

IOServer Java Demo 使用说明书

文件编号：		项目编号：	
项目名称：		部 门：	
版 本 号：1.0	受控状态：0		密 级：公开
总 页 数：9	正文：7		附 录：0
编 制：黄展智	审 核：		批 准：
日 期：2017.4.25	日 期：		日 期：

北京亚控科技发展有限公司

IOServer C# API 程序说明文档

时间	文档版本	人员	说明
2017.04.25	V1.0	黄展智	创建文档

目录

目录

IOServer C# API 程序说明文档.....	2
目录.....	3
1、简介	4
1.1 目的	4
1.2 参考资料	4
1.3 其他说明	4
2、类 classDemoClient 函数说明	5
2.1 概述	5
2.2 函数 funcConnect.....	5
2.3 函数 funcDisConnect.....	5
2.4 函数 funcIsConnect.....	5
2.5 函数 funcAsyncRead	5
2.6 函数 getAsyncReadValue	6
2.7 函数 funcStoreTagIdAndName(私有)	6
2.8 函数 funcSyncRead	6
2.9 函数 funcSubscribeAllTags.....	7
3.0 函数 funcGetTagValue	7
3、界面设计说明	7
3.1、功能分布.....	7
3.2、相关刷新定时器	9
4、环境配置说明	9
4.1、jdk 配置	9

1、简介

1.1 目的

本文档是 IO Server Java API 新版 Demo 的设计文档。由于旧版 demo 过于简单，没有界面，演示效果较差。一些函数的调用方式过于简单，缺少相应的保护、延时等处理过程。客户参照做二次开发时，容易出问题，增加后期维护的工作量。因此基于 IO Server 3.56 版本，开发该新版本 Demo 程序。该程序兼容 IO Server 3.55。

该文档用于描述程序的详细设计。作为理解产品设计的基础。

1.2 参考资料

《IO Server 3.6 详细设计文档 03_IO Server Java API》

1.3 其他说明

原函数中有颜色部分为系统关键字，包括数据类型。函数参数 `parameter` 中，只标出关键参数，非关键参数或由命名规则可以正确解释的参数不进行详细说明。

2、类 classDemoClient 函数说明

2.1 概述

该类对 java api 中的主要函数进行二次封装，目的是让程序更加稳定，调用更加方便。该类主要封装了订阅、异步读、同步读等几个常用函数。用户、测试员或者开发人员都可以根据需要在该类中增加函数，对 java api 相应的函数进行二次封装。

2.2 函数 funcConnect

Summary:连接函数

Detail: 主要调用 api 中 IOServerConnecton 函数。增加启动和连接判断。并在连接时注册相应的回调函数。通过返回值返回启动、连接以及回调的注册状态。

Parameter:

- 1、ip: string 类型的 ip 地址
- 2、port: string 类型的端口号(可以修改或重载成 int 类型)

Return:正确返回 0，获取句柄失败返回-2，其他错误返回-1

Function:

```
public int funcConnect(String ip, String port)
```

2.3 函数 funcDisConnect

Summary:断开连接函数

Detail:调用 api 函数 IOServerDisConnect 清除连接句柄。

Function:

```
public int funcDisConnect()
```

2.4 函数 funcIsConnect

Summary:连接状态判断函数

Detail:调用 api 函数 IOServerIsConnected 和 getIOServerWorkStatus 判断连接状态。

Return:正确返回 0，如果连接不成功或者 IOServer 未启动，则返回-1

```
public int funcIsConnect()
```

2.5 函数 funcAsyncRead

Summary:异步读函数

Detail: 主要调用 api 中 AsyncReadTagsValueByNames 函数。增加启动和连接判断。将传入参数从 jna.Wstring 类型转为 Java 通用的 String 类型。读取回调值获取在 getAsyncReadValue 中获取。

Parameter:

- 1、strTagName: 异步读变量数组

Return:正确返回 0，其他返回值为错误

Function:

```
public int funcConnect(String ip, String port)
```

2.6 函数 getAsyncReadValue

Summary:异步读获取数据值函数

Detail: 异步读取的数据会存放在 ClientDataBean 中的 ReadCompleteTag ValueMap 中, 该 map 为私有。调用方法 getReadComTag ValueByName 可以将数据获取出来。

Return:正确返回变量值结构体数组, 错误返回 null

Function:

```
public Struct_TagInfo[] getAsyncReadValue()
```

注:

以上异步读以及其回调时封装好了的, 调用读函数时, 最好让读函数处于单独一个线程, 否则可能造成读取等待与读回调会出现互相等待的情况。读取需要等到回调完成后才能获取数据。因此, api 中还提供了带有自定义回调函数参数的重载注册接口, 如下。通过输入自定义接口, 回调数据将会到达用户自定义的函数中。

```
public short RegisterReadCompleteCallbackFunc(int Handle, ReadCompleteCallBackInf readCompleteCallbackfunc)
```

定义回调函数如下:

```
ReadCompleteCallBackInf readCompleteCallbackfunc = new ReadCompleteCallBackImp();
```

接口源代码如下:

```
public interface ReadCompleteCallBackInf extends Callback {  
    public int invoke(int handle, Struct_TagInfo.ByReference TagValueArray, int Number);}
```

接口实现代码如下:

```
public class ReadCompleteCallBackImp implements ReadCompleteCallBackInf {  
    public int invoke(int handle, ByReference TagValueArray, int Number) {  
        GlobalCilentBean globalBean = GlobalCilentBean.getInstance();  
        ClientDataBean client= globalBean.getClientByHandle(handle);  
        client.AddReadCompleteTag ValueMap(TagValueArray.TagID, TagValueArray);  
        return 0;  
    }  
}
```

2.7 函数 funcStoreTagIdAndName(私有)

Summary:获取变量 Id 和 Name 配对的 map

Detail: 通过调用 api 函数 BrowserCollectTags, 获取所有变量的属性值, 根据属性值组成 Id 为主键, Name 为值的 map。该函数在连接时被调用。用于 id 和名称转换。

Return:正确返回变量个数, 错误返回-1

Function:

```
private int funcStoreTagIdAndName()
```

2.8 函数 funcSyncRead

Summary:同步读变量

Detail: 通过调用 api 函数 SyncReadTags ValueReturnNames, 主要增加连接判断和转换名称类型为 sting。

Parameter:

1、strTagName: 同步读变量数组

Return:正确返回数据值结构体 Struct_TagInfo_AddName[], 错误返回 null

Function:

public Struct_TagInfo_AddName[] funcSyncRead(String[] strTagName)

2.9 函数 funcSubscribeAllTags

Summary:订阅全部变量

Detail: 通过调用 api 函数中层次化浏览函数获取所有变量(还可以通过 funcStoreTagIdAndName 中的方式获取所有变量)。调用 api 中 SubscribeTagValuesChange 函数添加订阅的变量。Api 中 UnSubscribeTagValuesChange 是取消变量的订阅, demo 中没有做该部分的演示。

Return:正确返回所有变量名组成的容器 vecSubscribeTagsName

Function:

public Vector<WString> funcSubscribeAllTags()

3.0 函数 funcGetTagValue

Summary:订阅变量数据值获取

Detail:订阅的变量在其值发生变化时会在 ClientDataBean 中刷新, 存放在 map 中: CollectTagValueMap。通过调用 api 函数 getTagValueByName 可以获取相应变量的数值。

Parameter:

1、TagName: 获取订阅值的变量名数组

Return:返回变量值数组

Function:

public Struct_TagInfo funcGetTagValue(WString TagName)

3、界面设计说明

界面采用的是windows builder插件辅助开发。借助java自带的AWT工具包, AWT是用于创建图形用户界面的一个工具包, 它提供了一系列用于实现图形界面的组件。在JDK中针对每个组件都提供了对应的java类, 这些类位于java.awt包中。

3.1、功能分布

界面的左上角是模拟标签页切换按钮, 分为三个功能: 同步读写、异步读写、订阅。

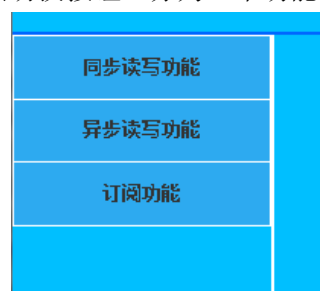


图 1:标签页切换区域

界面的左下角是连接区域, 具有连接、断开 Ioserver 的功能, 可以输入 ip 和端口, 有

指示灯显示当前连接状态。

指示灯：

开

IP：

127.0.0.1

PORT：

12380

连接

断开

图 2:连接区域

界面右侧为读写功能区域，点击对应的功能键，会显示相应的功能。首先介绍订阅功能，如下图，单击订阅按钮，表格中能够显示所有变化的变量值，一般为所有值。

订阅

TagName	TagValue	Time	Quality	Id
Tag1	2	2017-4-25 14:59:43	192	5001
Tag2	2.11	2017-4-25 14:59:43	192	5002
Tag3	0.0	2017-4-25 14:59:43	192	5003

订阅全部变量

图 3:订阅功能

异步读写功能，如图，左下边为读取，数据显示在表格中，右下边为写入

异步读写

TagName	TagValue	Time	Quality	Id
Tag1	2	2017-4-25 15:04:17	192	5001
Tag2	2.11	2017-4-25 15:04:17	192	5002

数据变量名称，用","分隔：

Tag1,Tag2

异步读

变量名：

Tag1

变量值：

3

int

float

string

异步写

图 4:异步读写功能

同步读写功能，如图，

同步读写

TagName	TagValue	Time	Quality	Id
Tag1	2	2017-4-25 15:07:22	192	5001

输入读变量，用","隔开：

Tag1

同步读

变量名：

Tag3

变量值：

2.22

☐ int ☐ string ☒ float

同步写

图 5:同步读写功能

3.2、相关刷新定时器

界面刷新定时器 timer，用于连接状态指示灯，标签页切换，radiobutton 切换。表格大小设置。

数据读取定时器 timer_subscribe，用于订阅时的数据刷新，异步读数据刷新，同步读数据显示。以及其他点击事件刷新检测。

4、环境配置说明

4.1、jdk 配置

4.1.1、jdk 版本说明

Java Api 封装文件 IOServerAPI.jar 目前不支持 1.8 版本的 jdk。并且由于调用的动态链接库位 32 位，所以只支持 32 位 jdk。可以使用资料文件中的 jdk-7u51-windows-i586 版本的 jdk。(适用于旧版的 demo，有时候配置 jar 文件失败就才用 1.7 的 jdk 解决)

新版 demo 支持 1.8 的 jdk。如图所示：



图 6:jdk 版本示意

IOServerAPI.jar 依赖的 dll 文件有：JNativeCpp.dll、kxCommon.dll、kxIOClient.dll、kxIOServrCommon.dll、kxPropX.dll、SORBA31x86.dll、SORBA35dx86.dll、SORBA35x86.dll。

4.1.2、配置环境变量

- 1、右击【我的电脑】---【属性】-----【高级】---【环境变量】
- 2、选择【新建系统变量】--弹出“新建系统变量”对话框，在“变量名”文本框输入“JAVA_HOME”,在“变量值”文本框输入 JDK 的安装路径（也就是步骤 5 的文件夹路径），单击“确定”按钮。
- 3、在“系统变量”选项区域中查看 PATH 变量，如果不存在，则新建变量 PATH，否则选中该变量，单击“编辑”按钮，在“变量值”文本框的起始位置添加“%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;”或者是直接“%JAVA_HOME%\bin;”，单击确定按钮。
- 4、在“系统变量”选项区域中查看 CLASSPATH 变量，如果不存在，则新建变量 CLASSPATH，否则选中该变量，单击“编辑”按钮，在“变量值”文本框的起始位置添加“.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;”。
- 5、windows builder 的安装及使用：<http://blog.csdn.net/kagoy/article/details/17465615>

5、注意事项

5.1、异步读无回调(bug)

如果同时使用了订阅和异步读功能，那么请在注册异步读回调之后加上这样一段代码：

```
int tagId[]=new int[1];
tagId[0]=5001;
client.AsyncReadTagsValueByIds(client.getHandle(), tagId, 1, 0);
```

这是因为程序本身存在 bug。如果不首先进行一次异步读，而是首先进行订阅，异步读回调函数将无法被调用。

5.2、订阅自定义回调函数

订阅回调函数中数据通过 `jna.Pointer` 返回，并采用管道的方式传输，过程复杂，不建议自定义回调函数。如果有这方面的需要，可以联系程序员获取相关的函数实现，不保证能够正常使用。