# OPCClientDLL库使用说明

使用方法：将OPCClientDLL.dll文件复制到项目工程目录下，在vs工程的解决方案中添加引用，在程序开头加入 “using OPCClientDLL;”。

使用C#客户端时，首先配置好Fins GateWay和OPCServer，OPCServer中需要提前建立好Tag列表。客户端搜索到服务器并连接时，一般会自动开启OPCServer。

OPC通讯方式如下：

OPC通讯采用客户端-服务器模式。OPC服务器（一般由生产厂家提供）负责与PLC通讯，并提供标准的高级编程语言接口；OPC客户端读取OPC服务器的数据，也就是间接读取PLC的数据。不同厂商提供的PLC的数据协议可能不一样，但使用OPC进行通讯后，我们用户不需要管pc如何读取PLC的数据，因为这部分已经由厂商提供的OPC服务器完成了。我们客户只需要读取OPC服务器的数据。因为OPC的接口是有固定标准的，所以不管下层的PLC硬件如何变化，我们客户都可以使用同一套OPC客户端代码去读取OPC服务器的数据，如此一来便大大加速了自己产品的开发。

## DLL库属性说明

1. updateTime——int类型，指示订阅方式的数据更新周期，缺省为300ms
2. isConnected2Server——bool类型，指示是否连接到服务器
3. tagCountsMax——int类型的静态变量，指示服务器中Tag的最大数量，缺省为50

## DLL库方法说明

1. OPCClientClass

描述：类的构造函数

语法：OPCClientClass()

示例：OPCClientClass myOPCClient =new OPCClientClass();

1. GetTagsCount

描述：设置数据获取方式为订阅时的数据更新周期，默认为200ms

语法：int GetTagsCount()

参数：return——返回int类型，Tag数量

示例：int tagNum=0;

tagNum=myOPCClient.GetTagsCount();

1. SetTagDataUpdateFunc

描述：设置数据发生改变时要调用的方法，一般用于**订阅方式或异步读取方式**

语法：void SetTagDataUpdateFunc(TagDataChange tdc)

参数：tdc——C#代理类型，传入的参数为函数名，该函数的固定格式为void DataChangeFunc(string tag, object itemValue, object quality, object timestamp)，其中tag指示opc服务器中建立的Tag，itemValue指示数据，quality指示品质，timeStamp时间戳

示例：void DataChangeFunc(string tag, object itemValue, object quality, object timestamp){

tbReadValue.Text = tag + ": " + itemValue.ToString() + "\r\n";

tbReadValue.Text += "品质: " + quality.ToString();}

myOPCClient.SetTagDataUpdateFunc(DataChangeFunc)

1. SearchOPCSevers

描述：搜索当前PC中的OPC服务器

语法：string SearchOPCSevers(ref object serverList)

参数：serverList——object类型的引用，存储搜索到的服务器列表

return——返回string类型，指示是否搜索到服务器。如果成功，返回”OK”，否则返回错误信息。

示例：object servers;

string outErrors=myOPCClient.SearchOPCSevers(ref servers);

if ("OK"==outErrors){

textbox1.Text = “已搜索到服务器”;}

1. ConnectToServer

描述：连接到OPC服务器

语法：string ConnectToServer(string serverName, bool initAllTagsFlag=false)

参数：serverName——string类型，指示服务器名称

initAllTagsFlag——bool类型，指示是否初始化所有Tag的数据获取方法为订阅方式

return——返回string类型，指示连接服务器是否成功。如果成功，返回”OK”，否则返回错误信息。

示例：string serverList = (Array)servers;

int tagNum=0;

string outErrors = myOPCClient.ConnectToServer(cmbServers.Text);

if ("OK" == outErrors){

textbox1.Text = “连接服务器成功”;}

1. GetTags

描述：获取服务器所有Tag

语法：void GetTags(string[] tags)

参数：tags——string类型的数组，存放tag数组

示例：string[] tags = new string[tagNum];

myOPCClient.GetTags(tags);

1. InitSomeTags

描述：初始化部分tag的数据获取方式为订阅方式

语法：void InitSomeTags(string[] tags, int count)

参数：tags——string类型的数组，存放要初始化的tag

count——int类型，指示要初始化的tag的数量

示例：string[] tags = new string[2];

tags[0]=”tag1”, tags[1]=”tag2”;

myOPCClient.InitSomeTags(tags, 2);

1. DisconnectToServer

描述：断开与服务器的连接

语法：void DisconnectToServer()

参数：无

示例：myOPCClient.DisconnectToServer();

1. AsyncReadTagValue

描述：异步读取数据。读取到的数据通过SetTagDataUpdateFunc方法获取

语法：string AsyncReadTagValue(string tag)

参数：tag——string类型，指示Tag名称

return——返回string类型，如果成功，返回”OK”，否则返回错误信息。

示例：myOPCClient.AsyncReadTagValue(“tag1”);

1. SyncReadTagValue

描述：同步读取数据

语法：string SyncReadTagValue(string tag, out Array outValues, out object qualities, out object timeStamps)

参数：tag——string类型，指示Tag名称

outValues——Array类型的引用，存储读取到的数据

qualities——object类型的引用，指示数据的品质

timeStamps——object类型的引用，指示时间戳

return——返回string类型，如果成功，返回”OK”，否则返回错误信息。

示例：Array readValues, object qualities, object timeStamps;

myOPCClient.SyncReadTagValue(“tag1”, out readValues, out qualities, out timeStamps);

textbox1.Text = “tag1”+ ": " + readValues.GetValue(1).ToString();//**数组下标从1开始**

1. AsyncWriteTagValue

描述：异步写入数据

语法：string AsyncWriteTagValue(string tag, string writeStr)

参数：tag——string类型，指示Tag名称

writeStr——string类型，指示要写入的数据

return——返回string类型，如果成功，返回”OK”，否则返回错误信息。

示例：myOPCClient.AsyncWriteTagValue(“tag1’’, “88”);

1. SyncWriteTagValue

描述：同步写入数据

语法：string SyncWriteTagValue(string tag, string writeStr)

参数：tag——string类型，指示Tag名称

writeStr——string类型，指示要写入的数据

return——返回string类型，如果成功，返回”OK”，否则返回错误信息。

示例：myOPCClient.SyncWriteTagValue(“tag1’’, “88”);