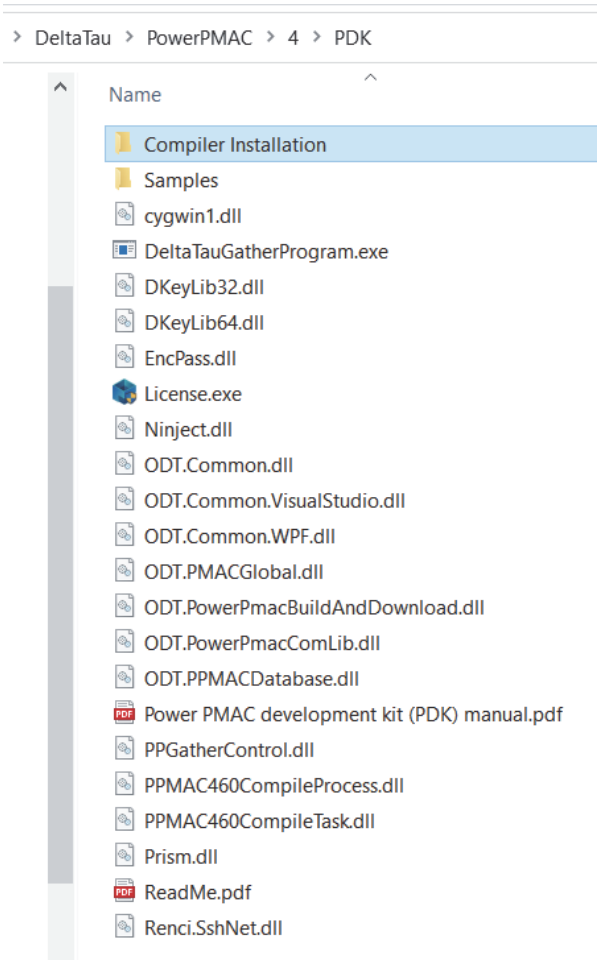
1. Client Alive Counter Max: Sets the number of client alive messages which may be sent without SSH receiving any messages back from the client. Default value is 3
2. Client Alive Interval: Sets the number of seconds that the server will wait before sending a null packet to the client. Default value is 15

Timeout is “Client Alive Counter Max” * “Client Alive interval”

To disable the timeout, these two values should be set to zero.

Installation

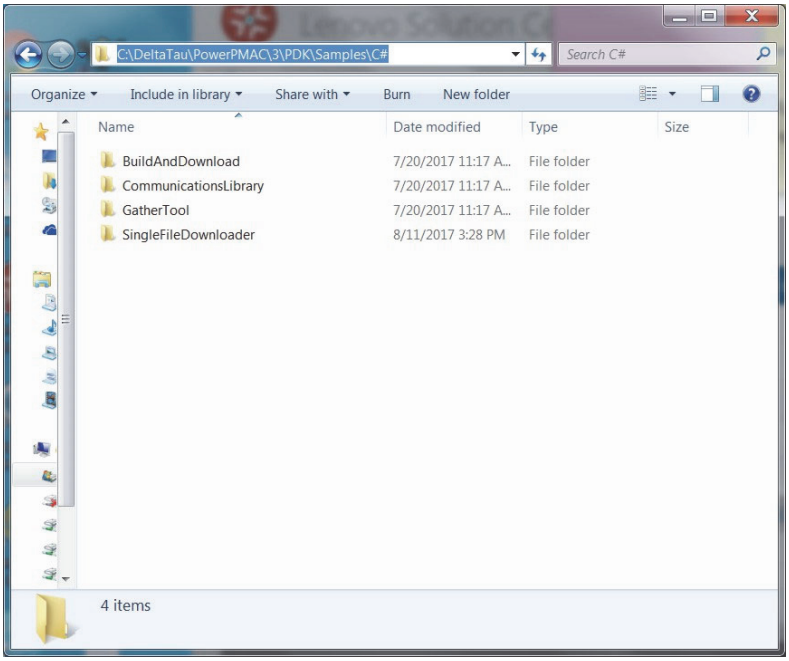
After successful installation the folder structure will look…Default installation path is C:\Deltatau



Note

Readme.pdf: Explains how to authorize the application with PDK on customer machine. Customer does not need to purchase the license. Only developers need to purchase the license.

The example projects are in the sample folder. There are ten example projects that are installed. The sample folder looks like this…



PDK Project Examples

Several example projects can be found in the PDK's installation folder. By default, its location is as follows:

C:\DeltaTau\Power PMAC\~\PDK\Samples

As of the date of updating this manual, there are four examples in C# and six examples in C ++. The example environment is

- VS 2015
- Language C#
- Language C ++

C#:

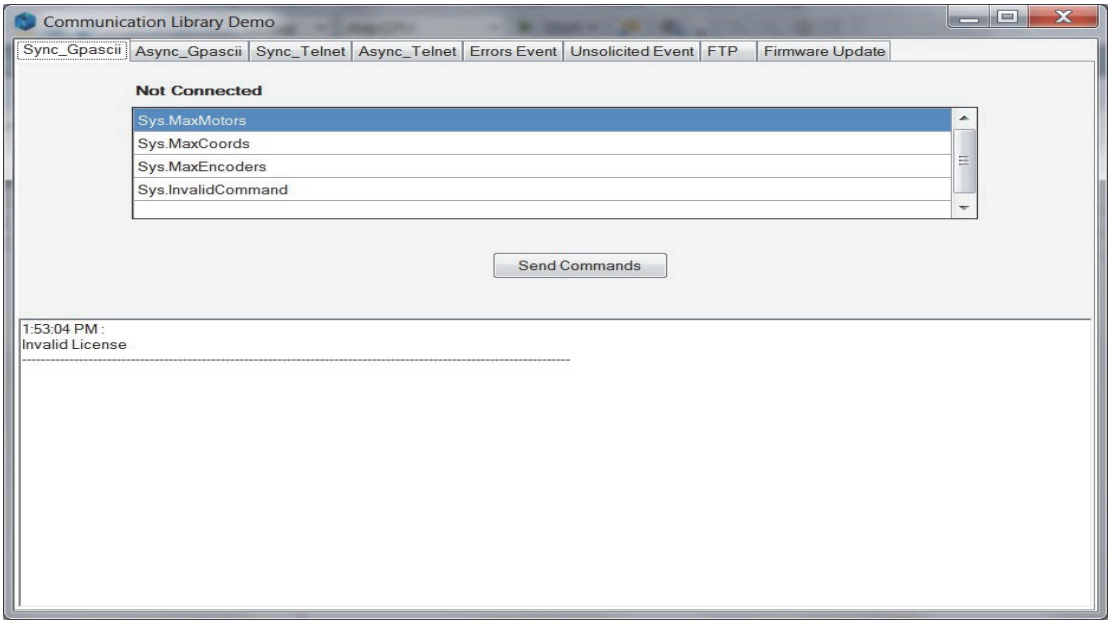
Project	Description
CommunicationLibrary	Example project shows the available communication objects clients and how to use it in HMI program.
BuildAndDownload	Example project shows how to Build and download the project from external HMI software
GatherTool	Example project shows how to gather Power PMAC data from external HMI software.
SingleFileDowloader	Example project shows how to download a single file to Power PMAC from external HMI application

C++:

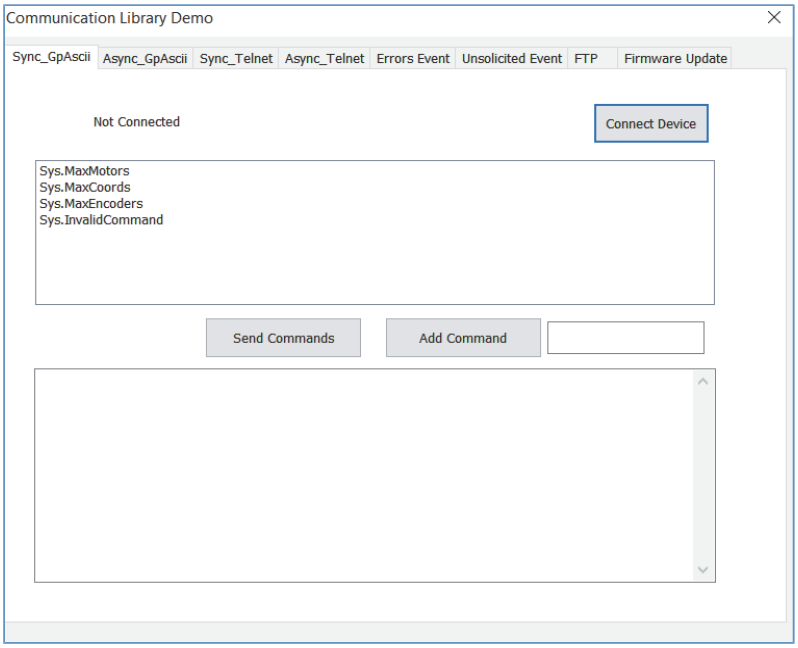
Project	Description
CommunicationLibrary Managed	Example project shows the available communication objects clients and how to use it in HMI program.
BuildAndDownload Unmanaged	Example project shows how to Build and download the project from external HMI software
GatherTool Unmanaged	Example project shows how to gather Power PMAC data from external HMI software.
SingleFileDownloader Managed	Example project shows how to download a single file to Power PMAC from external HMI application
DartSocketDemo Managed	Example project shows how to read and write user shared memory and P Variable using socket communication.
DartSocketDemo UnManaged	Example project shows how to read and write user shared memory and P Variable using socket communication.

Sample Project Screenshots

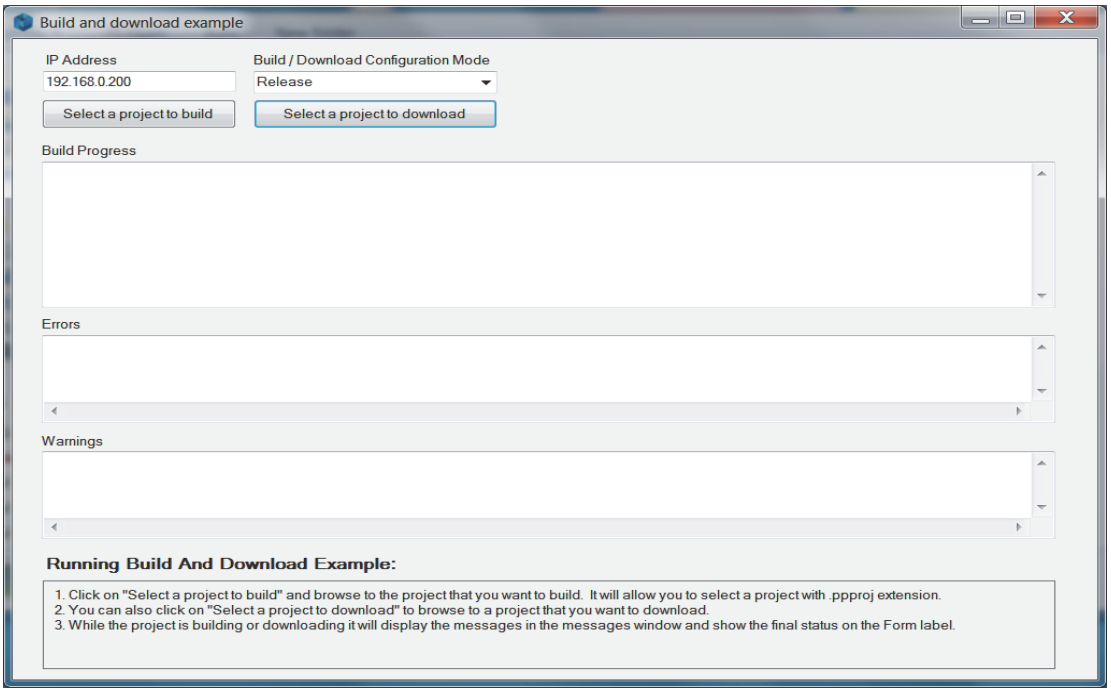
CommunicationLibrary C#:



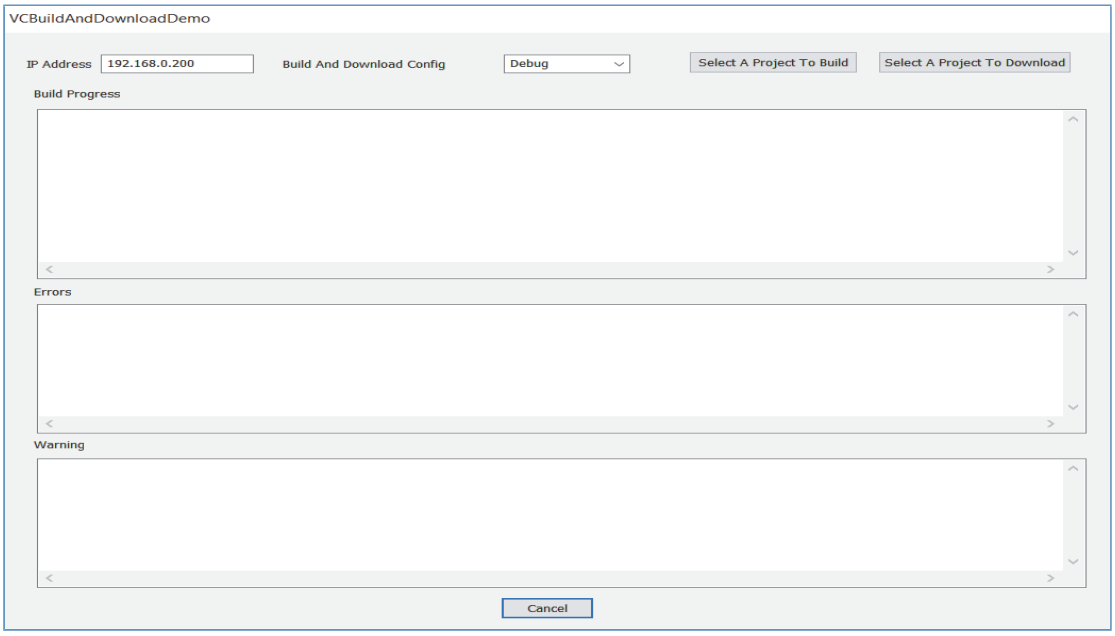
CommunicationLibrary C++:



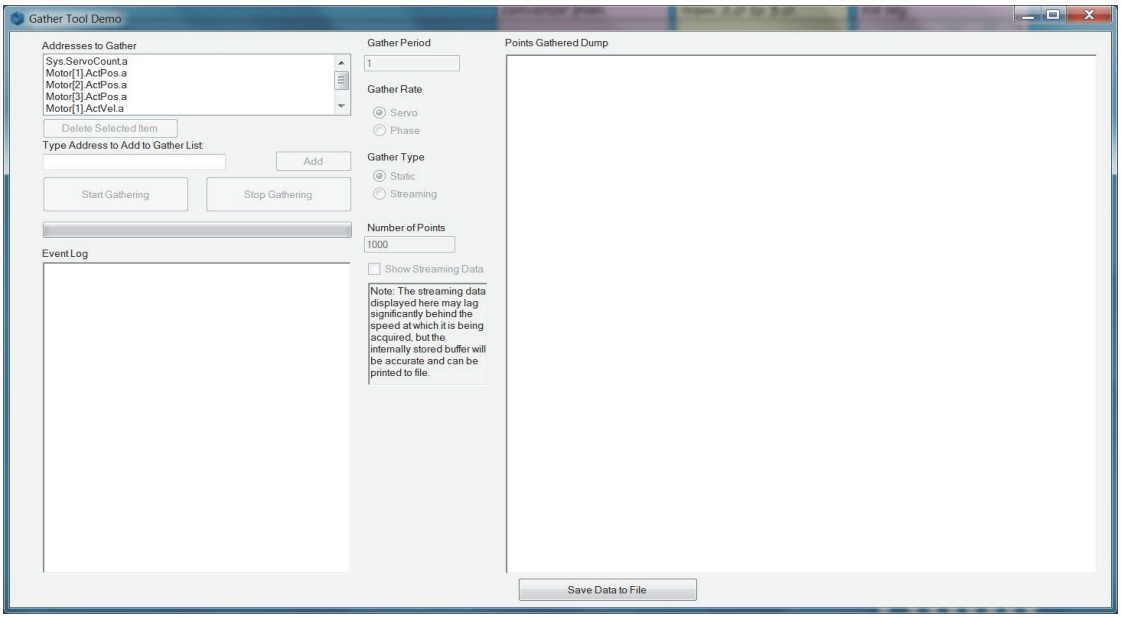
BuildAndDownload C#:



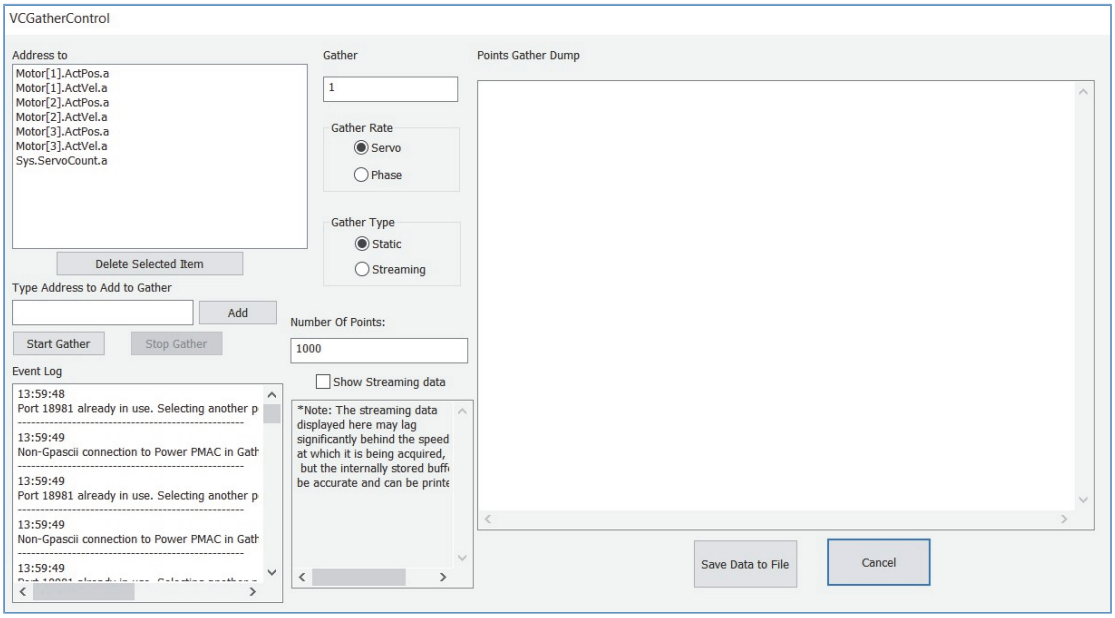
BuildAndDownload C++:



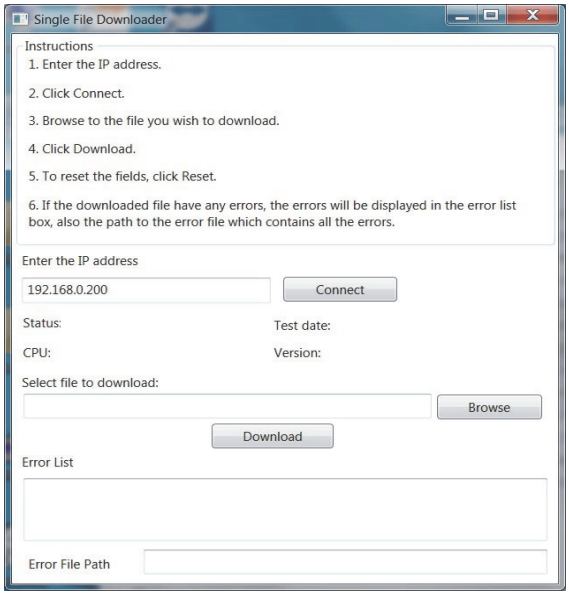
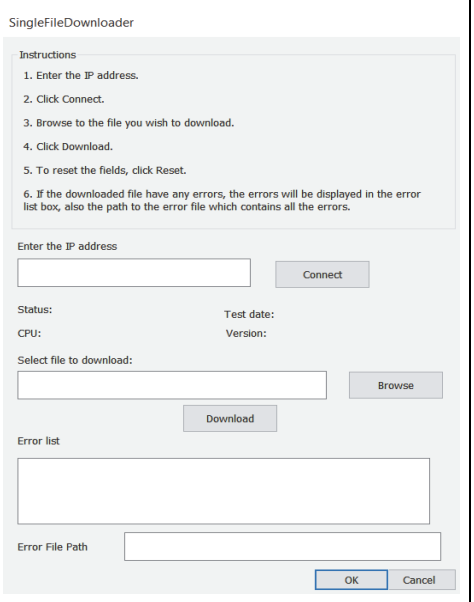
GatherTool C#:



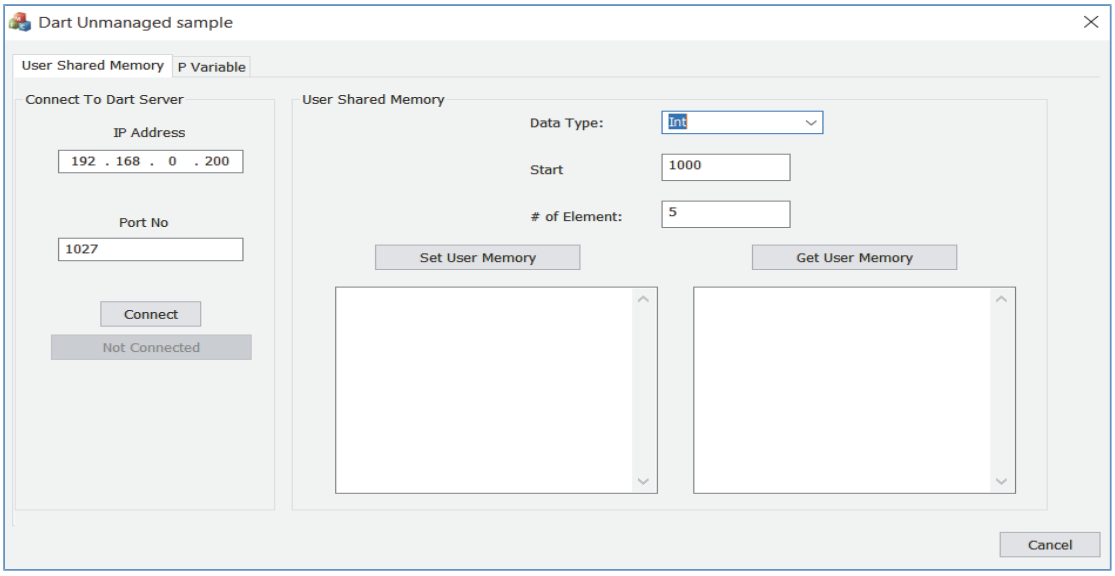
GatherTool C++:



SingleFileDownloader:

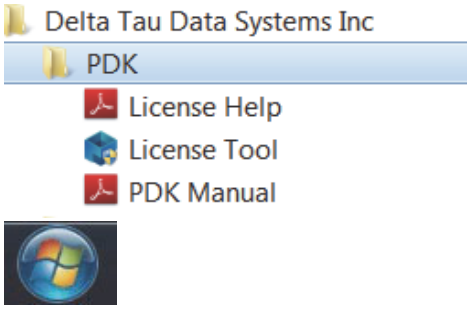
C#	C++
	

Dart Socket Demo:

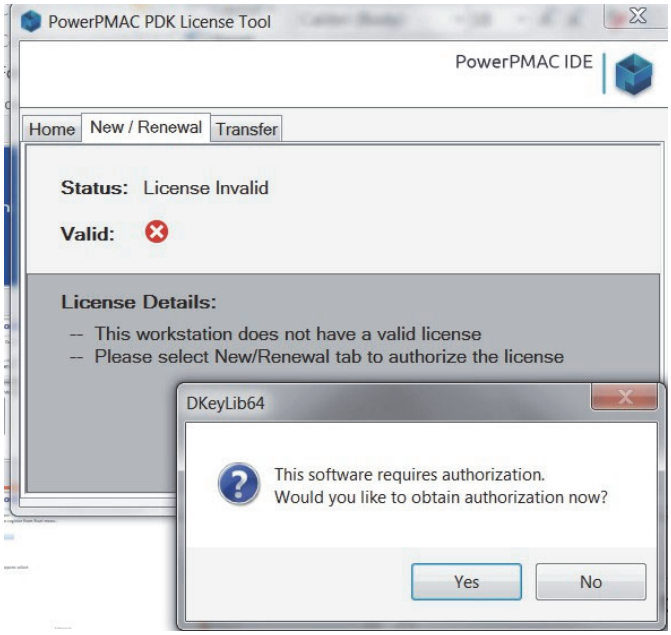


How to Authorize PDK Licensing

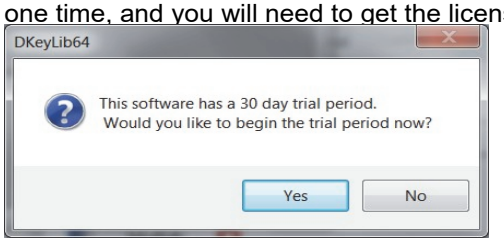
1. After completing the installation go to the start menu to authorize the license.
2. Run License Tool



3. When the license tool opens select New/Renewal tab. The question will be asked. Select “No” and you will get another message box with a question.

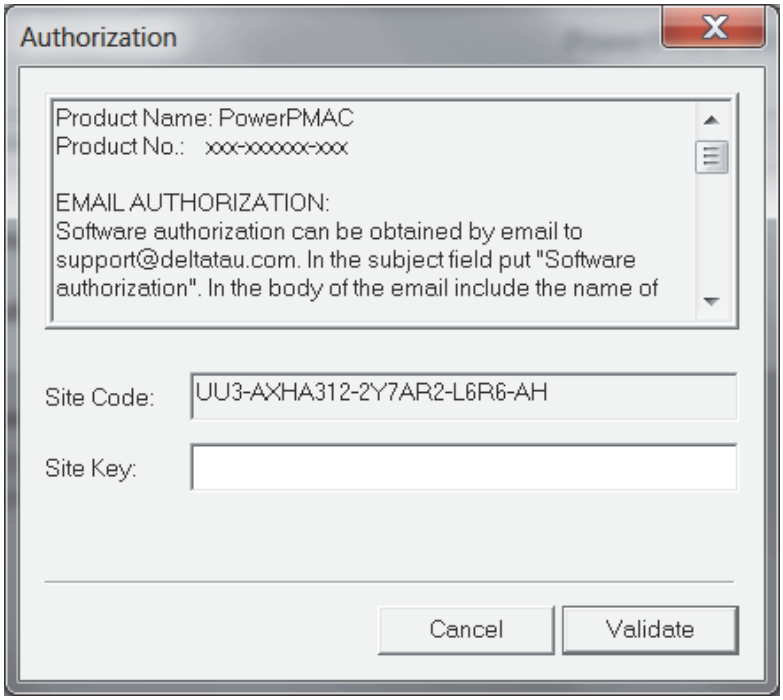


4. At this point it will provide you with a default trial period of 30 days. This trial period is one time, and you will need to get the license key before it expires.



5. Please select “YSE” to start the 30-day trial. If you click “Yes” at step 3 then authorization dialog appears.

6. The Authorization dialog explains how to get the authorization site key. Please follow the instructions and email customer support at ODT to get the 120-day license authorization site key. Once the customer support receives the site code license key will be emailed back.
7. Use the site key to validate



Classes

AsyncDataArgs

Asynchronous communication response argument.

Syntax

Syntax	Description
C#	**public class** AsyncDataArgs : EventArgs
Visual Basic	**public class** AsyncDataArgs Inherits the EventArgs
Visual C++	**public ref class** AsyncDataArgs : **public** EventArgs

Inheritance Hierarchy in the .NET Framework

System.Object
  System.EventArgs
    ODT.PowerPmacComLib.AsyncDataArgs

AsyncDataArgs Members

The AsyncDataArgs type exposes the members explained below.

Constructors

Name	Description
AsyncDataArgs	Constructor for AsyncDataArgs. Accepts response data as string.

Properties

Name	Description
Response	Communication response as string. This property is read-only.

AsyncDataArgs Constructor

Constructor for AsyncDataArgs that accepts response data as string.

Syntax

Syntax	Description

C#	Public AsyncDataArgs {
	string *responseE*
	}

Visual Basic	**Public Sub** New {
	responseE **As** String _
	}

Visual C++	**public:**
	AsyncDataArgs {
	String^ responseE
	}

Parameters

responseE
Type: System.String in the .NET Framework
Response data to the Power PMAC command.

AsyncDataArgs Properties

The AsyncDataArgs type exposes the property explained in the following table.

Name	Description
Response	Communication response as string. This property is read-only.

AsyncDataArgs.Response Property

Communication response as string. This property is read-only.

Syntax	Description
C#	public string Response { get; }
Visual Basic	**Public ReadOnly Property** Response **As** String Get

Visual C++	**public:**
	property String ^ Response {
	String ^ get () ;
	}

ComErArgs

Communication error argument. This class wraps any error messages that may occur at socket-level communication.

Syntax

Syntax	Description
C#	public class ComErArgs : EventArgs
Visual Basic	**Public class** ComErArgs _ Inherits EventArgs
Visual C++	**public ref class** ComErArgs : **public** EventArgs

Inheritance Hierarchy in the .NET Framework

System.Object
  System.EventArgs
    ODT.PowerPmacComLib.ComErArgs

ComErArgs Members

Properties

The ComErArgs type exposes the property explained in the following table.

Name	Description
ErrorMessage	Error message as string. This property is read-only.

ComErArgs Properties

The ComErArgs type exposes the property explained in the following table.

Name	Description
ErrorMessage	Error message as string. This property is read-only.

ComErArgs.ErrorMessage Property

Error message as string. This property is read-only.

Syntax	Description
C#	public string ErrorMessage { get; }
Visual Basic	**Public Read-Only Property** ErrorMessage **As** String Get

Visual C++	**public:**
	property String^ ErrorMessage {
	String* get ();
	}

CommandArgs

Operating system response argument.

Syntax

Syntax	Description
C#	**public class** Command Args : EventArgs
Visual Basic	**Public Class** CommandArgs _ Inherits EventArgs
Visual C++	**public ref class** CommandArgs : **public** EventArgs

Inheritance Hierarchy in the .NET Framework

System.Object
  System.EventArgs
    ODT.PowerPmacComLib.CommandArgs

CommandArgs Members

The CommandArgs type exposes the members explained below.

Constructor

Name	Description
CommandArgs	Constructor for CommandArgs that sends arguments for commands as string.

Properties

Name	Description
ExCommandMessage	Error message.

CommandArgs Constructor

The constructor for CommandArgs sends arguments for commands as string.

Syntax

Syntax	Description

C#	**public** CommandArgs {
	String exmessage
	}

Visual Basic	**Public Sub** New { _ *exmessage* As String _ }

Visual C++	**public:**
	CommandArgs {
	String^ exmessage
	}

Parameters

exmessage
Type: System.String in the .NET Framework
Message to the Power PMAC command.

CommandArgs Properties

Name	Description
ExCommandMessage	Error message.

ExCommandMessage Property

This property is an error message.

Syntax

Syntax	Description
C#	**public string** ExCommandMessage { **get**; }
Visual Basic	**Public ReadOnly Property** ExCommandMessage **As** String Get
Visual C++	**public:** **property** String^ ExCommandMessage {

Syntax	Description
	String^ get (); }

CommunicationGlobals

A static class for the PowerPmacComm library.

Syntax

Syntax	Description
C#	**public static class** CommunicationGlobals
Visual Basic	**Public NotInheritable Class** CommunicationGlobals
Visual C++	**public ref class** CommunicationGlobals **abstract sealed**

Inheritance Hierarchy in the .NET Framework

System.Object
    ODT.PowerPmacComLib.CommunicationGlobals

CommunicationGlobals Methods

The CommunicationGlobals type exposes the methods explained below.

Name	Description
CheckVersion	Confirms the version of PMAC firmware. Version string must be in the format of major, minor, minor revision, and build number. Version string must be a valid number.
IsGPAsciiReady	Checks whether GPAscii can start. Can also verify whether Power PMAC is initialized.
IsWow64Process	Confirms whether the Microsoft® Windows® operating system is 64-bit.

CommunicationGlobals Properties

The CommunicationGlobals type exposes the members explained below.

IsWow~~ProcessProperty

Confirms whether the Microsoft® Windows® operating system is 64-bit.

Syntax

Syntax	Description
C#	**public static** bool IsWINDOWS_OS { **get**; }
Visual Basic	**Public Shared ReadOnly Property** IsWINDOWS_OS **As** Boolean Get

Visual C++	**public:**
	static property bool IsWINDOWS_OS {
	bool **get** ();
	}

CommunicationGlobals.PowerPMACValidation

A static class for confirming the license status of the IDE, components, and Servo Analyzer.

Syntax

Syntax	Description
C#	**public class** PowerPMACValidation
Visual Basic	**Public Class** PowerPMACValidation
Visual C++	**public ref class** PowerPMACValidation

Inheritance Hierarchy in the .NET Framework

System.Object
    ODT.PowerPmacComLib.CommunicationGlobals.PowerPMACValidation

CommunicationGlobals.PowerPMACValidation Members

This static class is only available for the PowerPMACComm library. The CommunicationGlobals.PowerPMACValidation type exposes the members described below.

Constructors

Name	Description
CommunicationGlobals.PowerPMACValidation	Initializes a new instance of the CommunicationGlobals.PowerPMACValidation class.

Methods

Name	Description
IsValidComponent	Confirms whether licenses for the Power PMAC and components are valid.
isValidIDE	Confirms whether licenses for the Power PMAC and IDE are valid.
IsValidServoAnalyzer	Confirms whether license for the Servo Analyzer is valid.

CommunicationGlobals.PowerPMACValidation Constructor

Initializes a new instance of the CommunicationGlobals.PowerPMACValidation class.

Syntax

Syntax	Description
C#	**public** PowerPMACValidation()
Visual Basic	**Public Sub** New
Visual C++	**public:** PowerPMACValidation()

CommunicationGlobals.PowerPMACValidation Methods

The CommunicationGlobals.PowerPMACValidation type exposes the members described below.

Name	Description
IsValidComponent	Confirms whether the licenses for the Power PMAC and components are valid.
isValidIDE	Confirms whether the licenses for the Power PMAC and IDE are valid.
IsValidServoAnalyzer	Confirms whether the license for the Servo Analyzer is valid.

CommunicationGlobals.PowerPMACValidation.IsValidComponent Method

This function confirms whether a component license for the Power PMAC is valid.

Syntax

Syntax	Description
C#	**public static** bool IsValidComponent()
Visual Basic	**Public Shared Function** IsValidComponent **As** Boolean
Visual C++	**public:** **static** bool IsValidComponent()

Return Value

True on Valid Component License.

CommunicationGlobals.PowerPMACValidation.IsValidIDE Method

This function confirms whether licenses for the Power PMAC and IDE are valid.

Syntax

Syntax	Description
C#	**public static** bool IsValidIDE()
Visual Basic	**Public Shared Function** IsValidIDE **As** Boolean
Visual C++	**public:** **static** bool IsValidIDE()

Return Value

True on valid IDE license.

CommunicationGlobals.PowerPMACValidation.IsValidServoAnalyzer Method

This function confirms whether a license for the Server Analyzer is valid.

Syntax

Syntax	Description
C#	**public static** bool IsValidServoAnalyzer()
Visual Basic	**Public Shared Function** IsValidServoAnalyzer **As** Boolean
Visual C++	**public:** **static** bool IsValidServoAnalyzer()

Return Value

True on valid Servo Analyzer license.

Connect

This is the primary class of this library and the basis of all communications with the Power PMAC motion controller and its operating system. The motion controller runs as a real-time application in the host operating system. Motion commands can be sent using synchronous or asynchronous Gpascii communication objects.

The operating system of Power PMAC is Linux. Standard Linux commands can be sent using synchronous or asynchronous Terminal communication objects.

This is a factory class that creates the following communication objects and returns the corresponding interfaces:

- Synchronous Gpascii communication object
- Asynchronous Gpascii communication object
- Synchronous Terminal communication object
- Asynchronous Terminal communication object
- Unsolicited Event object
- Get Errors event object
- FTP Client object
- Socket Client object

Syntax

Syntax	Description
C#	**public static class** Connect
Visual Basic	**Public NotInheritable Class** Connect
Visual C++	**public ref class** Connect **abstract sealed**

Inheritance Hierarchy in the .NET Framework

System.Object
    ODT.PowerPmacComLib.Connect

Connect Members

The Connect type exposes the members explained in the following table.

Name	Description
CreateAsyncGpascii	Creates an asynchronous Gpascii communication object and returns as IASyncGpasciiCommunicationInterface. Establish communication with Power PMAC using SSH or PPmacServer.
CreateAsyncTerminal	Creates an asynchronous Terminal communication object and returns as IAsyncTerminalCommunicationInterface. Establish communication with the Power PMAC operating system using SSH or PPmacServer.
CreateFTPClient	Creates an FTP Client object and returns as IFTPClientInterface.
CreateGetErrorsEvent	Creates a GetError event object and returns as IGetErrorsEventInterface.
CreateSyncGpascii	Creates a synchronous Gpascii communication object and returns ISyncGpasciiCommunicationInterface.

Name	Description
	Establishes communication with the Power PMAC using SSH or PPmacServer.
CreateSyncTerminal	Creates a synchronous Terminal communication object and returns as ISyncTerminalCommunicationInterface. Establishes communication with the Power PMAC operating system using SSH or PPmacServer.
CreateUnsolicitedEvent	Creates an Unsolicited event object and returns as the IUnsolicitedEvent interface.
CreateSCPClient	Creates an SCP client object and returns the ISCPClient interface.
GetInvalidLicenseText	Gets text for an invalid license based on the local language of the operating system.
CreateSocketClient	Creates an socket Client object and returns as ITCPSocketCommunicationInterface.

Connect Methods

The Connect type exposes the members explained below.

Name	Description
CreateAsyncGpascii	Creates an asynchronous Gpascii communication object and returns as IASyncGpasciiCommunicationInterface. Establishes communication with Power PMAC using SSH or PPmacServer.
CreateAsyncTerminal	Creates an asynchronous Terminal communication object and returns as IAsyncTerminalCommunicationInterface. Establishes communication with the Power PMAC operating system using SSH or PPmacServer.
CreateFTPClient	Creates an FTP Client object and returns as IFTPClientInterface.
CreateGetErrorsEvent	Creates a GetError event object and returns as IGetErrorsEventInterface.
CreateSyncGpascii	Creates a synchronous Gpascii communication object and returns as ISyncGpasciiCommunicationInterface. Establishes communication with the Power PMAC using SSH or PPmacServer.
CreateSyncTerminal	Creates a synchronous Terminal communication object and returns as ISyncTerminalCommunicationInterface. Establishes communication with the Power PMAC operating system using SSH or PPmacServer.
CreateUnsolicitedEvent	Creates an Unsolicited event object and returns as the IUnsolicitedEvent interface.
CreateSCPClient	Creates an SCP client object and returns as the ISCPClient interface.
GetInvalidLicenseText	Gets text for an invalid license based on the local language of the operating system.
CreateSocketClient	Creates an socket Client object and returns as ITCPSocketCommunicationInterface.

Connect.CreateAsyncGpascii Method

Creates an asynchronous Gpascii communication object and returns as IASyncGpasciiCommunicationInterface.

Establish communication with Power PMAC using SSH or PPmacServer.

Syntax

Syntax	Description

C#	**public static** IAsynGpasciiCommunicationInterface CreateAsyncGpascii{
	CommunicationGlobasl.ConnectionTypes *type*,
	IAsyncGpasciiCommunicationInterface *prevObject*
	}

Visual Basic	**Public Shared Function** CreateAsyncGpascii { _ *type* As CommunicationGlobals.ConnectionTypes, _ *prevObject* As IAsyncGpasciiCommunicationInterface _ } **As** IAsyncGpasciiCommunicationInterface

Visual C++	**public:**
	static IAsyncGpasciiCommunicationInterface^ CreateAsyncGpascii {
	CommunicationGlobals.ConnectionTypes *type*,
	IAsyncGpasciiCommunicationInterface^ *prevObject*
	}

Parameters

Param.	Type	Description
type	ODT.PowerPmacComLib.CommunicationGlobals.ConnectionTypes	Connection: SSH or PPmacServer CommunicationGlobals.ConnectionTypes
prevObject	ODT.PowerPmacComLib.IAsyncGpasciiCommunicationInterface	An asynchronous Gpascii object that was previously created. It is returned for reuse if it is still valid and connected. If an object does not exist, pass null as the parameter.

Return Value

Interface to asynchronous Gpascii object. IAsyncGpasciiCommunicationInterface

Examples

Syntax	Description
C#	private IAsyncGpasciiCommunicationInterface AsyncGPAscii; AsyncGPAscii = Connect.CreateAsyncGpascii(CommunicationGlobals.ConnectionTypes.SSH, null);
VB.NET	Private AsyncGPAscii As IAsyncGpasciiCommunicationInterface = Nothing AsyncGPAscii = Connect.CreateAsyncGpascii(CommunicationGlobals.ConnectionTypes.SSH, null)

Connect.CreateAsyncTerminal Method

Creates an asynchronous Terminal communication object and returns as IAsyncTerminalCommunicationInterface.

Establish communication with the Power PMAC operating system using SSH or PPmacServer.

Syntax

Syntax	Description

C#	**public static** IAsynGpasciiCommunicationInterface CreateAsyncTerminal{
	CommunicationGlobasl.ConnectionTypes *type*,
	IAsyncTerminalCommunicationInterface *prevObject*
	}

Visual Basic	**Public Shared Function** CreateAsyncTerminal { _ *type* As CommunicationGlobals.ConnectionTypes, _ *prevObject* As IAsyncGpasciiCommunicationInterface _ } **As** IAsyncGpasciiCommunicationInterface

Visual C++	**public:**
	static IAsyncTerminalCommunicationInterface^ CreateAsyncTerminal {
	CommunicationGlobals.ConnectionTypes *type*,
	IAsyncGpasciiCommunicationInterface^ *prevObject*
	}

Parameters

Param.	Type	Description
type	ODT.PowerPmacComLib.CommunicationGlobals.ConnectionTypes	Connection: SSH or PPmacServer CommunicationGlobals.ConnectionTypes
prevObject	ODT.PowerPmacComLib.IAsyncTerminalCommunicationInterface	An asynchronous Terminal object that was previously created. It is returned for reuse if it is still valid and connected. If an object does not exist, pass null as the parameter.

Return Value

Interface to asynchronous operating system Terminal object.
IAsyncTerminalCommunicationInterface

Examples

Syntax	Description
C#	private IAsyncTerminalCommunicationInterface OSAsyncTerminal; OSAsyncTerminal = Connect.CreateAsyncTerminal(CommunicationGlobals.ConnectionTypes.SSH, null);
VB.NET	Private OSAsyncTerminal As IAsyncTerminalCommunicationInterface = Nothing

Syntax	Description
	OSAsyncTerminal = Connect.CreateAsyncTerminal(CommunicationGlobals.ConnectionTypes.SSH, null)

Connect.CreateFTPClient Method

Creates an FTP Client object and returns as IFTPClientInterface.

Syntax

Syntax	Description
C#	**public static** IFTPClientInterface CreateFTPClient(CommunicationsGlobals.FTPConnectionTypes *type*, IFTPClientInterface *prevObject*)
Visual Basic	**Public Shared Function** CreateFTPClient (_ *type* **As** CommunicationGlobals.FTPConnectionTypes, _ *prevObject* **As** IFTPClientInterface _) **As** IFTPClientInterface
Visual C++	**public:** **static** IFTPClientInterface^ Create FTPClient (CommunicationGlobals.FTPConnectionTypes *type*, IFTPClientInterface^ *prevObject*)

Parameters

Param.	Type	Description
type	ODT.PowerPmacComLib.CommunicationGlobals.FTPConnectionTypes	Connection: SSH or PPmacServer The type tells the function which internal library function to use. CommunicationGlobals.ConnectionTypes
prevObject	ODT.PowerPmacComLib.IFTPClientInterface	An FTP Client object that was previously created. It is returned for reuse if it is still valid and connected. If an object does not exist, pass null as the parameter.

Return Value

Interface to GetErrors event object. IFTPClientInterface

Examples

Syntax	Description
C#	private IFTPClientInterface FtpClient; FtpClient = Connect.CreateFTPClient(CommunicationGlobals.ConnectionTypes.SSH, null);
VB.NET	Private FtpClient As IFTPClientInterface = Nothing FtpClient = Connect.CreateFTPClient(CommunicationGlobals.ConnectionTypes.SSH, null)

Connect.CreateSocketClient Method

Creates an Socket Client object and returns as ITCPSocketCommunicationInterface.

Syntax

Syntax	Description
C#	**public static** ITCPSocketCommunicationInterface CreateSocketClient (ITCPSocketCommunicationInterface *prevObject*)
Visual Basic	**Public Shared Function** CreateSocketClient (_ *prevObject* **As** ITCPSocketCommunicationInterface_) **As** ITCPSocketCommunicationInterface
Visual C++	**public:** **static** ITCPSocketCommunicationInterface^ CreateSocketClient (ITCPSocketCommunicationInterface^ *prevObject*)

Parameters

Param.	Type	Description
prevObject	ODT.PowerPmacComLib. ITCPSocketCommunicationInterface	An Socket Client object that was previously created. It is returned for reuse if it is still valid and connected. If an object does not exist, pass null as the parameter.

Return Value

Interface to GetErrors event object. ITCPSocketCommunicationInterface

Examples

Syntax	Description
C#	private ITCPSocketCommunicationInterface SocketClient; SocketClient = Connect. CreateSocketClient (null);
VB.NET	Private SocketClient As ITCPSocketCommunicationInterface = Nothing SocketClient = Connect. CreateSocketClient (null)

Connect.CreateGetErrorsEvent Method

Syntax

Syntax	Description
C#	**public static** IGetErrorEventInterface CreateGetErrorsEvent (CommunicationGlobals.ConnectionTypes type, IGetErrorsEventInterface prevObject)
Visual Basic	**Public Shared Function** CreateGetErrorsEvent (_ *type* **As** CommunicationGlobals.ConnectionTypes, _ *prevObject* **As** IGetErrorsEventInterface _) As IGetErrorsEventInterface
Visual C++	**public:** **static** IGetErrorsEventInterface^ CreateGetErrorsEvent (CommunicationGlobals.ConnectionTypes *type*, IGetErrorsEventInterface^ *prevObject*)

Parameters

Param.	Type	Description
type	ODT.PowerPmacComLib.CommunicationGlobals.ConnectionTypes	Connection: SSH or PPmacServer
prevObject	ODT.PowerPmacComLib.IGetErrorsEventInterface	A GetErrors object that was previously created. It is returned for reuse if it is still valid and connected. If an object does not exist, pass null as the parameter.

Return Value

Interface to the GetErrors event object. IGetErrorsEventInterface

Examples

Syntax	Description
C#	private IGetErrorsEventInterface ErrorClient; ErrorClient = Connect.CreateGetErrorsEvent(CommunicationGlobals.ConnectionTypes.SSH, null);
VB.NET	Private ErrorClient As IGetErrorsEventInterface = Nothing ErrorClient = Connect.CreateGetErrorsEvent(CommunicationGlobals.ConnectionTypes.SSH, null)

Connect.CreateSyncGpascii Method

Creates a synchronous Gpascii communication object and returns as ISyncGpasciiCommunicationInterface.

Establish communication with the Power PMAC using SSH or PPmacServer.

Syntax

Syntax	Description
C#	**public static** ISyncGpasciiCommunicationInterface CreateSyncGpascii (CommunicationGlobals.ConnectionTypes *type*, ISyncGpasciiCommunicationInterface *prevObject*)
Visual Basic	**Public Shared Function** CreateSyncGpascii (_ *type* **As** CommunicationGlobals.ConnectionTypes, _ *prevObject* **As** ISyncGpasciiCommunicationInterface _

Syntax	Description
	) **As** ISyncGpasciiCommunicationInterface
Visual C++	**public:** **static** ISyncGpasciiCommunicationInterface^ CreateSyncGpascii (CommunicationGlobals.ConnectionTypes type, ISyncGpasciiCommunicationInterface^ prevObject)

Parameters

Param.	Type	Description
type	ODT.PowerPmacComLib.CommunicationGlobals.ConnectionTypes	Connection: SSH or PPmacServer
prevObject	ODT.PowerPmacComLib.ISyncGpasciiCommunicationInterface	A Gpascii object that was previously created. It is returned for reuse if it is still valid and connected. If an object does not exist, pass null as the parameter.

Return Value

Interface to synchronous Gpascii object. ISyncGpasciiCommunicationInterface

Examples

Syntax	Description
C#	private ISyncGpasciiCommunicationInterface SyncGPAscii; SyncGPAscii = Connect.CreateSyncGpascii(CommunicationGlobals.ConnectionTypes.SSH, null);
VB.NET	Private SyncGPAscii As ISyncGpasciiCommunicationInterface = Nothing SyncGPAscii = Connect.CreateSyncGpascii(CommunicationGlobals.ConnectionTypes.SSH, null)

Connect.CreateSyncTerminal Method

Creates a synchronous Terminal communication object and returns as ISyncTerminalCommunicationInterface.

Establish communication with the Power PMAC operating system using SSH or PPmacServer.

Syntax

Syntax	Description
C#	**public static** ISyncTerminalCommunicationInterface CreateSyncTerminal (CommunicationGlobals.ConnectionTypes *type*, ISyncTerminalCommunicationInterface *prevObject*)
Visual Basic	**Public Shared Function** CreateSyncTerminal (_ *type* **As** CommunicationGlobals.ConnectionTypes, _ *prevObject* **As** ISyncTerminalCommunicationInterface _) As ISyncTerminalCommunicationInterface
Visual C++	**public:** **static** ISyncTerminalCommunicationInterface^ CreateSyncTerminal (CommunicationGlobals.ConnectionTypes *type*, ISyncTerminalCommunicationInterface^ *prevObject*)

Parameters

Param.	Type	Description
type	ODT.PowerPmacComLib.CommunicationGlobals.ConnectionTypes	Connection: SSH or PPmacServer
prevObject	ODT.PowerPmacComLib.ISyncTerminalCommunicationInterface	A synchronous Terminal object that was previously created. It is returned for reuse if it is still valid and connected. If an object does not exist, pass null as the parameter.

Return Value

Interface to synchronous operating system Terminal object.
ISyncTerminalCommunicationInterface

Examples

Syntax	Description
C#	private ISyncTerminalCommunicationInterface OSSyncTerminal; OSSyncTerminal = Connect.CreateSyncTerminal(CommunicationGlobals.ConnectionTypes.SSH, null);
VB.NET	Private OSSyncTerminal As ISyncTerminalCommunicationInterface = Nothing

Syntax	Description
	OSSyncTerminal = Connect.CreateSyncTerminal(CommunicationGlobals.ConnectionTypes.SSH, null)

Connect.CreateUnsolicitedEvent Method

Creates an Unsolicited event object and returns the IUnsolicitedEvent interface.

Syntax

Syntax	Description
C#	**public static** IUnsolicitedEventInterface CreateUnsolicitedEvent (CommunicationGlobals.ConnectionTypes *type*, IUnsolicitedEventInterface *prevObject*)
Visual Basic	**Public Shared Function** CreateSyncGpascii (_ *type* **As** CommunicationGlobals.ConnectionTypes, _ *prevObject* **As** IUnsolicitedEventInterface _) **As** IUnsolicitedEventInterface
Visual C++	**public:** **static** IUnsolicitedEventInterface^ CreateUnsolicitedEvent (CommunicationGlobals.ConnectionTypes *type*, IUnsolicitedEventInterface^ *prevObject*)

Parameters

Param.	Type	Description
type	ODT.PowerPmacComLib.CommunicationGlobals.ConnectionTypes	Connection: SSH or PPmacServer
prevObject	ODT.PowerPmacComLib.ISyncTerminalCommunicationInterface	A synchronous Terminal object that was previously created. It is returned for reuse if it is still valid and connected. If an object does not exist, pass null as the parameter.

Return Value

Interface to unsolicited event object. IUnsolicitedEventInterface

deviceProperties

This class holds various properties for communication with Power PMAC.

Syntax

Syntax	Description
C#	**public class** deviceProperties
Visual Basic	**Public Class** deviceProperties
Visual C++	**public ref class** deviceProperties

Inheritance Hierarchy in the .NET Framework

System.Object
    ODT.PowerPmacComLib.deviceProperties

deviceProperties Members

The deviceProperties type exposes the members explained below.

Constructors

Name	Description
deviceProperties	Initializes a new instance of the deviceProperties class.

Methods

Name	Description
SetProperties	Sets the device properties.
SetProperties	Sets the device properties.

Properties

Name	Description
Gateway	Gateway of the network that includes the Power PMAC. Your network administrator can provide more information.
IPAddress	IPv4–based network address of the Power PMAC. The default address is 192.168.0.200.
Password	The password to log in to the Power PMAC. This property is read-only. The default password is *deltatau*.
PortNumber	The number of the network port of the Power PMAC. The default port number is 22.
Protocol	Communication protocol of the Power PMAC. The default protocol is SSH. See CommunicationGlobals.ConnectionTypes for more information.
SubnetMask	The subnet mask of the network that includes the Power PMAC. Your network administrator can provide more information.
User	The user name to log in to the Power PMAC. This property is read-only. The default user name is *root*.

deviceProperties Constructor

Initializes a new instance of the deviceProperties class.

Syntax

Syntax	Description
C#	**public** deviceProperties()
Visual Basic	**Public Sub** New
Visual C++	**public:** deviceProperties()

deviceProperties Methods

Name	Description
SetProperties	Sets the device properties.
SetProperties	Sets the device properties.

deviceProperties Properties

Name	Description
Gateway	Gateway of the network that includes the Power PMAC. Your network administrator can provide more information.
IPAddress	IPv4–based network address of the Power PMAC. The default address is 192.168.0.200.
Password	The password to log in to the Power PMAC. This property is read-only. The default password is *deltatau*.
PortNumber	The number of the network port of the Power PMAC. The default port number is 22.
Protocol	Communication protocol of the Power PMAC. The default protocol is SSH. See CommunicationGlobals.ConnectionTypes for more information.
SubnetMask	The subnet mask of the network that includes the Power PMAC. Your network administrator can provide more information.
User	The user name to log in to the Power PMAC. This property is read-only. The default user name is *root*.

deviceProperties.SetProperties Method

Sets the device properties.

Syntax

Syntax	Description
C#	**public void SetProperties{** deviceProperties properties **}**

Visual Basic	**Public Function** SetProperties {
	type **As** deviceProperties properties
	}

Visual C++	**public:**
	void SetProperties {
	deviceProperties *properties*
	}

Parameters

properties
Type: deviceProperties
A deviceProperties class.

deviceProperties.SetProperties Method

Sets the device properties.

Syntax

Syntax	Description
C#	**public void SetProperties{** DeviceConnectionSettings *connectionSettings* **}**
Visual Basic	**Public Function** SetProperties { _ type **As** DeviceConnectionSettings *connectionSettings* }

Visual C++	**public:**
	void SetProperties {
	DeviceConnectionSettings *connectionSettings*
	}

Parameters

connectionSettings
Type: DeviceConnectionSettings
A DeviceConnectionSettings class.

Return Value

VOID

Examples

Syntax	Description
C#	private deviceProperties AdeviceProperties; deviceProperties.SetProperties(MyDeviceConnectionSettings);
VB.NET	Private AdeviceProperties As deviceProperties deviceProperties.SetProperties (MyDeviceConnectionSettings)

deviceProperties.Gateway Property

Gateway of the network that includes the Power PMAC. Your network administrator can provide more information.

Syntax

Syntax	Description
C#	**public** string Gateway { **get**; **set**; }
Visual Basic	**Public Property** Gateway **As** String Get Set
Visual C++	**public:** **property** String^ Gateway { String^ **get** ();

Syntax	Description
	void set (String^ *value*); }

deviceProperties.IPAddress Property

IPv4–based network address of the Power PMAC. The default address is 192.168.0.200.

Syntax

Syntax	Description
C#	**public** string IPAddress { **get**; **set**; }
Visual Basic	**Public Property** IPAddress **As** String **Get** **Set**

Visual C++	**public:**
	property String^ IPAddress {
	String^ **get** ();
	void set (String^ *value*);
	}

deviceProperties.Password Property

The password to log in to the Power PMAC. This property is read-only. The default password is *deltatau*.

Syntax

Syntax	Description
C#	**public** string Password { **get**; **set**; }
Visual Basic	**Public Property** Password **As** String **Get** **Set**

Visual C++	**public:**
	property String^ Password {
	String^ **get** ();
	void set (String^ *value*);
	}

deviceProperties.PortNumber Property

The number of the network port of the Power PMAC. The default port number is 22.

Syntax

Syntax	Description
C#	**public** int PortNumber { **get**; **set**; }
Visual Basic	**Public Property** PortNumber **As** Integer **Get** **Set**

Visual C++	**public:**
	property int Portnumber {
	int **get** ();
	void set (int *value*);
	}

deviceProperties.Protocol Property

Communication protocol of the Power PMAC. The default protocol is SSH. See CommunicationGlobals.ConnectionTypes for more information.

Syntax

Syntax	Description
C#	**public** CommunicationGlobals.ConnectionTypes Protocol { **get**; **set**; }
Visual Basic	**Public Property** Protocol **As** CommunicationGlobals.ConnectionTypes **Get** **Set**

Visual C++	**public:**
	property CommunicationGlobals.ConnectionTypes Protocol {
	CommunicationGlobals.ConnectionTypes get ();
	void set (CommunicationGlobals.ConnectionTypes *value*);
	}

deviceProperties.SubnetMask Property

The subnet mask of the network that includes the Power PMAC. Your network administrator can provide more information.

Syntax	Description
C#	**public** string SubnetMask { **get**; **set**; }
Visual Basic	**Public Property** SubnetMask **As** String **Get** **Set**

Visual C++	**public:**
	property String^ SubnetMask {
	String^ **get** ();
	void set (String^ *value*);
	}

deviceProperties.User Property

The user name to log in to the Power PMAC. This property is read-only. The default user name is *root*.

Syntax

Syntax	Description
C#	**public** string User { **get**; **set**; }
Visual Basic	**Public Property** User **As** String **Get** **Set**

Visual C++	**public:**
	property String^ User {
	String^ **get** ();
	void set (String^ *value*);
	}

DevicePropertyPage

A Forms class of Microsoft® Windows® that enables you to view and edit communication properties of Power PMAC.

On this dialog you can save and display the connection properties of ten or fewer Power PMAC devices. You can switch between devices and test the connections.

If a Power PMAC is not accessible, this class uses RouteCommand to add a device.

Syntax

Syntax	Description
C#	**public class** devicePropertyPage : Form
Visual Basic	**Public Class** DevicePropertyPage _ **Inherits** Form
Visual C++	**public ref class** DevicePropertyPage **: public** Form

Inheritance Hierarchy in the .NET Framework

System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      System.Windows.Forms.Control
        System.Windows.Forms.ScrollableControl
          System.Windows.Forms.ContainerControl
            System.Windows.Forms.Form
              ODT.PowerPmacComLib.DevicePropertyPage

DevicePropertyPage Members

The DevicePropertyPage type exposes the members explained below.

Constructors

Name	Description
DevicePropertyPage(deviceProperties currDeviceProp)	Initializes a new instance of the DevicePropertyPage class.
DevicePropertyPage(deviceProperties currDeviceProp, bool IsConnected)	Initializes a new instance of the DevicePropertyPage class.

Events

Name	Description
OnClose	Occurs when the DevicePropertyPage form closes.
OnConnect	Occurs when the Power PMAC is available on the network and is ready for connection.
OnDisConnect	Occurs when the connection to the Power PMAC closes.

DevicePropertyPage Constructor

Overload List

Name	Description
DevicePropertyPage(deviceProperties)	Initializes a new instance of the DevicePropertyPage class.
DevicePropertyPage(deviceProperties, Boolean)	Initializes a new instance of the DevicePropertyPage class.

DevicePropertyPage Constructor (deviceProperties)

Initializes a new instance of the DevicePropertyPage class.

Syntax

Syntax	Description
C#	**public** DevicePropertyPage (deviceProperties *currDeviceProp*)
Visual Basic	**Public Sub** New (_ *currDeviceProp* **As** deviceProperties _)
Visual C++	**public**: DevicePropertyPage(deviceProperties^ *currDeviceProp*)

Parameters

currDeviceProp
Type: ODT.PowerPmacComLibdeviceProperties
deviceProperties of the Power PMAC.

DevicePropertyPage Constructor (deviceProperties, Boolean)

Initializes a new instance of the DevicePropertyPage class.

Syntax

Syntax	Description
C#	**public** DevicePropertyPage(deviceProperties *currDeviceProp*, bool IsConnected)
Visual Basic	**Public Sub** New (_ currDeviceProp **As** deviceProperties, _ IsConnected **As** Boolean _)
Visual C++	**public**: DevicePropertyPage(deviceProperties^ *currDeviceProp*, bool IsConnected)

Parameters

Name	Type	Description
currDeviceProp	ODT.PowerPmacComLib.deviceProperties	deviceProperties of the Power PMAC.
IsConnected	System.Boolean in the .NET Framework	Connection status of the Power PMAC.

ErrorsMaskArgs

A GetErrors argument. This class wraps Power PMAC error messages sent by the GetErrors process.

Syntax

Syntax	Description
C#	**public class** ErrorsMaskArgs : EventArgs
Visual Basic	**Public Class** ErrorsMaskArgs _ Inherits EventArgs
Visual C++	**public ref class** ErrorsMaskArgs : **public** EventArgs

Inheritance Hierarchy

System.Object
  System.EventArgs
    ODT.PowerPmacComLib.ErrorsMaskArgs

ErrorsMaskArgs Members

The ErrorsMaskArgs type exposes the members explained below.

Constructor

Name	Description
ErrorsMaskArgs	ErrorsMastArgs constructor.

Properties

Name	Description
ErrorMask	Error mask as UInt64 enumeration. This property is read-only.
ErrorType	Error type as char. This property is read-only.

ErrorsMaskArgs Constructor

Syntax

Syntax	Description
C#	**public** ErrorsMaskArgs(char error_type, ulong error_mask)

Syntax	Description
Visual Basic	**Public Sub** New (_ error_type **As** Char, _ error_mask **As** ULong _)
Visual C++	**public**: ErrorsMaskArgs (wchar_t error_type, unsigned long long *error_mask*)

Parameters

Param.	Type	Description
error_type	System.Char in the .NET Framework	Error type as character.
error_mask	System.UInt64 in the .NET Framework	Error mask, as UInt64.

ErrorMaskArgs Properties

The ErrorsMaskArgs type exposes the members explained in the following table.

Name	Description
ErrorMask	Error mask as UInt64 enumeration. This property is read-only.
ErrorType	Error type as char. This property is read-only.

ErrorsMaskArgs.ErrorMask Property

Error mask as UInt64 enumeration. This property is read-only.

Syntax

Syntax	Description
C#	**public** ulong ErrorMask { **get**; }
Visual Basic	**Public ReadOnly Property** ErrorMask **As** ULong **Get**
Visual C++	**public**: **property** unsigned long long ErrorMask (unsigned long long **get** (); }

ErrorsMaskArgs.ErrorType Property

Syntax

Syntax	Description
C#	**public** char ErrorType { **get**; }
Visual Basic	**Public ReadOnly Property** ErrorType **As** Char **Get**

Visual C++	**public**:
	property wchar_t Errortype {
	wchar_t **get** ();
	}

FirmwareUpdater

This class provides the ability to update the firmware of the Power PMAC.

Syntax

Syntax	Description
C#	**public class** FirmwareUpdater
Visual Basic	**Public class** FirmwareUpdater
Visual C++	**public ref class** FirmwareUpdater

Hierarchy in the .NET Framework

System.Object
    FirmwareUpdater

FirmwareUpdater Members

Constructors

Name	Description
FirmwareUpdater	Constructor for FirmwareUpdater class.

Methods

Name	Description
UpdateFirmware	Downloads the selected firmware and restarts the device to update the firmware. Callee must confirm the firmware version after the machine restarts, and must verify that the firmware successfully updated. Returns false or an exception when a failure or error occurs before reaching the restart state.

FirmwareUpdater Constructors

Syntax

Name	Description
FirmwareUpdater	Initializes a new instance of the FirmwareUpdater class.

FirmwareUpdater Methods

UpdateFirmware Method

Downloads the selected firmware and restarts the device to update the firmware. Callee must confirm the firmware version after the machine restarts, and must verify that the firmware successfully updated. Returns false or an exception when a failure or error occurs before reaching the restart state.

Syntax	Description
C#	**public** bool UpdateFirmware(string *FirmwareFilePath*, string *DeviceIP*, string *DeviceUsername*,

Syntax	Description
	string *DevicePassword*, int *port*)
Visual Basic	**Public** Bunction UpdateFirmware (_ *FirmwareFilePath* **As** String, _ *DeviceIP* **As** String, _ *Deviceusername* **As** String, _ *DevicePassword* **As** String, _ *port* **As** Int _) **As** bool
Visual C++	**public:** bool UpdateFirmware (string *FirmwareFilePath*, string *DeviceIP*, string *DeviceUsername*, string *DevicePassword*, int *port*)

Parameters

Param.	Type	Description
FirmwareFilePath	string	Path to the file that contains the new firmware.
DeviceIP	string	IP address of the device to be updated.
DeviceUsername	string	User name to log in to the device to be updated.
DevicePassword	string	Password to log in to the device to be updated.
port	integer	SSH port for the device to be updated.

Return Value

True when the firmware file is copied and the device restarts. Returns an exception or false when failure occurs.

Examples

Syntax	Description
C#	private FirmwareUpdater firmwareUpdater; firmwareUpdater.UpdateFirmware(“FirmwarePath”,“192.168.0.200”,“root”,Password,22);
VB.NET	Private firmwareUpdater As FirmwareUpdater firmwareUpdater.UpdateFirmware(“FirmwarePath”,”192.168.0.200”,“root”,Password,22)

GetErrorsArgs

A GetErrors argument. This class wraps Power PMAC error messages sent by the GetErrors process.

Syntax

Syntax	Description
C#	**public class** GetErrorsArgs : EventArgs
Visual Basic	**Public Class** GetErrorsArgs _ **Inherits** EventArgs
Visual C++	**public ref class** GetErrorsArgs : **public** EventArgs

Inheritance Hierarchy in the .NET Framework

System.Object
    System.EventArgs
        ODT.PowerPmacComLib.GetErrorsArgs

GetErrorsArgs Members

Constructor

Name	Description
GetErrorsArgs	UnsolicitedArg constructor.

Properties

Name	Description
ErrorMessage	Error message as string. This property is read-only.
StatusType	Status type of error as StatusType enumeration. This property is read-only.

GetErrorsArgs Constructor

Syntax

Syntax	Description
C#	**public** GetErrorsArgs(string errormessage, StatusType *errortype*)
Visual Basic	**Public Sub** New (_ *errormessage* **As** String, _ *errortype* **As** StatusType _)
Visual C++	**public**: GetErrorsArgs(String^ errormessage, StatusType *errortype*)

Parameters

Name	Type	Description
errormessage	System.String in the .NET Framework	Error message as string.

Name	Type	Description
errortype	ODT.PowerPmacComLib.StatusType	Status type of error as StatusType enumeration.

GetErrorsArgs Properties

The GetErrorsArgs type exposes the members explained in the following table.

Properties

Name	Description
ErrorMessage	Error message as string. This property is read-only.
StatusType	Status type of error as StatusType enumeration. This property is read-only.

GetErrorsArgs.ErrorMessage Property

Error message as string. This property is read-only.

Syntax

Syntax	Description
C#	**public** string ErrorMessage { **get**; }
Visual Basic	**Public ReadOnly Property** ErrorMessage **As** String **Get**

Visual C++	**public**:
	property String^ *errormessage* {
	String^ **get** ();
	}

GetErrorsArgs.StatusType Property

Syntax

Status type of error as StatusType enumeration. This property is read-only.

Syntax	Description
C#	**public** StatusType StatusType { **get**; }
Visual Basic	**Public ReadOnly Property** StatusType **As** StatusType **Get**

Visual C++	**public**:
	property StatusType StatusType {
	StatusType **get** ();
	}

LogArgs

This class wraps any general, exception or error message generated by this library

Syntax

Syntax	Description
C#	**public class** LogArgs : EventArgs

Syntax	Description
Visual Basic	**Public Class** LogArgs _ **Inherits** EventArgs
Visual C++	**public ref class** LogArgs : **public** EventArgs

Inheritance Hierarchy in the .NET Framework

System.Object
System.EventArgs
ODT.PowerPmacComLib.GetErrorsArgs

LogArgs Members

Constructor

Name	Description
LogArgs	LogArg constructor.

Properties

Name	Description
LogMessage	Logs message as a string. This property is read-only.

LogArgs Constructors

Syntax

Syntax	Description
C#	**public** LogArgs(string logmessage)
Visual Basic	**Public Sub** New (_ *logmessage* **As** String)
Visual C++	**public**: GetErrorsArgs(String^ logmessage)

LogArgs Properties

The LogArgs type exposes the members explained in the following table.

Properties

Name	Description
LogMessage	Logs message as a string. This property is read-only.

LogArgs.LogMessage Property

Log message as string. This property is read-only.

Syntax

Syntax	Description
C#	**public** string LogMessage { **get**; }

Syntax	Description
Visual Basic	**Public ReadOnly Property** LogMessage **As** String **Get**

Visual C++	**public**:
	property String^ *LogMessage* {
	String^ **get** ();
	}

RouteCommand

This class adds the Power PMAC subnet route to the subnet of the host computer. RouteCommand pings the host computer. If the ping succeeds, the route is permanently added to the host. If the ping fails, RouteCommand clears the route.

Syntax

Syntax	Description
C#	**public class** RouteCommand
Visual Basic	**Public Class** RouteCommand
Visual C++	**public ref class** RouteCommand

Inheritance Hierarchy in the .NET Framework

System.Object
ODT.PowerPmacComLib.RouteCommand

RouteCommand Members

The RouteCommand type exposes the members explained below.

Constructors

Name	Description
RouteCommand	Initializes a new instance of the RouteCommand class.

Methods

Name	Description
doPing(String)	Ping the specified IP address of the Power PMAC. The Power PMAC will return True if the ping is successful, and False if the ping is unsuccessful.
doPing(String, Int32)	Ping the specified IP address of the Power PMAC. The Power PMAC will return True if the ping is successful, and False if the ping is unsuccessful.

RouteCommand Constructor

Initializes a new instance of the RouteCommand class.

Syntax

Syntax	Description
C#	**public** RouteCommand()

Syntax	Description
Visual Basic	**Public sub** New
Visual C++	**public:** RouteCommand()

RouteCommand Methods

The RouteCommand type exposes the methods explained in the following table.

Name	Description
doPing(String)	Ping the specified IP address of the Power PMAC. The Power PMAC will return True if the ping is successful, and False if the ping is unsuccessful.
doPing(String, Int32)	Ping the specified IP address of the Power PMAC. The Power PMAC will return True if the ping is successful, and False if the ping is unsuccessful.

RouteCommand.doPing Method

Overload List

Name	Description
doPing(String)	Ping the specified IP address of the Power PMAC. The Power PMAC will return True if the ping is successful, and False if the ping is unsuccessful.
doPing(String, Int32)	Ping the specified IP address of the Power PMAC. The Power PMAC will return True if the ping is successful, and False if the ping is unsuccessful.

RouteCommand.doPing Method (String)

Ping the specified IP address of the Power PMAC. The Power PMAC will return True if the ping is successful, and False if the ping is unsuccessful.

Syntax

Syntax	Description
C#	**public** bool doPing(string *IpAddress*)
Visual Basic	**Public Function** doping (_ *IpAddress* **As** String _) **As** Boolean
Visual C++	**public:** bool doPing(String^ *IpAddress*

Syntax	Description
	)

Parameters

Parameter	Type	Description
IpAddress	System.String in the .NET Framework	IPv4–based network address of the Power PMAC. The default address is 192.168.0.200.

Return Value

True or False, based on the status of the ping.

RouteCommand.doPing Method (String, Int32)

Ping the specified IP address of the Power PMAC. The Power PMAC will return True if the ping is successful, and False if the ping is unsuccessful.

Syntax

Syntax	Description
C#	**public** bool doPing(string *IpAddress,* int *timeout*)
Visual Basic	**Public Function** doPing (_ *IpAddress* **As** String, _ timeout **As** Integer _) **As** Boolean
Visual C++	**public:** bool doPing(String^ *IpAddress,* int *timeout*)

Parameters

Parameter	Type	Description
IpAddress	System.String in the .NET Framework	IPv4–based network address of the Power PMAC. The default address is 192.168.0.200.

Return Value

True or False, based on the status of the ping.

RouteCommand.routeCommand Method

Execute the Route command, and add the IP address to the routing table of the host machine.

Syntax

Syntax	Description
C#	**public** bool routeCommand(string *IpAddress*)
Visual Basic	**Public Function** routeCommand (_ *IpAddress* **As** String _) **As** Boolean
Visual C++	**public:** bool routeCommand(String^ *IpAddress*)

Parameters

Parameter	Type	Description
IpAddress	System.String in the .NET Framework	IPv4–based network address of the Power PMAC. The default address is 192.168.0.200.

Return Value

True or False, based on the status of the ping.

SFTPArgs

SFTP arguments. This is a custom class derived from the EventArgs class. SFTPArgs wraps a standard FTP communication response message.

Syntax

Syntax	Description
C#	**public class** SFTPArgs : EventArgs
Visual Basic	**Public Class** SFTPArgs _ Inherits EventArgs
Visual C++	**public ref class** SFTPArgs : **public** EventArgs

Inheritance Hierarchy in the .NET Framework

System.Object
    System.EventArgs
        ODT.PowerPmacComLib.SFTPArgs

SFTPArgs Members

The SFTPArgs type exposes the members explained below.

Constructors

Name	Description
SFTPArgs	Initializes a new stance of the SFTPArgs class.

Properties

Name	Description
DST	Address of the target FTP machine (the Power PMAC). This property is read-only.
Message	Status of an FTP communication message. This property is read-only.
PercentComplete	The percentage of the total data that has successfully transferred. This property is read-only.
SRC	Address of the source FTP machine (the client computer). This property is read-only.
TOTALBYTES	Size of a file in bytes. This property is read-only.
XFERBYTES	Number of bytes that have transferred. This property is read-only.

SFTPArgs Constructor

Initializes a new stance of the SFTPArgs class.

Syntax

Syntax	Description
C#	**public** SFTPArgs(string *src*, string *dst*, int *transferredBytes*, int *totalBytes*, string *message*, int *percentComplete*)
Visual Basic	**Public Sub** New (_ *src* **As** String, _ *dst* **As** String, _ *transferredBytes* **As** Integer, _ *totalBytes* **As** Integer, _ *message* **As** String, _ *percentComplete* **As** Integer _)
Visual C++	**public:** String^ *src*, String^ *dst*, int *transferredBytes*, int *totalBytes*, String^ *message*, int *percentComplete*)

Parameters

Parameter	Type	Description
sSrc	System.String in the .NET Framework	Address of the source FTP machine (the client computer).
dst	System.String in the .NET Framework	Address of the target FTP machine (the Power PMAC).

Parameter	Type	Description
transferredBytes	System.Int.32 in the .NET Framework	The number of bytes transferred.
totalBytes	System.Int32 in the .NET Framework	The size of a file in bytes.
Message	System.String in the .NET Framework	The status of an FTP communication message.
percentComplete	System.Int32 in the .NET Framework	The percentage of the total data that has successfully transferred.

SFTPArgs Properties

The SFTPArgs type exposes the members explained in the following table.

Name	Description
DST	Address of the target FTP machine (the Power PMAC). This property is read-only.
Message	Status of an FTP communication message. This property is read-only.
PercentComplete	The percentage of the total data that has successfully transferred. This property is read-only.
SRC	Address of the source FTP machine (the client computer). This property is read-only.
TOTALBYTES	Size of a file in bytes. This property is read-only.
XFERBYTES	The number of bytes that have transferred. This property is read-only.

SFTPArgs/DST Property

Address of the target FTP machine (the Power PMAC). This property is read-only.

Syntax

Syntax	Description
C#	**public** string DST { **get**; }
Visual Basic	**Public ReadOnly Property** DST **As** String **Get**

Visual C++	**public:**
	property String^ DST {
	String^ **get** ();
	}

SFTPArgs.Message Property

Status of an FTP communication message. This property is read-only.

Syntax

Syntax	Description
C#	**public** string Message { **get**; }
Visual Basic	**Public ReadOnly Property** Message **As** String **Get**

Visual C++	**public:**
	property String^ Message {
	String^ **get** ();
	}

SFTPArgs.PercentComplete Property

The percentage of the total data that has successfully transferred. This property is read-only.

Syntax

Syntax	Description
C#	**public** int PercentComplete { **get**; }
Visual Basic	**Public ReadOnly Property** PercentComplete **As** Integer **Get**

Visual C++	**public:**
	property int PercentComplete {
	int **get** ();
	}

SFTPArgs.SRC Property

Address of the source FTP machine (the client computer). This property is read-only.

Syntax

Syntax	Description
C#	**public** string SRC { **get**; }
Visual Basic	**Public ReadOnly Property** SRC **As** String **Get**

Visual C++	**public:**
	property String^ SRC {
	String^ **get** ();
	}

SFTPArgs.TOTALBYTES Property

Size of a file in bytes. This property is read-only.

Syntax

Syntax	Description
C#	**public** int TOTALBYTES { **get**; }
Visual Basic	**Public ReadOnly Property** TOTALBYTES **As** Integer **Get**

Visual C++	**public:**
	property int TOTALBYTES {
	int **get** ();
	}

SFTPArgs.XFERBYTES Property

The number of bytes that have transferred. This property is read-only.

Syntax

Syntax	Description
C#	**public** int XFERBYTES { **get**; }
Visual Basic	**Public ReadOnly Property** XFERBYTES **As** Integer **Get**

Syntax	Description

Visual C++	**public:**
	property int XFERBYTES {
	int **get** ();
	}

SFTPLogArgs

SFTP Log arguments. This is a custom class derived from the EventArgs class. SFTPLogArgs wraps a response message from the Power PMAC.

Syntax

Syntax	Description
C#	**public class** SFTPLogArgs : EventArgs
Visual Basic	**Public Class** SFTPLogArgs _ **Inherits** EventArgs
Visual C++	**public ref class** SFTPLogArgs : **public** EventArgs

Inheritance Hierarchy in the .NET Framework

System.Object
    System.EventArgs
        ODT.PowerPmacComLib.SFTPLogArgs

SFTPLogArgs Members

The SFTPLogArgs type exposes the members explained below.

Constructor

Name	Description
SFTPLogArgs	Response data from the Power PMAC command.

Properties

Name	Description
Response	Response data from the Power PMAC command. This property is read-only.

SFTPLogArgs Constructor

Syntax

Syntax	Description
C#	**public** SFTPLogArgs(string *responseE*)
Visual Basic	**Public Sub** New (_ *responseE* **As** String _)

Syntax	Description
Visual C++	**Public:** SFTPLogArgs(String^ *responseE*)

Parameters

responseE
Type: System.String in the .NET Framework

SFTPLogArgs Properties

The SFTPLogArgs type exposes the members explained in the following table.

Name	Description
Response	Response data from the Power PMAC command. This property is read-only.

SFTPLogArgs.Response Property

Response data from the Power PMAC command. This property is read-only.

Syntax

Syntax	Description
C#	**public** string Response { **get**; }
Visual Basic	**Public ReadOnly Property** Response **As** String (_ **Get**

Visual C++	**public:**
	property String^ Response {
	String^ **get** ();
	}

UnsolicitedArgs

Unsolicited response argument. UnsolicitedArgs wraps unsolicited messages sent by the Power PMAC.

Syntax

Syntax	Description
C#	**public class** UnsolicitedArgs : EventArgs
Visual Basic	**Public Class** Unsolicited Args _ **Inherits** EventArgs
Visual C++	**Public ref class** UnsolicitedArgs : **public** EventArgs

Inheritance Hierarchy in the .NET Framework

System.Object
    System.EventArgs
        ODT.PowerPmacComLib.UnsolicitedArgs

UnsolicitedArgs Members

The UnsolicitedArgs type exposes the members explained below.

Constructor

Name	Description
UnsolicitedArgs	UnsolicitedArgs constructor

Properties

Name	Description
Status	Returns the status of an unsolicited message port.
UnsolicitedMessage	Unsolicited message as string. This property is read-only.
UnsolicitePort	Unsolicited port number as integer. This property is read-only.

UnsolicitedArgs Constructor

Syntax

Syntax	Description
C#	**public** UnsoliciteArgs(string *unsolicitemessage*, int *Port*, UnsolicitedPortStatus *st*)
Visual Basic	**Public Sub** New (_ *unsolicitemessage* **As** String, _ *Port* **As** Integer, _ *st* **As** UnsolicitedPortStatus _)
Visual C++	**public:** UnsolicitedArgs(String^ *unsolicitemessage*, int *Port*, UnsolicitedPortStatus *st*)

Parameters

Param.	Type	Description
unsolicitemessage	System.String in the .NET Framework	Unsolicited message as string.
Port	System.Int32 in the .NET Framework	Port number or buffer number where a message will be available as an integer.
st	ODT.PowerPmacComLib.UnsolicitedPortStatus	Returns the status word of the port.

UnsolicitedArgs Properties

The UnsolicitedArgs type exposes the members explained in the following table.

Name	Description
Status	Returns the status of an unsolicited message port.
UnsolicitedMessage	Unsolicited message as string. This property is read-only.
UnsolicitePort	Unsolicited port number as integer. This property is read-only.

UnsolicitedArgs.Status Property

Returns the status of an unsolicited message port.

Syntax

Syntax	Description
C#	**public** UnsolicitedPortStatus Status { **get**; }
Visual Basic	**Public ReadOnly Property** Status **As** UnsolicitedPortStatus **Get**

Visual C++	**public:**
	property UnsolicitedPortStatus Status {
	UnsolicitedPortStatus get ();
	}

UnsolicitedArgs.UnsoliciteMessage Property

Unsolicited message as string. This property is read-only.

Syntax

Syntax	Description
C#	**public** string UnsoliciteMessage { **get**; }
Visual Basic	**Public ReadOnly Property** UnsoliciteMessage **As** String **Get**

Visual C++	**public:**
	property String^ UnsoliciteMessage {
	String^ get ();
	}

UnsolicitedArgs.UnsolicitePort Property

Unsolicited port number as integer. This property is read-only.

Syntax

Syntax	Description
C#	**public** int UnsolicitePort { **get**; }
Visual Basic	**Public ReadOnly Property** UnsolicitePort **As** Integer **Get**

Visual C++	**public:**
	property int UnsolicitePort {
	int get ();
	}

UnsolicitedCompleteStatus

Represents the class for Unsolicited port status including port busy and status string

Syntax

Syntax	Description
C#	**public class** UnsolicitedCompleteStatus
Visual Basic	**Public Class** UnsolicitedCompleteStatus
Visual C++	**public ref class** UnsolicitedCompleteStatus

Inheritance Hierarchy in the .NET Hierarchy

System.Object
    ODT.PowerPmacComLib.UnsolicitedCompleteStatus

UnsolicitedCompleteStatus Members

The UnsolicitedCompleteStatus type exposes the members explained below.

Constructors

Name	Description
UnsolicitedCompleteStatus	Initializes a new instance of the UnsolicitedCompleteStatus class.

Fields

Name	Description
portBusy	portBusy Boolean flag is set if the port is busy and can not send or receive any more messages
status	status is exposed to public for reference in all targets
statusmessage	statusmessage contains the actual status string in case there is any error
version	We use current version of firmware to determine its compatibility with latest IDE environment

UnsolicitedCompleteStatus Constructor

Initializes a new instance of the UnsolicitedCompleteStatus class.

Syntax

Syntax	Description
C#	**public class** UnsolicitedCompleteStatus()
Visual Basic	**Public Sub** New
Visual C++	**public:** UnsolicitedCompleteStatus()

UnsolicitedCompleteStatus Fields

The UnsolicitedCompleteStatus type exposes the members explained in the following table.

Name	Description
portBusy	portBusy Boolean flag is set if the port is busy and can not send or receive any more messages
status	status is exposed to public for reference in all targets
statusmessage	statusmessage contains the actual status string in case there is any error
version	We use current version of firmware to determine its compatibility with latest IDE environment

UnsolicitedCompleteStatus.portBusy Field

portBusy Boolean flag is set if the port is busy and can not send or receive any more messages

Syntax

Syntax	Description
C#	**public** bool[] portBusy
Visual Basic	**Public** portBusy **As** Boolean()
Visual C++	**public:** **array**<bool>^ portBusy

UnsolicitedCompleteStatus.status Field

status is exposed to public for reference in all targets

Syntax

Syntax	Description
C#	**public** UnsolicitedStatus status
Visual Basic	**Public** status **As** UnsolicitedStatus
Visual C++	**public:** UnsolicitedStatus status

UnsolicitedCompleteStatus.statusmessage Field

statusmessage contains the actual status string in case there is any error

Syntax

Syntax	Description
C#	**public** string statusmessage
Visual Basic	**Public** statusmessage **As** String
Visual C++	**public:** String^ statusmessage

UnsolicitedCompleteStatus.version Field

We use current version of firmware to determine its compatibility with latest IDE environment

Syntax

Syntax	Description
C#	**public** string version
Visual Basic	**Public** version **As** String

Syntax	Description
Visual C++	**public:** String^ version

Interfaces

IAsyncGpasciiCommunicationInterface

Provides an interface for asynchronous Gpascii communication to Power PMAC

Syntax

Syntax	Description
C#	**public interface** IAsyncGpasciiCommunicationInterface
Visual Basic	**Public Interface** IAsyncGpasciiCommunicationInterface
Visual C++	**public interface class** IAsyncGpasciiCommunicationInterface

iAsyncGpasciiCommunicationInterface Members

The IAsyncGpasciiCommunicationInterface type exposes the members explained below.

Methods

Name	Description
AsyncGetResponse(String)	Sends command to the Power PMAC over SSH Server
AsyncGetResponse(String, Int32)	Sends command to the Power PMAC over SSH Server
ConnectGPAscii	This function open the Telnet port and opens Gpascii port with -2 option for PMAC communication.
DisconnectGpascii	Disconnect Gpascii port and then close the Telnet Port.
GPAscii2Connect	Obsolete. Why included then?
GPAsciiConnect	Obsolete. Why included then?
PowerPMACReset	Reset Command $$$, $$$** handled differently.
PowerPMACSave	Issues SAVE command to Power PMAC

Properties

Name	Description
EchoOn	Boolean Value indicating if EchoMode is on.
GpAsciiConnected	Boolean Value whether GPAscii is connected.
IP_Address	IP Address of PMAC
SocketConnected	Boolean Value indicating if Socket is connected.

Events

Name	Description
AsyncDataAvailable	AsyncDataAvailable Event need to be consume by Client for receiving GetResponse Data.
ComERROR	Event handler when there is a Socket Error

Name	Description
Log	Log of Error Events

IAsyncGpasciiCommunicationInterface Methods

The IAsyncGpasciiCommunicationInterface type exposes the members explained in the following table.

Name	Description
AsyncGetResponse(String)	Sends command to the Power PMAC over SSH Server
AsyncGetResponse(String, Int32)	Sends command to the Power PMAC over SSH Server
ConnectGPAscii	This function open the Telnet port and opens Gpascii port with -2 option for PMAC communication.
DisconnectGpascii	Disconnect Gpascii port and then close the Telnet Port.
BeginConnectGPAscii	This asynchrounous function open the Telnet port and opens Gpascii port with -2 option for PMAC communication.
PowerPMACReset	Reset Command $$$, $$$** handled differently.
PowerPMACSave	Issues SAVE command to Power PMAC

IAsyncGpasciiCommunicationInterface.AsyncGetResponse Method

Overload List

Name	Description
AsyncGetResponse(String)	Gets Response from PMAC using a command
AsyncGetResponse(String, Int32)	Gets Response from PMAC using a command, and notifies if EchoMode is active.

IAsyncGpasciiCommunicationInterface.AsyncGetResponse Method (String)

Sends command to the Power PMAC over SSH Server.

Syntax

Syntax	Description
C#	Status AsyncGetResponse{ string *cmd*)
Visual Basic	**Function** AsyncGetResponse (_ *cmd* **As** String _) **As** Status
Visual C++	Status AsyncGetResponse(String^ *cmd*)

Parameters

name="cmd". Command to be send to Power PMAC

Return Value

VOID

IAsyncGpasciiComunicationInterface.AsyncGetResponse Method (String, Int˜ ˜)

Sends command to the Power PMAC over SSH Server.

Syntax

Syntax	Description
C#	Status AsyncGetResponse{ string *cmd,* int *echoMode*)
Visual Basic	**Function** AsyncGetResponse (_ *cmd* **As** String, _ *echoMode* **As** Integer _) **As** Status
Visual C++	Status AsyncGetResponse(String^ *cmd,* int *echoMode*)

Parameters

Parameter	Type	Description
cmd	System.String in the .NET Framework	Command to be send to Power PMAC
echoMode	System.Int32 in the .NET Framework	Enables EchoMode

Return Value

VOID

IAsyncGpasciiComunicationInterface.ConnectGPAscii Method

This function open the SSH port and opens Gpascii port with -2 option for PMAC communication.

Syntax

Syntax	Description
C#	bool ConnectGPAscii(string *ipaddr*, int *port*, string *userName*, string password)
Visual Basic	**Function** ConnectGPAscii (_ *ipaddr* **As** String, _ *port* **As** Integer, _ *username* **As** String, _ *password* **As** String _) **As** Boolean

Syntax	Description
Visual C++	Bool ConnectGPAscii(String^ *ipaddr*, int *port*, String^ *userName*, String^ *password*)

Parameters

Parameter	Type	Description
ipaddr	System.String in the .NET Framework	IP Address
port	System.Int32 in the .NET Framework	Port Number
userName	System.String in the .NET Framework	Username
password	System.String in the .NET Framework	Password

Return Value

True on successful GPASCII Connection

IAsyncGpasciiCommunicationInterface.DisconnectGpascii Method

Disconnect Gpascii port and then close the SSH Port.

Syntax

Syntax	Description
C#	bool DisconnectGPAscii()
Visual Basic	**Function** DisconnectGPAscii **As** Boolean
Visual C++	bool DisconnectGPAscii()

Return Value

True on Successful GPAscii Disconnect.

IAsyncGpasciiCommunicationInterface.BeginConnectGPAscii Method

This asynchronous function opens the SSH port and opens Gpascii port with -2 option for PMAC communication.

Syntax

Syntax	Description
C#	Task<bool> BeginConnectGPAscii(string ipaddr, Int˜ 2 port, string userName, string password)
Visual Basic	**Function** BeginConnectGPAscii () **As** Task(Of Boolean)
Visual C++	Task<bool> BeginConnectGPAscii (string ipaddr, Int˜ 2 port, string userName, string password)

Return Value

True on successful GPASCII Connection

IAsyncGpasciiCommunicationInterface.PowerPMACReset Method

Reset Command $$$,$$$** handled differently.

Syntax

Syntax	Description
C#	bool PowerPMACReset(ResetType *cmdType*)
Visual Basic	**Function** PowerPMACReset (_ *cmdType* **As** ResetType _) **As** Boolean
Visual C++	bool PowerPMACReset(ResetType *cmdType*)

Parameters

cmdType
Type: ODT.PowerPmacComLib.ResetType
Either $$$ or $$$***

Return Value

True on Successful Reset.

IAsyncGpasciiCommunicationInterface.PowerPMACSave Method

Issues SAVE command to Power PMAC.

Syntax

Syntax	Description
C#	bool PowerPMACSave()
Visual Basic	**Function** PowerPMACSave **As** Boolean
Visual C++	bool PowerPMACSave()

Return Value

True on Successful save.

IAsyncGpasciiCommunicationInterface Properties

The IAsyncGpasciiCommunicationInterface type exposes the members explained in the following table.

Name	Description
EchoOn	Boolean Value indicating if EchoMode is on.
GpAsciiConnected	Boolean Value whether GPAscii is connected.
IP_Address	IP Address of PMAC
SocketConnected	Boolean Value indicating if Socket is connected.

IASyncCommunicationInterface.EchoOn Property

Boolean Value whether EchoMode is active.

Syntax

Syntax	Description
C#	bool EchoOn (**get**; **set**;)

Syntax	Description
Visual Basic	**Property** EchoOn **As** Boolean **Get** **Set**

Visual C++	**property** bool EchoOn {
	bool **get** ();
	void set (bool value);
	}

IASyncGpasciiCommunicationInterface.GpAsciiConnected Property

Boolean Value indicating whether GPAsciiConnection is available.

Syntax

Syntax	Description
C#	bool GpAsciiConnected (**get**;)
Visual Basic	**ReadOnly Property** GpAsciiConnected **As** Boolean **Get**

Visual C++	**property** bool GpAsciiConnected {
	bool **get** ();
	}

IAsyncGpasciiCommunicationInterface.IP_Address Property

IP Address of PMAC.

Syntax

Syntax	Description
C#	string IP_Address (**get**;)
Visual Basic	**ReadOnly Property** IP_Address **As** String **Get**

Visual C++	**property** String^ IP_Address {
	String^ **get** ();
	}

IAsyncGpasciiCommunicationInterface.SocketConnected Property

Boolean Value indicating whether or not the Socket Connected.

Syntax

Syntax	Description
C#	bool SocketConnected (**get**;)
Visual Basic	**ReadOnly Property** SocketConnected **As** Boolean **Get**

Visual C++	**property** bool SocketConnected {
	bool **get** ();
	}

IAsyncGpasciiCommunicationInterface Events

The IAsyncCommunicationInterface type exposes the members explained in the following table.

Name	Description
AsyncDataAvailable	AsyncDataAvailable Event need to be consume by Client for receiving GetResponse Data.
ComERROR	Event handler when there is a Socket Error
log	Log of Error Events

IAsyncGpasciiCommunicationInterface.AsyncDataAvailable Event

AsyncDataAvailable Event need to be consume by Client for receiving GetResponse Data.

Syntax

Syntax	Description
C#	event AsyncDataReceiveEvent AsyncDataAvailable
Visual Basic	Event AsyncDataAvailable **As** AsyncDataReceiveEvent

Visual C++	**event** AsyncDataREeceiveEvent^ AsyncDataAvailable {
	void add (AsyncDataReceiveEvent^ *value*);
	void remove (AsyncDataReceiveEvent^ *value*);
	}

IAsyncGpasciiCommunicationInterface.ComERROR Event

Event handler when there is a Socket Error.

Syntax

Syntax	Description
C#	event SocketErMessages ComERROR
Visual Basic	Event ComERROR **As** SocketErMessages

Visual C++	**event** SocketErMessages^ ComERROR {
	void add (SocketErMessages^ *value*);
	void remove (SocketErMessages^ *value*);
	}

IAsyncGpasciiCommunicationInterface.log Event

Log of Error Events.

Syntax

Syntax	Description
C#	event LogMessages log
Visual Basic	Event logR **As** LogMessages

Visual C++	**event** LogMessages^ log {
	void add (LogMessages^ *value*);
	void remove (LogMessages^ *value*);
	}

IAsyncTerminalCommunicationInterface

Power PMAC communication Library.

Syntax

Syntax	Description
C#	**public interface** IAsyncTerminalCommunicationInteface
Visual Basic	**Public Interface** IAsyncTerminalCommunicationInterface
Visual C++	**public interface class** IAsyncTerminalCommunicationInterface

IASyncTerminalCommunicationInterface Members

The IAsyncTerminalCommunicationInterface type exposes the members explained below.

Methods

Name	Description
AsyncSendCommand	Sends Command to Power PMAC
ConnectTerminal(String, Int32, String, String)	Opens the SSH port communication
ConnectTerminal(String, Int32, Int32, String, String)	Opens the SSH port communication
DisconnectTerminal	Closes the SSH port communication
BeginConnectTerminal(String, Int32, String, String)	Asynchronous open SSH port communication

Properties

Name	Description
EchoOn	Boolean Value indicating if EchoMode is on.
IP_Address	IPAddress of PMAC
SocketConnected	Boolean Value indicating if the Socket is connected.

Events

Name	Description
AsyncDataAvailable	AsyncDataAvailable Event need to be consume by Client for receiving GetResponse Data.
ComERROR	Event handler when there is a Socket Error
log	Log of Error Events

IAsyncTerminalCommunicationInterface Methods

The IAsyncTerminalCommunicationInterface type exposes the members explained in the following table.

Name	Description
AsyncSendCommand	Sends Command to Power PMAC
ConnectTerminal(String, Int32, String, String)	Opens the SSH port communication
ConnectTerminal(String, Int32, Int32, String, String)	Opens the SSH port communication
DisconnectTerminal	Closes the SSH port communication
BeginConnectTerminal(String, Int32, String, String)	Asynchronous open SSH port communication

IAsyncTerminalCommunicationInterface.AsyncSendCommand Method

Sends Command to Power PMAC.

Syntax

Syntax	Description
C#	Status AsyncSendCommand(string *cmd*)
Visual Basic	**Function** AsyncSendCommand (_ *cmd* **As** String _) **As** Status
Visual C++	Status AsyncSendCommand(String^ *cmd*)

Parameters

cmd
Type: System.String in the .NET Framework
Terminal Command to be sent.

Return Value

True if Successful.

IAsyncTerminalCommunicationInterface.ConnectTerminal Method

Overload List

Name	Description
ConnectTerminal(String, Int32, String, String)	Opens the SSH port communication
ConnectTerminal(String, Int32, Int32, String, String)	Opens the SSH port communication

IASyncTerminalCommunicationInterface.ConnectTerminal Method (String, Int˜ ˜ , String, String)

Opens the SSH port communication.

Syntax

Syntax	Description
C#	bool ConnectTerminal(string *ipaddr*, int *port*, string *userName*, string *password*)

Syntax	Description
Visual Basic	**Function** ConnectTerminal (_ *ipaddr* **As** String, _ *port* **As** Integer, _ *userName* **As** String, _ *password* **As** String _) **As** Boolean
Visual C++	bool ConnectTerminal(String^ *ipaddr*, int *port*, String^ *userName*, String^ *password*)

Parameters

Parameter	Type	Description
ipaddr	System.String in the .NET Framework	IP Address
port	System.Int32 in the .NET Framework	Port Number
userName	System.String in the .NET Framework	Username
password	System.String in the .NET Framework	Password

Return Value

True if Successful.

IAsyncTerminalCommunicationInterface.ConnectTerminal Method (String, Int˜ ˜ , Int˜ ˜ , String, String)

Opens the SSH port communication.

Syntax

Syntax	Description
C#	bool ConnectTerminal(string *ipaddr*, int *port*, int *BufferSize*, string *userName*, string *password*)
Visual Basic	**Function** ConnectTerminal (_ *ipaddr* **As** String, _ *port* **As** Integer, _ *BufferSize* **As** Integer, _ *userName* **As** String, _ *password* **As** String _) **As** Boolean
Visual C++	bool ConnectTerminal(String^ *ipaddr*, int *port*, int *BufferSize*, String^ *userName*, String^ *password*)

Parameters

Parameter	Type	Description
ipaddr	System.String in the .NET Framework	IP Address
port	System.Int32 in the .NET Framework	Port Number
bufferSize	System.Int32 in the .NET Framework	Username
userName	System.String in the .NET Framework	Password
password	System.String in the .NET Framework	IP Address

Return Value

True if successful.

IAsyncTerminalCommunicationInterface.DisconnectTerminal Method

Closes the SSH port communication.

Syntax

Syntax	Description
C#	bool DisconnectTerminal()
Visual Basic	**Function** DisconnectTerminal **As** Boolean
Visual C++	bool DisconnectTerminal()

Return Value

True if Successful.

IAsyncGpasciiCommunicationInterface.BeginConnectTerminal Method

This asynchronous function opens the SSH port for Power PMAC communication.

Syntax

Syntax	Description
C#	Task<bool> BeginConnectTerminal(string ipaddr, Int˜ 2 port, string userName, string password)
Visual Basic	**Function** BeginConnectTerminal () **As** Task(Of Boolean)
Visual C++	Task<bool> BeginConnectTerminal (string ipaddr, Int˜ 2 port, string userName, string password)

Return Value

True if successful.

IAsyncTerminalCommunicationInterface Properties

Name	Description
EchoOn	Boolean Value indicating if EchoMode is on.
IP_Address	IP Address of PMAC
SocketConnected	Boolean Value indicating if the Socket is Connected.

IAsyncTerminalCommunicationInterface.EchoOn Property

Boolean Value indicating if EchoMode is on.

Syntax

Syntax	Description
C#	bool EchoOn (**get**; **set**;)
Visual Basic	**Property** EchoOn **As** Boolean **Get** **Set**

Visual C++	**property** bool EchoOn {
	bool **get** ();
	void set (bool *value*);
	}

IAsyncTerminalCommunicationInterface.IP_Address Property

IP Address of PMAC.

Syntax

Syntax	Description
C#	string IP_Address (**get**;)
Visual Basic	**ReadOnly Property** IP_Address **As** String **Get**

Visual C++	**property** String^ IP_Address {
	String^ **get** ();
	}

IAsyncTerminalCommunicationInterface.SocketConnected Property

Boolean Value indicating if the Socket is Connected.

Syntax

Syntax	Description
C#	bool SocketConnected (**get**;)
Visual Basic	**ReadOnly Property** SocietConnected **As** Boolean **Get**

Visual C++	**property** bool SocietyConnected {
	bool **get** ();
	}

IAsncTerminalCommunicationInterface Events

The IAsyncTerminalCommunicationInterface type exposes the members explained in the following table.

Name	Description
AsyncDataAvailable	AsyncDataAvailable Event need to be consume by Client for receiving GetResponse Data.
ComERROR	Event handler when there is a Socket Error
log	Log of Error Events

IAsyncTerminalCommunicationInterface.AsyncDataAvailable Event

AsyncDataAvailable Event need to be consume by Client for receiving GetResponse Data.

Syntax

Syntax	Description
C#	**event** AsyncDataReceiveEvent AsyncDataAvailable
Visual Basic	**Event** AsyncDataAvailable **As** AsyncDataReceiveEvent

Visual C++	**event** AsyncDataReceiveEvent^ AsyncDataAvailable {
	void add (AsyncDataReceiveEvent^ *value*);
	void remove (AsyncDataReceiveEvent^ *value*);
	}

IAsyncTerminalCommunicationInterface.ComERROR Event

Event handler when there is a Socket Error.

Syntax

Syntax	Description
C#	**event** SocketErMessages ComERROR
Visual Basic	**Event** ComERROR **As** SocketErMessages

Visual C++	**event** SocketErMessages^ ComERROR {
	void add (SocketErMessages^ *value*);
	void remove (SocketErMessages^ *value*);
	}

IAsyncTerminalCommunicationInterface.log Event

Log of Error Events.

Syntax

Syntax	Description
C#	**event** LogMessages log
Visual Basic	**Event** log **As** LogMessages

Visual C++	**event** LogMessages^ log {
	void add (LogMessages^ *value*);
	void remove (LogMessages^ *value*);
	}

ITCPSocketCommunicationInterface

Library implementing TCP Socket Client communication.

Syntax

Syntax	Description
C#	**public interface** ITCPSocketCommunicationInterface
Visual Basic	**Public Interface** ITCPSocketCommunicationInterface
Visual C++	**public interface class** ITCPSocketCommunicationInterface

ITCPSocketCommunicationInterface Members

The ITCPSocketCommunicationInterface type exposes the members explained below.

Methods

Name	Description
connectToServer	Connection of client to the server.
disconnectFromServer	Disconnection of client to the server.
getConnectionStatus	Returns the connection status
DartGetUserMemChar	Returns the char data type values from the user memory
DartSetUserMemChar	Set char value in the user memory
DartGetUserMemShort	Returns the short data type values from the user memory
DartSetUserMemShort	Set short value in the user memory
DartGetUserMemInt	Returns the Int data type values from the user memory
DartSetUserMemInt	Set Int value in the user memory
DartGetUserMemFloat	Returns the float data type values from the user memory
DartSetUserMemFloat	Set Float value in the user memory
DartGetUserMemDouble	Returns the double data type values from the user memory
DartSetUserMemDouble	Set double value in the user memory
DartGetUserMemPvar	Returns the PVar value from the user memory
DartSetUserMemPvar	Set the PVar value

Properties

Name	Description
dTypeDict	Dictonary property for data type
getRequestDict	Returns the data type
setRequestDict	Set the data type value

ITCPSocketCommunicationInterface Methods

The ITCPSocketCommunicationInterface type exposes the members explained in the following table.

Name	Description
connectToServer	Connection of client to the server.
disconnectFromServer	Disconnection of client to the server.
getConnectionStatus	Returns the connection status
DartGetUserMemChar	Returns the char data type values from the user memory
DartSetUserMemChar	Set char value in the user memory
DartGetUserMemShort	Returns the short data type values from the user memory
DartSetUserMemShort	Set short value in the user memory
DartGetUserMemInt	Returns the Int data type values from the user memory
DartSetUserMemInt	Set Int value in the user memory
DartGetUserMemFloat	Returns the float data type values from the user memory
DartSetUserMemFloat	Set Float value in the user memory
DartGetUserMemDouble	Returns the double data type values from the user memory
DartSetUserMemDouble	Set double value in the user memory
DartGetUserMemPvar	Returns the PVar value from the user memory
DartSetUserMemPvar	Set the PVar value

ITCPSocketCommunicationInterface.connectToServer Method

Asynchrounous Connection of client to the server.

Syntax

Syntax	Description
C#	Task<bool> connectToServer (string *ipaddress*, int *port*, int *timeoutMs*)
Visual Basic	Private Function connectToServer(ByVal ipaddress As String, ByVal port As Integer, ByVal timeoutMs As Integer) As Task(Of Boolean)
Visual C++	Task<bool> connectToServer (String^ *ipaddress*, int^ *port*, int^ *timeoutMs*)

Parameters

Parameter	Type	Description
ipaddress	System.String in the .NET Framework	IP Address
port	System.Int in the .NET Framework	Port No
timeoutMs	System.Int in the .NET Framework	Teme out MS

Return Value

1 if successful.

ITCPSocketCommunicationInterface.disconnectFromServer Method

Disconnect connection from the server.

Syntax

Syntax	Description
C#	int disconnectFromServer ()
Visual Basic	**Function** disconnectFromServer () **As** Integer
Visual C++	int disconnectFromServer ()

Return Value

1 if successful.

ITCPSocketCommunicationInterface.getConnectionStatus Method

Boolean Value indicating connection status.

Syntax

Syntax	Description
C#	bool getConnectionStatus ()
Visual Basic	**Function** getConnectionStatus () **As** Boolean
Visual C++	bool getConnectionStatus ()

Return Value

True if if the Client is communicating with the server.

ITCPSocketCommunicationInterface.DartGetUserMemChar Method

Read character values from user memory address.

Syntax

Syntax	Description
C#	Task<string[]> DartGetUserMemChar (int *address*, int *numvars*)
Visual Basic	Private Function DartGetUserMemChar(ByVal address As Integer, ByVal numvars As Integer) As Task(Of String())
Visual C++	public: array<String>^ DartGetUserMemChar (int address, int numVars)

Parameters

Parameter	Type	Description
address	System.Int in the .NET Framework	Starting address from where values to read
numvars	System.Int in the .NET Framework	Total number of values read from the start address

Return Value

String[] if successful.

ITCPSocketCommunicationInterface.DartGetUserMemShort Method

Read values of the short data type from the user memory address.

Syntax

Syntax	Description
C#	Task<short[]> DartGetUserMemShort (int *address*, int *numvars*)
Visual Basic	Private Function DartGetUserMemShort (ByVal address As Integer, ByVal numvars As Integer) As Task(Of Short())
Visual C++	public: array<short>^ DartGetUserMemShort (int address, int numVars)

Parameters

Parameter	Type	Description
address	System.Int in the .NET Framework	Starting address from where values to read
numvars	System.Int in the .NET Framework	Total number of values read from the start address

Return Value

Short[] if successful.

ITCPSocketCommunicationInterface.DartGetUserMemInt Method

Read values of the Int data type from the user memory address.

Syntax

Syntax	Description
C#	Task<int[]> DartGetUserMemInt (int *address*, int *numvars*)
Visual Basic	Private Function DartGetUserMemInt (ByVal address As Integer, ByVal numvars As Integer) As Task(Of Integer())
Visual C++	public: array<int>^ DartGetUserMemInt (int address, int numVars)

Parameters

Parameter	Type	Description
address	System.Int in the .NET Framework	Starting address from where values to read
numvars	System.Int in the .NET Framework	Total number of values read from the start address

Return Value

Int[] if successful.

ITCPSocketCommunicationInterface.DartGetUserMemFloat Method

Read values of the float data type from the user memory address.

Syntax

Syntax	Description
C#	Task<float[]> DartGetUserMemFloat (int *address*, int *numvars*)
Visual Basic	Private Function DartGetUserMemFloat (ByVal address As Integer, ByVal numvars As Integer) As Task(Of float())
Visual C++	public: array<float>^ DartGetUserMemFloat (int address, int numVars)

Parameters

Parameter	Type	Description
address	System.Int in the .NET Framework	Starting address from where values to read
numvars	System.Int in the .NET Framework	Total number of values read from the start address

Return Value

float[] if successful.

ITCPSocketCommunicationInterface.DartGetUserMemDouble Method

Read values of the double data type from the user memory address.

Syntax

Syntax	Description
C#	Task<double[]>DartGetUserMemDouble (int *address*, int *numvars*)
Visual Basic	Private Function DartGetUserMemDouble (ByVal address As Integer, ByVal numvars As Integer) As Task(Of double())
Visual C++	public: array<double>^ DartGetUserMemDouble (int address, int numVars)

Parameters

Parameter	Type	Description
address	System.Int in the .NET Framework	Starting address from where values to read
numvars	System.Int in the .NET Framework	Total number of values read from the start address

Return Value

double[] if successful.

ITCPSocketCommunicationInterface.DartGetUserMemPvar Method

Read Pvar values from user memory address.

Syntax

Syntax	Description
C#	Task<string[]> DartGetUserMemPvar (int *address*, int *numvars*)
Visual Basic	Private Function DartGetUserMemPvar (ByVal address As Integer, ByVal numvars As Integer) As Task(Of String())
Visual C++	public: array<short>^ DartGetUserMemPvar (int address, int numVars)

Parameters

Parameter	Type	Description
address	System.Int in the .NET Framework	Starting address from where values to read
numvars	System.Int in the .NET Framework	Total number of values read from the start address

Return Value

String[] if successful.

ITCPSocketCommunicationInterface.DartSetUserMemChar Method

Set char data type values in the user memory.

Syntax

Syntax	Description
C#	Task<bool> DartSetUserMemChar(int address, int numVars, string[] strData)
Visual Basic	Private Function DartSetUserMemChar(ByVal address As Integer, ByVal numVars As Integer, ByVal strData As String()) As Task(Of Boolean)
Visual C++	public: bool DartSetUserMemChar (int address, int numVars, array<String^> ^ strData)

Parameters

Parameter	Type	Description
address	System.Int in the .NET Framework	Starting address from where values to read
numvars	System.Int in the .NET Framework	Total number of values to read from the start address
strData	System.String in the .NET Framework	String array of values to write in the User memory

Return Value

True if successful.

ITCPSocketCommunicationInterface.DartSetUserMemShort Method

Set short data type values in the user memory.

Syntax

Syntax	Description
C#	Task<bool> DartSetUserMemShort (int address, int numVars, string[] strData)
Visual Basic	Private Function DartSetUserMemShort (ByVal address As Integer, ByVal numVars As Integer, ByVal strData As String()) As Task(Of Boolean)
Visual C++	public: bool DartSetUserMemShort (int address, int numVars, array<String^> ^ strData)

Parameters

Parameter	Type	Description
address	System.Int in the .NET Framework	Starting address from where values to read
numvars	System.Int in the .NET Framework	Total number of values to read from the start address
strData	System.String in the .NET Framework	String array of values to write in the User memory

Return Value

True if successful.

ITCPSocketCommunicationInterface.DartSetUserMemInt Method

Set Int data type values in the user memory.

Syntax

Syntax	Description
C#	Task<bool> DartSetUserMemInt (int address, int numVars, string[] strData)
Visual Basic	Private Function DartSetUserMemInt (ByVal address As Integer, ByVal numVars As Integer, ByVal strData As String()) As Task(Of Boolean)
Visual C++	public: bool DartSetUserMemInt (int address, int numVars, array<String^> ^ strData)

Parameters

Parameter	Type	Description
address	System.Int in the .NET Framework	Starting address from where values to read
numvars	System.Int in the .NET Framework	Total number of values to read from the start address
strData	System.String in the .NET Framework	String array of values to write in the User memory

Return Value

True if successful.

ITCPSocketCommunicationInterface.DartSetUserMemDouble Method

Set double data type values in the user memory.

Syntax

Syntax	Description
C#	Task<bool> DartSetUserMemDouble (int address, int numVars, string[] strData)
Visual Basic	Private Function DartSetUserMemDouble (ByVal address As Integer, ByVal numVars As Integer, ByVal strData As String()) As Task(Of Boolean)
Visual C++	public: bool DartSetUserMemDouble (int address, int numVars, array<String^> ^ strData)

Parameters

Parameter	Type	Description
address	System.Int in the .NET Framework	Starting address from where values to read
numvars	System.Int in the .NET Framework	Total number of values to read from the start address
strData	System.String in the .NET Framework	String array of values to write in the User memory

Return Value

True if successful.

ITCPSocketCommunicationInterface. DartSetUserMemFloat Method

Set float data type values in the user memory.

Syntax

Syntax	Description
C#	Task<bool> DartSetUserMemFloat (int address, int numVars, string[] strData)
Visual Basic	Private Function DartSetUserMemFloat (ByVal address As Integer, ByVal numVars As Integer, ByVal strData As String()) As Task(Of Boolean)
Visual C++	public: bool DartSetUserMemFloat (int address, int numVars, array<String^> ^ strData)

Parameters

Parameter	Type	Description
address	System.Int in the .NET Framework	Starting address from where values to read
numvars	System.Int in the .NET Framework	Total number of values to read from the start address
strData	System.String in the .NET Framework	String array of values to write in the User memory

Return Value

True if successful.

ITCPSocketCommunicationInterface.DartSetUserMemPvar Method

Set Pvar values in the user memory.

Syntax

Syntax	Description
C#	Task<bool> DartSetUserMemPvar (int address, int numVars, string[] strData)
Visual Basic	Private Function DartSetUserMemPvar (ByVal address As Integer, ByVal numVars As Integer, ByVal strData As String()) As Task(Of Boolean)
Visual C++	public: bool DartSetUserMemPvar (int address, int numVars, array<String^> ^ strData)

Parameters

Parameter	Type	Description
address	System.Int in the .NET Framework	Starting address from where values to read
numvars	System.Int in the .NET Framework	Total number of values to read from the start address
strData	System.String in the .NET Framework	String array of values to write in the User memory

Return Value

True if successful.

IFTPClientInterface

Library implementing FTP Client communication.

Syntax

Syntax	Description
C#	**public interface** IFTPClientInterface
Visual Basic	**Public Interface** IFTPClilentInterface
Visual C++	**public interface class** IFTPClientInterface

IFTPClientInterface Members

The IFTPClientInterface type exposes the members explained below.

Methods

Name	Description
Connect	NotImplemented
BeginConnectFTP	Asynchrounous Connection of client the server.
ConnectFTP	Connects client to the server.
CreateDirectory	Creates a directory on the server side.
DirectoryExists	Boolean Value indicating if Directory Exists
DisconnectFTP	Disconnects client to the server.
DownloadDirectory	Returns true if Directory is successfully downloaded
DownloadDirectoryAbsPath	Returns true if Directory is successfully downloaded
DownloadFile	Returns True if file is successfully downloaded
FileInfoList	NotImplemented
GetFileSize	Returns the file size of a file path from the client.
GetFileTreeNode	NotImplemented
GetRemoteFileList	NotImplemented
RemoveFile	Deletes file from client.
UploadDirectory	Adds directoy to targeted folder under specified folder name. .
UploadDirectoryAbsPath	Adds directoy to targeted folder under specified folder name. .
UploadFile	Adds file to targeted file path

Properties

Name	Description
FTPConnected	Boolean value that indicated if the program is connected via FTP.
IsConnected	Boolean value that indicates if the program is connected.

Events

Name	Description
LogMessages	Represents the Method that handles IFTPClientInterface.LogMessages Event
OnEnd	Event on end of transfer period.
OnStart	Event on start of transfer period.
Progress	Event that is triggered during the transfer process.

IFTPClientInterface Methods

The IFTPClientInterface type exposes the members explained in the following table.

Name	Description
Connect	NotImplemented
BeginConnectFTP	Asynchrounous Connection of client the server.
ConnectFTP	Connects client to the server.
CreateDirectory	Creates a directory on the server side.
DirectoryExists	Boolean Value indicating if Directory Exists
DisconnectFTP	Disconnects client to the server.
DownloadDirectory	Returns true if Directory is successfully downloaded
DownloadDirectoryAbsPath	Returns true if Directory is successfully downloaded
DownloadFile	Returns True if file is successfully downloaded
FileInfoList	NotImplemented
GetFileSize	Returns the file size of a file path from the client.
GetFileTreeNode	NotImplemented
GetRemoteFileList	NotImplemented
RemoveFile	Deletes file from client.
UploadDirectory	Adds directoy to targeted folder under specified folder name. .
UploadDirectoryAbsPath	Adds directoy to targeted folder under specified folder name. .
UploadFile	Adds file to targeted file path

IFTPClientInterface.BeginConnectFTP Method

Asynchronous Connection of client the server.

Syntax

Syntax	Description
C#	Task<bool> BeginConnectFTP(string *ipaddress*, string *username*, string *password*)
Visual Basic	**Function** BeginConnectFTP (_ *ipaddress* As String, _ *userName* As String, _ *password* As String _) **As** Task(of Boolean)
Visual C++	Task<bool> ConnectFTP(String^ *ipaddress*, String^ *userName*, String^ *password*)

Parameters

Parameter	Type	Description
ipaddress	System.String in the .NET Framework	IP Address
userName	System.String in the .NET Framework	UserName
password	System.String in the .NET Framework	Password

Return Value

True if successful.

IFTPClientInterface.ConnectFTP Method

Syntax

Syntax	Description
C#	bool ConnectFTP(string *ipaddress*, string *username*, string *password*)
Visual Basic	**Function** ConnectFTP (_ *ipaddress* As String, _ *userName* As String, _ *password* As String _) **As** Boolean
Visual C++	bool ConnectFTP(String^ *ipaddress*, String^ *userName*, String^ *password*)

Parameters

Parameter	Type	Description
ipaddress	System.String in the .NET Framework	IP Address
userName	System.String in the .NET Framework	UserName
password	System.String in the .NET Framework	Password

Return Value

True if successful.

IFTPClientInterface.CreateDirectory Method

Creates a directory on the Power PMAC. the new direcotries are relative to /var/ftp location.

Syntax

Syntax	Description
C#	bool CreateDirectory(string *TargetFolderPath*)
Visual Basic	**Function** CreateDirectory (_ *TargetFolderPath* As String _) **As** Boolean
Visual C++	bool CreateDirectory (String^ *TargetFolderPath*)

Parameters

TargetFolderPath

Type: System.String in the .NET

Framework

Path of the Directory.

Return Value

VOID

IFTPClientInterface.DirectoryExists Method

Boolean Value indicating if Directory exists.

Syntax

Syntax	Description
C#	bool DirectoryExists(string *dirName*)
Visual Basic	**Function** DirectoryExists (_ *dirName* As String _) **As** Boolean
Visual C++	bool DirectoryExists(String^ *dirName*)

Parameters

dirName
Type: System.String in the .NET Framework
Name of the Directory.

Return Value

True if Directory exists.

IFTPClientInterface.DisconnectFTP Method

Disconnects client to the server.

Syntax

Syntax	Description
C#	bool DisconnectFTP()
Visual Basic	**Function** DisconnectFTP **As** Boolean
Visual C++	bool DisconnectFTP()

Return Value

True if successful.

IFTPClientInterface.DownloadDirectory Method

Downloads a directory to Power PMAC relative to /var/ftp location.

Syntax

Syntax	Description
C#	bool DownloadDirectory(string *srcFolderPath*, string *tragetFolderPath*)
Visual Basic	**Function** DownloadDirectory (_ *srcFolderPath* As String, _ *tragetFolderPath* As String _) **As** Boolean
Visual C++	bool DownloadDirectory(String^ *srcFolderPath*, String^ *tragetFolderPath*)

Parameters

Parameter	Type	Description
srcFolderPath	System.String in the .NET Framework	Path of the source folder
targetFolderPath	System.String in the .NET Framework	Path of the targeted folder.

IFTPClientInterface.DownloadDirectoryAbsPath Method

Downloads a directory to Power PMAC relative to /var/ftp location.

Syntax

Syntax	Description
C#	bool DownloadDirectoryAbsPath(string *LocalSourceFolder*, string *RemoteDestinationFolder*)
Visual Basic	**Function** DownloadDirectoryAbsPath (_ *LocalSourceFolder* As String, _ *RemoteDestinationFolder* As String _) **As** Boolean
Visual C++	bool DownloadDirectoryAbsPath (String^ *LocalSourceFolder*, String^ *RemoteDestinationFolder*)

Parameters

Parameter	Type	Description
LocalSourceFolder	System.String in the .NET Framework	Path of the source folder
RemoteDestinationFolder	System.String in the .NET Framework	Path of the targeted folder.

Return Value

True if successful.

IFTPClientInterface.DownloadFile Method

Returns True if file is successfully downloaded.

Syntax

Syntax	Description
C#	bool DownLoadFile(string Sourcefilename, string DestinationFilename)
Visual Basic	**Function** DownloadFile (_ *inFilePath* **As** String, _ *ServerFullPath* **As** String _) **As** Boolean
Visual C++	bool DownloadFile(String^ *sourceFileName*, String^ *DestinationFilename*)

Parameters

Parameter	Type	Description
sourceFilename	System.String in the .NET Framework	File name at source.
DestinationFilename	System.String in the .NET Framework	Name of file at Destination

Return Value

True if Successful.

IFTPClientInterface.GetFileSize Method

Returns the file size of a file located on the Power PMAC.

Syntax

Syntax	Description
C#	int GetFileSize(string *filename*)
Visual Basic	**Function** GetFileSize (_ *filename* **As** String _) **As** Integer
Visual C++	int GetFileSize(String^ *filename*)

Parameters

filename

Type: System.String in the .NET Framework.

Name of the file involved.

Return Value

Returns true if successful.

IFTPClientInterface.RemoveFile Method

Deletes file from client.

Syntax

Syntax	Description
C#	**void** RemoveFile(string *filename*)
Visual Basic	**Sub** RemoveFile (_ *filename* **As** String _)
Visual C++	**void** RemoveFile(String^ *filename*)

Parameters

filename
Type: System.String in the .NET Framework
Name of selected file.

IFTPClientInterface.UploadDirectory Method

Adds directory to targeted folder under specified folder name.

Syntax

Syntax	Description
C#	bool UploadDirectory(string *infolderPath*, string *outfolderPath*, bool *bUseFolderName*)
Visual Basic	**Function** UploadDirectory (_ *infolderPath* **As** String, _ *outfolderPath* **As** String, _ *bUseFolderName* **As** Boolean _) **As** Boolean
Visual C++	bool UploadDirectory(String^ *infolderPath*, String^ *outfolderPath*, bool *bUseFolderName*)

Parameters

Parameter	Type	Description
infolderPath	System.String in the .NET Framework	Initial folder path
outfolderPath	System.String in the .NET Framework	New folder path
bUseFolderName	System.Boolean in the .NET Framework	Folders name in new directory

Return Value

Returns true if successful transfer.

IFTPClientInterface.UploadDirectoryAbsPath Method

Uploads a directory to Power PMAC relative to /var/ftp location.

Syntax

Syntax	Description
C#	bool UploadDirectoryAbsPath(string *RemoteSourceFolder*, string *LocalDestinationFolder*)
Visual Basic	**Function** DownloadDirectoryAbsPath (_ *RemoteSourceFolder* As String, _ *LocalDestinationFolder* As String _) **As** Boolean
Visual C++	bool DownloadDirectoryAbsPath (String^ *RemoteSourceFolder*, String^ *LocalDestinationFolder*)

Parameters

Parameter	Type	Description
RemoteSourceFolder	System.String in the .NET Framework	Path of the source folder
LocalDestinationFolder	System.String in the .NET Framework	Path of the targeted folder.

Return Value

True if successful

IFTPClientInterface.UploadFile Method

Adds file to targeted file path.

Syntax

Syntax	Description
C#	bool UploadFile(string *filename*, string *targetfilePath*)
Visual Basic	**Function** UploadFile (_ *filename* **As** String, _ *targetfilePath* **As** String _) **As** Boolean
Visual C++	bool UploadFile(String^ *filename*, String^ *targetfilePath*)

Parameters

Parameter	Type	Description
filename	System.String in the .NET Framework	Name of selected file
targetfilePath	System.String in the .NET Framework	Path of file destination.

Return Value

Returns true if successfully transferred.

IFTPClientInterface Properties

The IFTPClientInterface type exposes the members explained in the following table.

Name	Description
FTPConnected	Boolean value that indicated if the program is connected via FTP.
IsConnected	Boolean value that indicates if the program is connected.

IFTPClientInterface.FTPConnected Property

Boolean value that indicates if the program is connected via FTP.

Syntax

Syntax	Description
C#	bool FTPConnected { **get**; }
Visual Basic	**ReadOnly Property** FTPConnected **As** Boolean **Get**

Visual C++	**property** bool FTPConnected {
	bool **get** ();
	}

IFTPClientInterface.IsConnected Property

Boolean value that indicates if the program is connected.

IFTPClientInterface Events

The IFTPClientInterface type exposes the members explained in the following table.

Name	Description
LogMessages	Represents the Method that handles IFTPClientInterface.LogMessages Event
OnEnd	Event on end of transfer period.
OnStart	Event on start of transfer period.
Progress	Event that is triggered during the transfer process.

IFTPClientInterface.LogMessages Event

Represents the Method that handles IFTPClientInterface.LogMessages Event.

Syntax

Syntax	Description
C#	**event** sftp_LogMessages LogMessages
Visual Basic	**Event** LogMessages **As** sftp_LogMessages

Visual C++	**event** sftp_LogMessages^ LogMessages {
	void add (sftp_LogMessages^ *value*);
	void remove (sftp_LogMessages^ *value*);
	}

IFTPClientInterface.OnEnd Event

Event on end of transfer period.

Syntax

Syntax	Description
C#	**event** sftp_OnTransferEnd OnEnd
Visual Basic	**Event** OnEnd **As** sftp_OnTransferEnd

Visual C++	**event** sftp_OnTransferEnd^ OnEnd {
	void add (sftp_OnTransferEnd^ *value*);
	void remove (sftp_OnTransferEnd^ *value*);
	}

IFTPClientInterface.OnStart Event

Event on start of transfer period.

Syntax

Syntax	Description
C#	**event** sftp_OnTransferStart OnStart
Visual Basic	**Event** OnStart **As** sftp_OnTransferStart

Visual C++	**event** sftp_OnTransferStart^ OnStart {
	void add (sftp_OnTransferStart^ *value*);
	void remove (sftp_OnTransferStart^ *value*);
	}

IFTPClientInterface.Progress Event

Event that is triggered during the transfer process.

Syntax

Syntax	Description
C#	**event** sftp_TransferProgress Progress
Visual Basic	**Event** Progress **As** sftp_TransferProgress

Visual C++	**event** sftp_TransferProgress^ Progress {
	void add (sftp_TransferProgress^ *value*);
	void remove (sftp_TransferProgress^ *value*);
	}

IGetErrorsEventInterface

Implements methods involving Error events.

Syntax

Syntax	Description
C#	**public interface** IGetErrorsEventInterface
Visual Basic	**Public Interface** IGetErrorsEventInterface
Visual C++	**public interface class** IGetErrorsEventInterface

IGetErrorsEventInterface Members

The IGetErrorsEventInterface type exposes the members explained below.

Methods

Name	Description
ConnectEvent	This function open the Telnet port
DisconnectEvent	Disconnect SSH port and then close the SSH Port.
StartGetErrorsEvent	Sends geterrors command.
StopGetErrorsEvent	Stops getsends command

Properties

Name	Description
ErrMaskWords	List of error message words.
IP_Address	IP address
IsRunning	Boolean value indicating if program is running.
SocketConnected	Boolean value indicating if socket is connected.
UpdatePeriod	Incremental value on which to update.

Events

Name	Description
ComERROR	Communication error messages.
ErrorsMask	Gets current errormasks for all four items, Motor mask, Coordinate System mask, Global mask, and MACRO ring mask.
ErrorsResponse	Response when error event is triggered.
log	Log of messages.

IGetErrorsEventInterface Methods

The IGetErrorsEventInterface type exposes the members explained in the following table.

Name	Description
ConnectEvent	This function open the Telnet port
DisconnectEvent	Disconnect SSH port and then close the SSH Port.
StartGetErrorsEvent	Sends geterrors command.
StopGetErrorsEvent	Stops getsends command

IGetErrorsEventInterface.ConnectEvent Method

This function open the Telnet port.

Syntax

Syntax	Description
C#	bool ConnectEvent (string *ipaddr*, int *port*, string *username*, string *password*)

Syntax	Description
Visual Basic	**Function** ConnectEvent (_ *ipaddr* **As** String, _ *port* **As** Integer, _ *userName* **As** String, _ *password* **As** String _) **As** Boolean
Visual C++	bool ConnectEvent(String^ *ipaddr*, int *port*, String^ *userName*, String^ *password*)

Parameters

Parameter	Type	Description
ipaddr	System.String in the .NET Framework	IP Address
port	System.Int32 in the .NET Framework	Port
userName	System.String in the .NET Framework	User Name
password	System.String in the .NET Framework	Password

Return Value

True on successful Connection.

IGetErrorsEventInterface.DisconnectEvent Method

Disconnect SSH port and then close the SSH Port.

Syntax

Syntax	Description
C#	bool DisconnectEvent()
Visual Basic	**Function** DisconnectEvent **As** Boolean
Visual C++	bool DisconnectEvent()

Return Value

True if successful.

IGetErrorsEventInterface.StartGetErrorsEvent Method

Sends geterrors command. Following options can be specified in geterrors command: -t updateperiod (in msec) specifies how often to read error table from the memory -0 -˜ -2 -˜ -˜ specify which port(s) to monitor -g -m -c -o specify global, motor, coordinate system and/or ring custom masks if needed.

Syntax

Syntax	Description
C#	Status StartGetErrorsEvent()
Visual Basic	**Function** StartGetErrorsEvent **As** Status
Visual C++	Status StartGetErrorsEvent()

Return Value

True on successful Connection.

IGetErrorsEventInterface.StopGetErrorsEvent Method

Stops getsends command.

Syntax

Syntax	Description
C#	**void** StopGetErrorsEvent()
Visual Basic	**Sub** StopGetErrorsEvent
Visual C++	**void** StopGetErrorsEvent()

IGetErrorsEventInterface Properties

The IGetErrorsEventInterface type exposes the members explained in the following table.

Name	Description
ErrMaskWords	List of error message words.
IP_Address	IP address
IsRunning	Boolean value indicating if program is running.
SocketConnected	Boolean value indicating if socket is connected.
UpdatePeriod	Incremental value on which to update.

IGetErrorsEventInterface.ErrMaskWords Property

List of error message words.

Syntax

Syntax	Description
C#	string[] ErrMask Words { **get**; **set**; }
Visual Basic	**Property** ErrMaskWords **As** String() **Get** **Set**

Visual C++	**property array**<String^>^ ErrMaskWords {
	array<String^>^ **get** ();
	void set (**array**<String^>^ value);
	}

IGetErrorsEventInterface.IP_Address Property

IP Address.

Syntax

Syntax	Description
C#	String IP_Address { **get**; }
Visual Basic	**ReadOnly Property** IP_Address **As** String **Get**

Visual C++	**property** String^ IP_Address {
	String^ **get** ();
	}

IGetErrorsEventInterface.IsRunning Property

Boolean value indicating if program is running.

Syntax

Syntax	Description
C#	bool IsRunning { **get**; }
Visual Basic	**ReadOnly Property** IsRunning **As** Boolean **Get**

Visual C++	**property** bool IsRunning {
	bool **get** ();
	}

IGetErrorsEventInterface.SocketConnected Property

Boolean value indicating if socket is connected.

Syntax

Syntax	Description
C#	bool SocketConnected { **get**; }
Visual Basic	**ReadOnly Property** SocketConnected **As** Boolean **Get**

Visual C++	**property** bool SocketConnected {
	bool **get** ();
	}

IGetErrorsEventInterface.UpdatePeriod Property

Incremental value on which to update.

Syntax

Syntax	Description
C#	float UpdatePeriod { **get**; **set**; }
Visual Basic	**Property** UpdatePeriod **As** Single **Get** **Set**

Visual C++	**property** float UpdatePeriod {
	float **get** ();
	void set (float *value*);
	}

IGetErrorsEventInterface Events

The IGetErrorsEventInterface type exposes the members explained in the following table.

Name	Description
AsyncDataAvailable	Event notifying Asynchronous data is available.
ComERROR	Communication error messages.
log	Log of messages.

IGetErrorsEventInterface.ComERROR Event

Communication Error messages.

Syntax

Syntax	Description
C#	**event** SocketErMessages ComERROR
Visual Basic	**Event** ComERROR **As** SocketErMessages

Visual C++	**event** SocketErMessages^ ComError {
	void add (SocketErMessages^ *value*);
	void remove (SocketErMessages^ *value*);
	}

IGetErrorsEventInterface.ErrorsMask Event

Gets current errormasks for all four items, Motor mask, Coordinate System mask, Global mask, and MACRO ring mask.

Syntax

Syntax	Description
C#	**event** GetErrorsMask ErrorsMask
Visual Basic	**Event** ErrorsMask **As** GetErrorsMask

Visual C++	**event** GetErrorsMask^ ErrorsMask {
	void add (GetErrorsMask^ *value*);
	void remove (GetErrorsMask^ *value*);
	}

IGetErrorsEventInterface.ErrorsResponse Event

Response when error event is triggered.

Syntax

Syntax	Description
C#	**event** GetErrorsResponse ErrorsResponse
Visual Basic	**Event** ErrorsResponse **As** GetErrorsResponse

Visual C++	**event** GetErrorsResponse^ ErrorsResponse {
	void add (GetErrorsResponse^ *value*);
	void remove (GetErrorsResponse^ *value*);
	}

IGetErrorsEventInterface.log Event

Log of messages.

Syntax

Syntax	Description
C#	**event** LogMessages log
Visual Basic	**Event** log **As** LogMessages

Visual C++	**event** LogMessages^ log {
	void add (LogMessages^ *value*);
	void remove (LogMessages^ *value*);
	}

ISyncGpasciiCommunicationInterface

Provides an interface for synchronous Gpascii communication to Power PMAC.

Syntax

Syntax	Description
C#	**public interface** ISyncGpasciiCommunicationInterface
Visual Basic	**Public Interface** ISyncGpasciiCommunicationInterface
Visual C++	**public interface class** IScyncGpasciiCommunicationInterface

ISyncGpasciiCommunicationInterface Members

The ISyncGpasciiCommunicationInterface type exposes the members below.

Methods

Name	Description
BeginConnectGpAscii(String, Int32, String, String)	This function open the SSH port and opens Gpascii port with -2 option for PMAC communication.
ConnectGpAscii(String, Int32, String, String)	This function open the SSH port and opens Gpascii port with -2 option for PMAC communication.
ConnectGpAscii(String, Int32, String, String, Int32)	This function open the SSH port and opens Gpascii port with -2 option for PMAC communication.
Disconnect Gpascii	Disconnect Gpascii port and then close the SSH Port.
GetResponse(List(string), List(String))	Receive the string command and 'Out' the string response.
GetResponse(String, String)	Receive the string command and 'Out' the string response.
GetResponse(List(String), Boolean, List(String))	Receive the string command and 'Out' the string response.
GetResponse(List(String), List(String), Int32)	Receive the string command and 'Out' the string response.
GetResponse(String, String, Int32)	Receive the string command and 'Out' the string response.
GetResponse(List(String), Boolean, List(String), Int32)	Receive the string command and 'Out' the string response.
GPASciiConnect(String, Int32, String, string, Int32)	This function open the SSH port and opens Gpascii port with -2 option for PMAC communication.
PowerPMACReset	Reset Command $$$, $$$** handled differently.
PowerPMACSave	Issues SAVE command to Power PMAC

Properties

Name	Description
EchoOn	Boolean value indicating if Echo mode is on.
GpAsciiConnected	Boolean value indicating if GPAscii is connected.
IP_Address	IP Address
SocketConnected	Boolean value indicating if socket is connected.

Events

Name	Description
ComERROR	Communication error messages.
log	Log of messages.

ISyncGpasciiCommunicationInterface Methods

The ISyncGpasciiCommunicationInterface type exposes the members explained in the following table.

Name	Description
BeginConnectGpAscii(String, Int32, String, String)	This function open the SSH port and opens Gpascii port with -2 option for PMAC communication.
ConnectGpAscii(String, Int32, String, String)	This function open the SSH port and opens Gpascii port with -2 option for PMAC communication.
ConnectGpAscii(String, Int32, String, String, Int32)	This function open the SSH port and opens Gpascii port with -2 option for PMAC communication.
Disconnect Gpascii	Disconnect Gpascii port and then close the SSH Port.
GetResponse(String, String, Int32)	Receive the string command and 'Out' the string response.
GetResponse(List(String), Boolean, List(String), Int32)	Receive the string command and 'Out' the string response.
GetResponse(List(string), List(String))	Receive the string command and 'Out' the string response.
GetResponse(String, String)	Receive the string command and 'Out' the string response.
GetResponse(List(String), Boolean, List(String))	Receive the string command and 'Out' the string response.
GetResponse(String, String, Int32)	Receive the string command and 'Out' the string response.
PowerPMACReset	Reset Command $$$, $$$** handled differently.
PowerPMACSave	Issues SAVE command to Power PMAC

ISyncGpasciiCommunicationInterface.BeginConnectGpAscii Method

This function opens the SSH port and opens Gpascii port with -2 option for PMAC communication. This is a asynchronous call.

Syntax

Syntax	Description
C#	bool BeginConnectGpAscii(string *ipaddr*, int *port*, string *userName*, string *password*)
Visual Basic	**Function** BeginConnectGpAscii (_ *ipaddr* **As** String, _ *port* **As** Integer, _ *userName* **As** String, _ *password* **As** String _) **As** Boolean
Visual C++	bool BeginConnectGpAscii(String^ *ipaddr*, int *port*, String^ *userName*, String^ *password*)

Parameters

Parameter	Type	Description
ipaddr	System.String in the .NET Framework	IP Address
port	System.Int32 in the .NET Framework	Port
userName	System.String in the .NET Framework	Username
password	System.String in the .NET Framework	Password

Return Value

Returns True upon success.

ISyncGpasciiCommunicationInterface.ConnectGpAscii Method

Overload List

Name	Description
ConnectGpAscii(String, Int32, String, String)	Sends command to the Power PMAC over SSH Server
ConnectGpAscii(String, Int32, String, String, Int32)	Sends command to the Power PMAC over SSH Server

ISyncGpasciiCommunicationInterface.ConnectGpAscii Method (String, Int32, String, String)

Sends command to the Power PMAC over SSH Server.

Syntax

Syntax	Description
C#	bool ConnectGpAscii(string *ipaddr*, int *port*, string *userName*, string *password*)
Visual Basic	**Function** ConnectGpAscii (_ *ipaddr* **As** String, _ *port* **As** Integer, _ *userName* **As** String, _ *password* **As** String _) **As** Boolean
Visual C++	bool ConnectGpAscii(String^ *ipaddr*, int *port*, String^ *userName*, String^ *password*)

Parameters

Parameter	Type	Description
ipaddr	System.String in the .NET Framework	IP Address
port	System.Int32 in the .NET Framework	Port
userName	System.String in the .NET Framework	Username
password	System.String in the .NET Framework	Password

Return Value

Returns True upon success.

ISyncGpasciiCommunicationInterface.ConnectGpAscii Method (String, Int32, String, String, Int32)

Sends command to the Power PMAC over SSH Server.

Syntax

Syntax	Description
C#	bool ConnectGpAscii(string *ipaddr*, int *port*, string *userName*, string *password,* int *buffSize*)
Visual Basic	**Function** ConnectGpAscii (_ *ipaddr* **As** String, _ *port* **As** Integer, _ *userName* **As** String, _ *password* **As** String, _ *buffSize* **As** Integer _) **As** Boolean

Syntax	Description
Visual C++	bool ConnectGpAscii(String^ *ipaddr*, int *port*, String^ *userName*, String^ *password,* int *buffSize*)

Parameters

Parameter	Type	Description
ipaddr	System.String in the .NET Framework	IP Address
port	System.Int32 in the .NET Framework	Port
userName	System.String in the .NET Framework	Username
password	System.String in the .NET Framework	Password
buffSize	System.Int32 in the .NET Framework	Buffer Size

Return Value

Return true upon success.

ISyncGpasciiCommunicationInterface.DisconnectGpascii Method

Disconnect Gpascii port and then close the SSH Port.

Syntax

Syntax	Description
C#	bool DisconnectGpascii()
Visual Basic	**Function** DisconnectGpascii **As** Boolean
Visual C++	bool DisconnectGpascii()

Return Value

True upon success.

ISyncGpasciiCommunicationInterface.GetResponse Method

Overload List

Name	Description
GetResponse(List(string), List(String))	Receive the string command and 'Out' the string response.
GetResponse(String, String)	Receive the string command and 'Out' the string response.
GetResponse(List(String), Boolean, List(String))	Receive the string command and 'Out' the string response.
GetResponse(List(String), List(String), Int32)	Receive the string command and 'Out' the string response.
GetResponse(String, String, Int32)	Receive the string command and 'Out' the string response.
GetResponse(List(String), Boolean, List(String), Int32)	Receive the string command and 'Out' the string response.

ISyncGpasciiCommunicationInterface.GetResponse Method (List(String), List(String))

Receive the string command and 'Out' the string response.

Syntax

Syntax	Description
C#	Status GetResponse(List<string> *cmd*, **out** List<string> *response*)
Visual Basic	**Function** GetResponse (_ *cmd* **As** List(**Of** String), _ <OutAttribute> **ByRef** *response* **As** List(**Of** String) _) **As** Status
Visual C++	Status GetResponse(List<String^>^ *cmd*, [OutAttribute] List<String^>^% *response*)

Parameters

Parameter	Type	Description
cmd	System.Collections.Generic.List(String) in the .NET Framework	Command to PMAC
response	System.Collections.Generic.List(String) in the .NET Framework	Response from PMAC

Return Value

Returns true upon success.

ISyncGpasciiCommunicationInterface.GetResponse Method (String, String)

Receive the string command and 'Out' the string response.

Syntax

Syntax	Description
C#	Status GetResponse(string *cmd*, **out** string *response*)
Visual Basic	**Function** GetResponse (_ *cmd* **As** String, _ <OutAttribute> **ByRef** *response* **As** String _) **As** Status
Visual C++	Status GetResponse(String^ *cmd*, [OutAttribute] String^% *response*)

Parameters

Parameter	Type	Description
cmd	System.Collections.Generic.List(String) in the .NET Framework	Command to PMAC
response	System.Collections.Generic.List(String) in the .NET Framework	Response from PMAC

Return Value

Return True upon success.

ISyncGpasciiCommunicationInterface.GetResponse Method (List(String), Boolean, List(String))

Receive the string command and 'Out' the string response.

Syntax

Syntax	Description
C#	Status GetResponse(List<string> *cmd*, bool *parseData*, **out** List<string> *response*)
Visual Basic	**Function** GetResponse (_ *cmd* **As** List(**Of** String), _ *parseData* **As** Boolean, <OutAttribute> **ByRef** *response* **As** List(**Of** String) _) **As** Status
Visual C++	Status GetResponse(List<String^>^ *cmd*, [OutAttribute] List<String^>^% *response*)

Parameters

Parameter	Type	Description
cmd	System.Collections.Generic.List(String) in the .NET Framework	Command to PMAC
parseData	System.Boolean in the .NET Framework	Data Parsed
response	System.Collections.Generic.List(String) in the .NET Framework	Response from PMAC

Return Value

Return true upon success.

ISyncGpasciiCommunicationInterface.Get Response Method (List(String), List(String), Int˜ ˜)

Receive the string command and 'Out' the string response.

Syntax

Syntax	Description
C#	Status GetResponse(List<string> *cmd*, **out** List<string> *response,* int *echoMode*)
Visual Basic	**Function** GetResponse (_ *cmd* **As** List(**Of** String), _ <OutAttribute> **ByRef** *response* **As** List(**Of** String), _ *echoMode* **As** Integer _) **As** Status
Visual C++	Status GetResponse(List<String^>^ *cmd*, [OutAttribute] List<String^>^% *response,* int *echoMode*)

Parameters

Parameter	Type	Description
cmd	System.Collections.Generic.List(String) in the .NET Framework	Command to PMAC
response	System.Collections.Generic.List(String) in the .NET Framework	Response from PMAC
echoMode	System.Int32 in the .NET Framework	Echo mode is on or off.

Return Value

Return True upon success.

ISyncGpasciiCommunicationInterface.GetResponse Method (String, String, Int˜ ˜)

Receive the string command and 'Out' the string response.

Syntax

Syntax	Description
C#	Status GetResponse(string *cmd*, **out** string *response,* int *echoMode*)
Visual Basic	**Function** GetResponse (_ *cmd* **As** String, _ <OutAttribute> **ByRef** *response* **As** String, _ *echoMode* **As** Integer _) **As** Status

Syntax	Description
Visual C++	Status GetResponse(String^ *cmd*, [OutAttribute] String^% *response,* int *echoMode*)

Parameters

Parameter	Type	Description
cmd	System.Collections.Generic.List(String) in the .NET Framework	Command to PMAC
response	System.Collections.Generic.List(String) in the .NET Framework	Response from PMAC
echoMode	System.Int32 in the .NET Framework	Echo mode is on or off.

Return Value

Return true upon success.

ISyncGpasciiCommunicationInterface.GetResponse Method (List(String), Boolean, List(String), Int˜ ˜)

Receive the string command and 'Out' the string response.

Syntax

Syntax	Description
C#	Status GetResponse(List<string> *cmd*, bool *parseData*, **out** List<string> *response,* int *echoMode*)
Visual Basic	**Function** GetResponse (_ *cmd* **As** List(**Of** String), _ *parseData* **As** Boolean, _ <OutAttribute> **ByRef** *response* **As** List(**Of** String), _ *echoMode* **As** Integer _) **As** Status
Visual C++	Status GetResponse(List<String^>^ *cmd*, [OutAttribute] List<String^>^% *response,* int *echoMode*)

Parameters

Parameter	Type	Description
cmd	System.Collections.Generic.List(String) in the .NET Framework	Command to PMAC
parseData	System.Boolean in the .NET Framework	Data Parsed
response	System.Collections.Generic.List(String) in the .NET Framework	Response from PMAC
echoMode	System.Int32 in the .NET Framework	Echo mode is on or off.

Parameter	Type	Description

Return Value

Return true upon success.

ISyncGpasciiCommunicationInterface.PowerPMACResetMethod

Reset Command $$$, $$$** handled differently.

Syntax

Syntax	Description
C#	bool PowerPMACReset(ResetType *cmdType*)
Visual Basic	**Function** PowerPMACReset (_ *cmdType* **As** ResetType _) **As** Boolean
Visual C++	bool PowerPMACReset(ResetType *cmdType*)

Parameters

Parameter	Type	Description
cmdType	ODT.PowerPmacComLib.ResetType	Command Type $$$ or $$$***

Return Value

Returns true upon success.

ISyncGpasciiCommunicationInterface.PowerPMACSave Method

Issues SAVE command to Power PMAC.

Syntax

Syntax	Description
C#	bool PowerPMACSave()
Visual Basic	**Function** PowerPMACSave **As** Boolean
Visual C++	bool PowerPMACSave

Return Value

Returns true upon success.

ISyncGpasciiCommunicationInterface Properties

The ISyncGpasciiCommunicationInterface type exposes the members explained in the following table.

Name	Description
EchoOn	Boolean value indicating whether or not Echo mode is on.
GpAsciiConnected	Boolean value indicating if GPAscii is connected.
IP_Address	IP address

Name	Description
SocketConnected	Boolean value indicating if socket is connected.

ISyncGpasciiCommunicationInterface.EchoOn Property

Boolean value indicating whether or not Echo mode is on.

Syntax

Syntax	Description
C#	bool EchoOn { **get**; **set**; }
Visual Basic	**Property** EchoOn **As** Boolean **Get** **Set**
Visual C++	**property** bool EchoOn { bool **get** (); **void set** (bool *value*);)

ISyncGpasciiCommunicationInterface.GpAsciiConnected Property

Boolean value indicating if GPAscii is connected.

Syntax

Syntax	Description
C#	bool GpAsciiConnected { **get**; }
Visual Basic	**ReadOnly Property** GpAsciiConnected **As** Boolean **Get**
Visual C++	**property** bool GpAsciiConnected { bool **get** ();)

ISyncGpasciiCommunicationInterface.IP_Address Property

IP Address.

Syntax

Syntax	Description
C#	string IP_Address { **get**; }
Visual Basic	**ReadOnly Property** IP_Address **As** String **Get**
Visual C++	**property** String^ IP_Address { String^ **get** ();)

ISyncGpasciiCommunicationInterface.SocketConnected Property

Boolean value indicating if socket is connected.

Syntax

Syntax	Description
C#	bool SocketConnected { **get**; }
Visual Basic	**ReadOnly Property** SocketConnected **As** Boolean **Get**

Syntax	Description
Visual C++	**property** bool SocketConnected { bool **get** ();)

ISyncGpasciiCommunicationInterface Events

The ISyncGpasciiCommunicationInterface type exposes the members explained in the following table.

Name	Description
ComERROR	Communication error messages.
log	Log of messages.

ISyncGpasciiCommunicationInterface.ComERROR Event

Communication error messages.

Syntax

Syntax	Description
C#	**event** SocketErMessages ComERROR
Visual Basic	**Event** ComERROR **As** SocketErMessages

Visual C++	**event** SocketErMessages^ ComError {
	void add (SocketErMessages^ *value*);
	void remove (SocketErMessages^ *value*);
	}

ISyncGpasciiCommunicationInterface.log Event

Log of messages.

Syntax

Syntax	Description
C#	**event** LogMessages log
Visual Basic	**Event** log **As** LogMessages

Visual C++	**event** L9ogMessages^ log {
	void add (LogMessages^ *value*);
	void remove (LogMessages^ *value*);
	}

ISyncTerminalCommunicationInterface

Provides an interface for synchronous Terminal communication to Power PMAC.

Syntax

Syntax	Description
C#	**public interface** ISyncTerminalCommunicationInterface
Visual Basic	**Public Interface** ISyncTerminalCommunicationInterface
Visual C++	**public interface class** ISyncTerminalCommunicationInterface

ISyncTerminalCommunicationInterface Members

The ISyncTerminalCommunicationInterface type exposes the members explained below.

Methods

Name	Description
ChangeDir	Change directory
ConnectTerminal	This function open the Telnet port as Telnet terminal.
BeginConnectTerminal	This function open the Telnet port as Telnet terminal.
DisconnectTerminal	Disconnect Telnet port and then close the Telnet Port.
ExecCommand(String)	Receive the Operating System(OS) string command or PowerPMAC command and 'out' the OS string response.
ExecCommand(String, Int32)	Receive the Operating System(OS) string command or PowerPMAC command and 'out' the OS string response.
SendCommand	Receive the Operating System(OS) string command and 'out' the OS string response.

Properties

Name	Description
EchoOn	Boolean value indicating whether or not Echo mode is on.
IP_Address	IP Address
SocketConnected	Boolean value indicating if socket is connected.

Events

Name	Description
ComERROR	Communication error messages.
ExResponse	Execution response.
log	Log of messages

ISyncTerminalCommunicationInterface Methods

Name	Description
ChangeDir	Change directory
ConnectTerminal	This function open the Telnet port as Telnet terminal.
BeginConnectTerminal	This function open the Telnet port as Telnet terminal.
DisconnectTerminal	Disconnect Telnet port and then close the Telnet Port.
ExecCommand(String)	Receive the Operating System(OS) string command or PowerPMAC command and 'out' the OS string response.
ExecCommand(String, Int32)	Receive the Operating System(OS) string command or PowerPMAC command and 'out' the OS string response.
SendCommand	Receive the Operating System(OS) string command and 'out' the OS string response.

ISyncTerminalCommunicationInterface.ChangeDir Method

Change the directory.

Syntax

Syntax	Description
C#	bool ChangeDir(string *path*)
Visual Basic	**Function** ChangeDir (_ *path* **As** String _) **As** Boolean
Visual C++	bool ChangeDir(String^ *path*)

Parameters

path
Type: System.String in the .NET Framework
File path

Return Value

True upon success.

ISyncTerminalCommunicationInterface.BeginConnectTerminal Method

This function open the Telnet port as Telnet terminal. This is an asynchronous call.

Syntax

Syntax	Description
C#	bool BeginConnectTerminal(string *ipaddr,* int *port,* string *userName,* string *password*)
Visual Basic	**Function** BeginConnectTerminal (_ *ipaddr* **As** String, _ *port* **As** Integer, _ *userName* **As** String, _ *password* **As** String _) **As** Boolean
Visual C++	bool BeginConnectTerminal(String^ *ipaddr,* int *port,* String^ *userName,* String^ *password*)

Parameters

Parameter	Type	Description
ipaddr	System.String in the .NET Framework	IP Address
port	System.Int32 in the .NET Framework	Port
userName	System.String in the .NET Framework	Username
password	System.String in the .NET Framework	Password

Return Value

Return true upon success.

ISyncTerminalCommunicationInterface.ConnectTerminal Method

This function open the Telnet port as Telnet terminal.

Syntax

Syntax	Description
C#	bool ConnectTerminal(string *ipaddr,* int *port,* string *userName,* string *password*)
Visual Basic	**Function** ConnectTerminal (_ *ipaddr* **As** String, _ *port* **As** Integer, _ *userName* **As** String, _ *password* **As** String _) **As** Boolean
Visual C++	bool ConnectTerminal(String^ *ipaddr,* int *port,* String^ *userName,* String^ *password*)

Parameters

Parameter	Type	Description
ipaddr	System.String in the .NET Framework	IP Address
port	System.Int32 in the .NET Framework	Port
userName	System.String in the .NET Framework	Username
password	System.String in the .NET Framework	Password

Return Value

Return true upon success.

ISyncTerminalCommunicationInterface.DisconnectTerminal Method

Disconnect Telnet port and then close the SSH Port.

Syntax

Syntax	Description
C#	bool DisconnectTerminal()
Visual Basic	**Function** DisconnectTerminal **As** Boolean
Visual C++	bool DisconnectTerminal()

Return Value

Return true upon success.

ISyncTerminalCommunicationInterface.ExecCommand Method

Overload List

Name	Description
ExecCommand(String)	Receive the Operating System(OS) string command or PowerPMAC command and 'out' the OS string response.
ExecCommand(String, Int32)	Receive the Operating System(OS) string command or PowerPMAC command and 'out' the OS string response.

ISyncTerminalCommunicationInterface.ExecCommand Method (String)

Syntax

Syntax	Description
C#	Status ExecCommand(string *cmd*)
Visual Basic	**Function** ExecCommand (_ *cmd* **As** String _) **As** Status
Visual C++	Status ExecCommand(String^ *cmd*)

Parameters

cmd
Type: System.String in the .NET Framework
Command to PMAC

Return Value

Return True upon success.

ISyncTerminalCommunicationInterface.ExecCommand Method (String, Int32)

Receive the Operating System(OS) string command or PowerPMAC command and 'out' the OS string response.

Syntax

Syntax	Description
C#	Status ExecCommand(string *cmd,* int *timeout*)
Visual Basic	**Function** ExecCommand (_ *cmd* **As** String, _ *timeout* **As** Integer _) **As** Status
Visual C++	Status ExecCommand(String^ *cmd,* int *timeout*)

Parameters

Parameter	Type	Description
cmd	System.String in the .NET Framework	Command to PMAC
timeout	System.Int32 in the .NET Framework	Integer indicating time out.

Return Value

Return true upon success.

ISyncTerminalCommunicationInterface.SendCommand Method

Receive the Operating System(OS) string command and 'out' the OS string response.

Syntax

Syntax	Description
C#	Status SendCommand(string *cmd,* out string *response*)
Visual Basic	**Function** SendCommand (_ *cmd* **As** String, _ <OutAttribute> **ByRef** *response* As String_) **As** Status
Visual C++	Status SendCommand(String^ *cmd,* [OutAttribute] String^% *response*)

Parameters

Parameter	Type	Description
cmd	System.String in the .NET Framework	Command to PMAC
response	System.String in the .NET Framework	Response from PMAC

Return Value

Return true upon success.

ISyncTerminalCommunicationInterface Properties

The ISyncTerminalCommunicationInterface type exposes the members explained in the following table.

Name	Description
EchoOn	Boolean value indicating whether or not Echo mode is on.
IP_Address	IP Address
SocketConnected	Boolean value indicating whether or not Echo mode is on.

ISyncTerminalCommunicationInterface.EchoOn Property

Boolean value indicating whether or not Echo mode is on.

Syntax

Syntax	Description
C#	bool EchoOn { **get**; **set**; }
Visual Basic	**Property** EchoOn **As** Boolean **Get** **Set**

Visual C++	**property** bool EchoOn {
	bool **get** ();
	void **set** (bool *value*);
	}

ISyncTerminalCommunicationInterface.IP_Address Property

IP Address.

Syntax

Syntax	Description
C#	string IP_Address { **get**; }
Visual Basic	**ReadOnly Property** IP_Address **As** String **Get**

Visual C++	**property** String^ IP_Address {
	String^ **get** ();
	}

ISyncTerminalCommunicationInterface.SocketConnected Property

Boolean value indicating if socket is connected.

Syntax

Syntax	Description
C#	bool SocketConnected { **get**; }
Visual Basic	**ReadOnly Property** SocketConnected **As** Boolean **Get**

Visual C++	**property** bool SocketConnected {
	bool **get** ();
	}

ISyncTerminalCommunicationInterface Events

The ISyncTerminalCommunicationInterface type exposes the members explained in the following table.

Name	Description
ComERROR	Communication error messages.
ExResponse	Execution response.
log	Log of messages

ISyncTerminalCommunicationInterface.ComERROR Event

Communication error messages.

Syntax

Syntax	Description
C#	**event** SocketErMessages ComERROR
Visual Basic	Event ComError As SocketErMessages

Visual C++	**event** SocketErMessages^ ComERROR {
	void add (SocketErMessages^ *value*);
	void remove (SocketErMessages^ *value*);
	}

ISyncTerminalCommunicationInterface.ExResponse Event

Execution response.

Syntax

Syntax	Description
C#	**event** CommandResponse ExResponse
Visual Basic	Event ExResponse As CommandResponse

Visual C++	**event** CommandResponse^ ExResponse {
	void add (CommandResponse^ *value*);
	void remove (CommandResponse^ *value*);
	}

ISyncTerminalCommunicationInterface.log Event

Log of messages.

Syntax

Syntax	Description
C#	**event** LogMessages log
Visual Basic	Event log As LogMessages

Visual C++	**event** LogMessages^ log {
	void add (LogMessages^ *value*);
	void remove (LogMessages^ *value*);
	}

IUnsolicitedEventInterface

Provides an interface for Unsolicited Messages communication to Power PMAC.

Syntax

Syntax	Description
C#	**public interface** IUnsolicitedEventInterface
Visual Basic	Public Interface IUnsolicitedEventInterface
Visual C++	**public interface class** IUnsolicitedEventInterface

IUnsolicitedEventInterface Members

The IUnsolicitedEventInterface type exposes the members explained below.

Methods

Name	Description
ConnectEvent	This function open the SSH port
DisconnectEvent	Disconnect Telnet port and then close the Telnet Port.
SendString	Returns all the Terminal data as String.
StartUnsolicitedEvent	Sends getsends command for Unsolicited event
StopUnsolicitedEvent	Stops getsends command
unsolicitedcompletestatus	Represents the class for Unsolicited port status including port busy and status string.

Properties

Name	Description
Buffer0	Buffer 0
Buffer1	Buffer 1
Buffer2	Buffer 2
Buffer3	Buffer 3
Buffer4	Buffer 4
IP_Address	IP address
IsRunning	Boolean value indicating if program is running.
SocketConnected	Boolean value indicating if socket is connected.

Events

Name	Description
ComERROR	Communication error messages.
log	Log of messages.
UResponse	Event for handling Unsolicited response

IUnsolicitedEventInterface Methods

The IUnsolicitedEventInterface type exposes the members explained in the following table.

Name	Description
ConnectEvent	This function open the SSH port
DisconnectEvent	Disconnect Telnet port and then close the Telnet Port.
SendString	Returns all the Terminal data as String.
StartUnsolicitedEvent	Sends getsends command for Unsolicited event
StopUnsolicitedEvent	Stops getsends command
unsolicitedcompletestatus	Represents the class for Unsolicited port status including port busy and status string.

IUnsolicitedEventInterface.ConnectEvent Method

This function open the SSH port.

Syntax

Syntax	Description
C#	bool ConnectEvent (string *ipaddr*, int *port*, string *userName*, string *password*)
Visual Basic	Function ConnectEvent (_ *ipaddr* As String, _ *port* As Integer, _ *userName* As String, _ *password* As String _) As Boolean
Visual C++	bool ConnectEvent (String^ *ipaddr*, int *port*, String^ *userName*, String^ *password*)

Parameters

Parameter	Type	Description
ipaddr	System.String in the .NET Framework	IP Address
port	System.Int32 in the .NET Framework	Port
userName	System.String in the .NET Framework	Username
password	System.String in the .NET Framework	Password

Return Value

Return true upon success.

IUnsolicitedEventInterface.DisconnectEvent Method

Disconnect Telnet port and then close the Telnet Port.

Syntax

Syntax	Description
C#	bool DisconnectEvent ()
Visual Basic	Function DisconnectEvent As Boolean
Visual C++	bool DisconnectEvent ()

Return Value

Return true upon success.

IUnsolicitedEventInterface.SendString Method

Returns all the Terminal data as String.

Syntax

Syntax	Description
C#	UnsolicitedStatus SendString (int *portno*, string *cmd*)
Visual Basic	Function SendString (*portno* As Integer, _ *cmd* As String _) As UnsolicitedStatus
Visual C++	UnsolicitedStatus SendString (int *portno*, string^ *cmd*)

Parameters

Parameter	Type	Description
portno	System.Int32 in the .NET Framework	Port Number
cmd	System.String in the .NET Framework	Command to PMAC

Return Value

Return True upon success.

IUnsolicitedEventInterface.StartUnsolicitedEvent Method

Sends getsends command for Unsolicited event.

Syntax

Syntax	Description
C#	UnsolicitedStatus StartUnsolicitedEvent()
Visual Basic	Function StartUnsolicitedEvent As UnsolicitedStatus
Visual C++	UnsolicitedStatus StartUnsolicitedEvent()

Return Value

Return true upon success.

IUnsolicitedEventInterface.StopUnsolicitedEvent Method

Stops getsends command.

Syntax

Syntax	Description
C#	**void** StopUnsolicitedEvent()
Visual Basic	Sub StopUnsolicitedEvent
Visual C++	**void** StopUnsolicitedEvent()

IUnsolicitedEventInterface.unsolicitedcompletestatus Method

Represents the class for Unsolicited port status including port busy and status string.

Syntax

Syntax	Description
C#	UnsolicitedCompleteStatus unsolicitedcompletestatus(string *cmd*)
Visual Basic	Function unsolicitecompletestatus (_ *cmd* As String _) As UnsolicitedCompleteStatus
Visual C++	UnsolicitedCompleteStatus^ unsolicitedcompletestatus(String^ *cmd*)

Parameters

cmd
Type: System.String in the .NET Framework
Command to PMAC.

Return Value

Return true upon success.

IUnsolicitedEventInterface Properties

The IUnsolicitedEventInterface type exposes the members explained in the following table.

Name	Description
Buffer0	Buffer 0
Buffer1	Buffer 1
Buffer2	Buffer 2
Buffer3	Buffer 3
Buffer4	Buffer 4
IP_Address	IP address
IsRunning	Boolean value indicating if program is running.
SocketConnected	Boolean value indicating if socket is connected.

IUnsolicitedEventInterface.Buffer˜ Property

Buffer 0.

Syntax

Syntax	Description
C#	bool Buffer0 { **get**; **set**; }
Visual Basic	Property Buffer0 As Boolean Get Set

Visual C++	**Property** bool Buffer0 {
	bool **get** ();
	void **set** (bool *value*);
	}

IUnsolicitedEventInterface.Buffer˜Property

Buffer 1.

Syntax

Syntax	Description
C#	bool Buffer1 { **get**; **set**; }
Visual Basic	Property Buffer1 As Boolean Get Set

Visual C++	**Property** bool Buffer1 {
	bool **get** ();
	void **set** (bool *value*);
	}

IUnsolicitedEventInterface.Buffer˜ Property

Buffer 2.

Syntax

Syntax	Description
C#	bool Buffer2 { **get**; **set**; }
Visual Basic	Property Buffer2 As Boolean Get Set

Visual C++	**Property** bool Buffer2 {
	bool **get** ();
	void **set** (bool *value*);
	}

IUnsolicitedEventInterface.Buffer˜ Property

Buffer 3.

Syntax

Syntax	Description
C#	bool Buffer3 { **get**; **set**; }
Visual Basic	Property Buffer3 As Boolean Get Set

Visual C++	**Property** bool Buffer3 {
	bool **get** ();
	void **set** (bool *value*);
	}

IUnsolicitedEventInterface.Buffer˜ Property

Buffer 4.

Syntax

Syntax	Description
C#	bool Buffer4 { **get**; **set**; }
Visual Basic	Property Buffer4 As Boolean Get Set

Syntax	Description

Visual C++	**Property** bool Buffer4 {
	bool **get** ();
	void **set** (bool *value*);
	}

IUnsolicitedEventInterface.IP_Address Property

IP Address.

Syntax

Syntax	Description
C#	string IP_Address { **get**; }
Visual Basic	**ReadOnly Property** IP_Address **As** String **Get**

Visual C++	**property** String^ IP_Address {
	String^ **get** ();
	}

IUnsolicitedEventInterface.IsRunning Property

Boolean value indicating if program is running.

Syntax

Syntax	Description
C#	bool IsRunning { **get**; }
Visual Basic	**ReadOnly Property** IsRunning **As** String **Get**

Visual C++	**property** bool IsRunning {
	bool **get** ();
	}

IUnsolicitedEventInterface.SocketConnected Property

Boolean value indicating if socket is connected.

Syntax

Syntax	Description
C#	bool SocketConnected { **get**; }
Visual Basic	**ReadOnly Property** SocketConnected **As** String **Get**

Visual C++	**property** bool SocketConnected {
	bool **get** ();
	}

IUnsolicitedEventInterface Events

The IUnsolicitedEventInterface type exposes the members explained in the following table.

Name	Description
ComERROR	Communication error messages.
log	Log of messages.
UResponse	Event for handling Unsolicited response

IUnsolicitedEventInterface.ComERROR Event

Communication error messages.

Syntax

Syntax	Description
C#	**event** SocketErMessages ComERROR
Visual Basic	**Event** ComERROR **As** SocketErMessages

Visual C++	**event** SocketErMessages^ ComERROR {
	void add (SocketErMessages^ *value*);
	void remove (SocketErMessages^ *value*);
	}

IUnsolicitedEventInterface.log Event

Log of messages.

Syntax

Syntax	Description
C#	**event** LogMessages log
Visual Basic	**Event** log **As** LogMessages

Visual C++	**event** LogMessages^ log {
	void add (LogMessages^ *value*);
	void remove (LogMessages^ *value*);
	}

IUnsolicitedEventInterface.UResponse Event

Event for handling Unsolicited response.

Syntax

Syntax	Description
C#	**event** UnsolicitedResponse UResponse
Visual Basic	**Event** UResponse **As** UnsolicitedResponse

Visual C++	**event** UnsolicitedResponse^ UResponse {
	void add (UnsolicitedResponse^ *value*);
	void remove (UnsolicitedResponse^ *value*);
	}

Delegates

AsyncDataReceiveEvent

Represents the method that handles the AsyncDataAvailable event.

Syntax

Syntax	Description

C#	**Public Delegate void** AsyncDataReceiveEvent {
	Object *sender,*
	AsyncDataArgs *e*
	}

Visual Basic	**Public Delegate Sub** AsyncDataReceiveEvent { _ *sender* **As** Object, _ *e* **As** AsyncDataArgs _ }

Visual C++	**Public Delegate void** AsyncDataReceiveEvent {
	Object^ sender,
	AsyncDataArgs^ e
	}

Parameters

Parameter	Type	Description
sender	System.Object in the .NET Framework	Sender object of the event
e	ODT.PowerPmacComLib.AsyncDataArgs	Event argument of type AsyncDataArgs

CommandResponse

Represents the method that handles the CommandResponse event.

Syntax

Syntax	Description

C#	**Public delegate void** CommandResponse {
	Object *sender,*
	CommandArgs *e*
	}

Visual Basic	**Public Delegate Sub** CommandResponse { _ *sender* **As** Object, _ *e* **As** CommandArgs _ }

Visual C++	**public delegate void** CommandResponse {
	Object^ *sender*,
	CommandArgs^ e
	}

Parameters

Parameter	Type	Description
sender	System.Object in the .NET Framework	Sender object of the event
e	ODT.PowerPmacComLib.AsyncDataArgs	Event argument of type AsyncDataArgs

GetErrorsMask

Obtains the current error masks for the following masks:

- Motor
- Coordinate System
- Global
- MACRO ring

Syntax

Syntax	Description
C#	**public delegate void** GetErrorsMask(Object *sender*, ErrorsMaskArgs *e*)
Visual Basic	**Public Delegate Sub** GetErrorsMask (_ *sender* **As** Object, _ *e* **As** ErrorsMaskArgs _)
Visual C++	**public delegate void** GetErrorsMask(Object^ *sender*, ErrorsMaskArgs^ *e*)

Parameters

Parameter	Type	Description
sender	System.Object in the .NET Framework	Sender object of the event
e	ODT.PowerPmacComLib.AsyncDataArgs	Event argument of type AsyncDataArgs

GetErrorsResponse

Represents the method that handles the GetErrorsResponse event.

Syntax

Syntax	Description
C#	**public delegate void** GetErrorsResponse(Object *sender*, GetErrorsArgs *e*)
Visual Basic	**Public Delegate Sub** GetErrorsResponse (_ *sender* **As** Object, _

Syntax	Description
	e **As** GetErrorsArgs _)
Visual C++	**public delegate void** GetErrorsResponse(Object^ *sender*, GetErrorsArgs^ *e*)

Parameters

Parameter	Type	Description
sender	System.Object in the .NET Framework	Sender object of the event
e	ODT.PowerPmacComLib.AsyncDataArgs	Event argument of type AsyncDataArgs

LogMessages

Represents the method that handles the LogMessages event.

Syntax

Syntax	Description
C#	**public delegate void** LogMessages(Object *sender*, LogArgs *e*)
Visual Basic	**Public Delegate Sub** LogMessages (_ *sender* **As** Object, _ *e* **As** LogArgs _)
Visual C++	**public delegate void** LogMessages(Object^ *sender*, LogArgs^ *e*)

Parameters

Parameter	Type	Description
sender	System.Object in the .NET Framework	Sender object of the event
e	ODT.PowerPmacComLib.LogArgs	Event argument of type LogArgs

sftp_LogMessages

Represents the method that handles the IFTPClientInterface.LogMessages event.

Syntax

Syntax	Description
C#	**public delegate void** sftp_LogMessages(Object *sender*, SFTPLogArgs *e*)
Visual Basic	**Public Delegate Sub** sftp_LogMessages (_ *sender* **As** Object, _ *e* **As** SFTPLogArgs _)
Visual C++	**public delegate void** sftp_LogMessages(Object^ *sender*, SFTPLogArgs^ *e*)

Parameters

Parameter	Type	Description
sender	System.Object in the .NET Framework	Sender object of the event
e	ODT.PowerPmacComLib.SFTPLogArgs	Event argument of type SFTPlogArgs

sftp_OnTransferEnd

Represents the method that handles the IFTPClientInterface.OnEnd event.

Syntax

Syntax	Description
C#	**public delegate void** sftp_OnTransferEnd (Object *sender*, SFTPArgs *e*)
Visual Basic	**Public Delegate Sub** sftp_OnTransferEnd (_ *sender* **As** Object, _ *e* **As** SFTPArgs _)
Visual C++	**public delegate void** sftp_OnTransferEnd(Object^ *sender*, SFTPArgs^ *e*)

Parameters

Parameter	Type	Description
sender	System.Object in the .NET Framework	Sender object of the event
e	ODT.PowerPmacComLib.SFTPLogArgs	Event argument of type SFTPLogArgs

sftp_OnTransferStart

Represents the method that handles the IFTPClientInterface.OnStart event.

Syntax

Syntax	Description
C#	**public delegate void** sftp_OnTransferStart(Object *sender*, SFTPArgs *e*)
Visual Basic	**Public Delegate Sub** sftp_OnTransferStart (_ *sender* **As** Object, _ *e* **As** SFTPArgs _)
Visual C++	**public delegate void** sftp_OnTransferStart(Object^ *sender*, SFTPArgs^ *e*)

Parameters

Parameter	Type	Description
sender	System.Object in the .NET Framework	Sender object of the event
e	ODT.PowerPmacComLib.SFTPArgs	Event argument of type SFTPArgs

sftp_TransferProgress

Represents the method that handles the IFTPClientInterface.Progress event.

Syntax

Syntax	Description
C#	**public delegate void** sftp_TransferProgress (Object *sender*, SFTPArgs *e*)
Visual Basic	**Public Delegate Sub** sftp_TransferProgress (_ *sender* **As** Object, _ *e* **As** SFTPArgs _)
Visual C++	**public delegate void** sftp_TransferProgress(Object^ *sender*, SFTPArgs^ *e*)

Parameters

Parameter	Type	Description
sender	System.Object in the .NET Framework	Sender object of the event
e	ODT.PowerPmacComLib.SFTPArgs	Event argument of type SFTPArgs

SocketErMessages

Represents the method that handles the SocketErMessages event.

Syntax

Syntax	Description
C#	**public delegate void** SocketErMessages(Object *sender,* ComErArgs *e*)
Visual Basic	**Public Delegate Sub** SocketErMessages (_ *sender* **As** Object, _ *e* **As** ComErArgs _)
Visual C++	**public delegate void** SocketErMessages(Object^ *sender*, ComErArgs^ *e*)

Parameters

Parameter	Type	Description
sender	System.Object in the .NET Framework	Sender object of the event
e	ODT.PowerPmacComLib.ComErArgs	Event argument of type ComErArgs

UnsolicitedResponse

Represents the method that handles the UnsolicitedResponse event.

Syntax

Syntax	Description
C#	**public delegate void** SUnsolicitedResponse(Object *sender,* UnsolicitedArgs *e*)
Visual Basic	**Public Delegate Sub** SUnsolicitedResponse (_ *sender* **As** Object, _ *e* **As** UnsolicitedArgs _)

Syntax	Description
Visual C++	**public delegate void** UnsolicitedResponse(Object^ *sender*, UnsolicitedArgs^ *e*)

Parameters

Parameter	Type	Description
sender	System.Object in the .NET Framework	Sender as object.
e	ODT.PowerPmacComLib.UnsolicitedArgs	UnsolicitedResponse argument.

Enumerations

ErEventMessageType

Enumeration for message types from an error or event generated by a Power PMAC.

Syntax

Syntax	Description
C#	**public enum** ErEventMessageType
Visual Basic	**Public Enumeration** ErEventMessageType
Visual C++	**public enum class** ErEventMessageType

Members

Member Name	Description
Unsolicited	An Unsolicited message sent from Power PMAC
Error	An Error message from GetError process from Power PMAC. It will be pre-defined by the mask
STDOUT	A message sent on STDOUT port of Power PMAC
STDERR	A message sent on STDERR port of Power PMAC
MonitorStart	A flag to monitor the start of the GetError process on Power PMAC
MonitorStop	A flag to monitor when GetError process stops on Power PMAC

ResetType

Reset types are Reset and FullReset.

Syntax

Syntax	Description
C#	**public enum** ResetType
Visual Basic	**Public Enumeration** ResetType
Visual C++	**public enum class** ResetType

Members

Member Name	Description
Reset	Issues $$$ command. It restores Power PMAC to last saved settings.
FullReset	Issues $$$** command. It restores Power PMAC to factory default settings.

Status

Returns the status of the GetResponse method.

Syntax

Syntax	Description
C#	**public enum** Status
Visual Basic	**Public Enumeration** Status
Visual C++	**public enum class** Status

Members

Member Name	Description
Ok	GetResponse returned ok.
Exception	An exception occurred while executing GetResponse.
TimeOut	Execution of GetResponse is taking too long and timed out.
NotConnected	Can not execute GetResponse. Connection to Power PMAC is not established.
Failed	Execution of GetResponse failed.

StatusType

Enumeration for the status type of an error message sent by the GetError process from the Power PMAC.

Syntax

Syntax	Description
C#	**public enum** StatusType
Visual Basic	**Public Enumeration** StatusType
Visual C++	**public enum class** StatusType

Members

Member Name	Description
Global	The error message is a Global error as pre-defined by Global mask
Coordinate	The error message relates to a Coordinate System as pre-defined by Coordinate System mask
Motor	The error message relates to a Motor as pre-defined by Motor mask
Macros	The error message relates to a Macro as pre-defined by Macro mask

Member Name	Description
Unknown	The type of error message is unknown

UnsolicitedPortStatus

Returns the status of the port that sent an unsolicited response.

Syntax

Syntax	Description
C#	**public enum** UnsolicitedPortStatus
Visual Basic	**Public Enumeration** UnsolicitedPortStatus
Visual C++	**public enum class** UnsolicitedPortStatus

Members

Member Name	Description
CommandSent_OK	Command sent to Power PMAC successfully
OK	Command sent to Power PMAC successfully
Illegal_Format	Incorrect syntax of Power PMAC send command
PortBusy	Power PMAC receive port is busy to accept a command
PortFull	Power PMAC receive port is full and can not accept another command
InvalidResponse	Invalid response

UnsolicitedStatus

Returns the status of an unsolicited response.

Syntax

Syntax	Description
C#	**public enum** UnsolicitedStatus
Visual Basic	**Public Enumeration** UnsolicitedStatus
Visual C++	**public enum class** UnsolicitedStatus

Members

Member Name	Description
OK	Unsolicited command successful
Exception	Socket or Unsolicited process exception
TimeOut	Command sent to Power PMAC timed out
NotConnected	Socket not connected to start Unsolicited process on Power PMAC
Failed	Unsolicited process could not be started on Power PMAC
True	Unsolicited process started successfully on Power PMAC
False	Unsolicited process turned off on Power PMAC
BufferNotDefined	Unsolicited buffer is not defined or invalid buffer number
UnsolicitedNotStarted	Unable to start unsolicited port

OMRON Corporation **Industrial Automation Company**

Kyoto, JAPAN **Contact : www.ia.omron.com**

Regional Headquarters

OMRON EUROPE B.V.
Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31) 2356-81-300 Fax: (31) 2356-81-388

OMRON ELECTRONICS LLC
2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A.
Tel: (1) 847-843-7900 Fax: (1) 847-843-7787

OMRON ASIA PACIFIC PTE. LTD.
438B Alexandra Road, #08-01/02 Alexandra
Technopark, Singapore 119968
Tel: (65) 6835-3011 Fax: (65) 6835-3011

OMRON (CHINA) CO., LTD.
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-6023-0333 Fax: (86) 21-5037-2388

Authorized Distributor:

Cat. No. O033-E-03 0125 (0125)