**目 录**

1. 计算机配置 2

1.1. 注册msvcp100d 2

1.2. 注册msvcr100d 2

2. 总线服务配置 3

2.1. NetRouterConfigure.xml配置文件 3

2.1.1. 示例 3

2.1.2. 广域网配置详细说明 4

2.2. log4cplus.properties 6

3. 应用开发 7

3.1. 配置文件描述 7

3.1.1. Configure.xml 7

3.1.2. log4cplus.properties 8

3.2. Java适配 8

3.2.1. 数据结构描述 8

3.2.2. 接口描述 9

3.2.3. 集成方法 11

3.3. C++适配 11

3.3.1. 数据结构描述 11

3.3.2. 接口描述 12

3.3.3. 集成方法 14

3.4. C#适配 14

3.4.1. 数据结构描述 14

3.4.2. 接口描述 15

3.4.3. 集成方法 17

3.4.4. 引用配置 17

# 计算机配置

所有使用通信总线的计算机都需要注册msvcp100d与msvcr100d。

## 注册msvcp100d

32位操作系统：

将msvcp100d.dll（位于NetRouter软件组\NetRouterClient\CSharp文件夹中）复制到C:\Windows\SYSTEM32下，然后点击开始-运行-输入regsvr32 msvcp100d.dll

64位操作系统：

将msvcp100d.dll(位于NetRouter软件组\NetRouterClient\CSharp文件夹中）复制到c:\Windows\SysWOW64，开始-程序-附件-命令提示符，右键点击，管理员身份运行。键入：

regsvr32 c:\Windows\SysWOW64\msvcp100d.dll

## 注册msvcr100d

32位操作系统：

将msvcr100d.dll文件复制到C:\Windows\SYSTEM32下，然后点击开始-运行-输入regsvr32 msvcr100d.dll

64位操作系统：

将msvcr100d.dll文件复制到c:\Windows\SysWOW64，开始-程序-附件-命令提示符，右键点击，管理员身份运行。键入：

regsvr32 c:\Windows\SysWOW64\msvcr100d.dll

# 总线服务配置

NetRouter的配置文件位于“NetRouter软件组\NetRouter\etcwen”文件夹中，分为NetRouterConfigure.xml应用程序配置文件和log4cplus.properties日志配置文件。

## NetRouterConfigure.xml配置文件

### 示例

<NetRouter>

<LocalAddr>//本网的局域地址

<bureau\_code>8</bureau\_code>//局码

<unit\_type>center</unit\_type>//单位类型

<unit\_id>0</unit\_id>//单位ID

</LocalAddr>

<InNeT>//局域网配置

<NetAddr>//局域网总线服务地址，双网

<ip1>191.168.99.98</ip1>（总线服务本机地址）

<port1>9003</port1>

<ip2>10.2.48.123</ip2>（总线服务本机地址）

<port2>9005</port2>

</NetAddr>

<Config>

<receive\_thread\_num>8</receive\_thread\_num>//接收线程数（机器配置不够的情况下，减小该数值）

<send\_thread\_num>4</send\_thread\_num>//发送线程数，最大为4（机器配置不够的情况下，减小该数值）

<client\_links>100</client\_links>//接入的客户端数量最大值

<patrol\_duration>3</patrol\_duration>//断开检测间隔，单位是秒

</Config>

</InNeT>

<OutNet>//广域网配置，可配置单双网

<svrIP>191.168.99.98</svrIP>//本机服务ip

<svrPort>9001</svrPort>//本机服务port

<remote>//若需连入多个，则需要配置多个remote

<!--必须配置，所要连往的服务器ip和端口号 -->

<server\_ip> 191.168.99.98 </server\_ip>

<server\_port> 9002 </server\_port>

</remote>

</OutNet>

</NetRouter>

### 广域网配置详细说明



图2 总体结构图

上图是数据总线的总体结构图，以Net1和Net2为例。

* 1. 首先看LocalAddr局域地址的作用

假设Net1的局域地址为：

<LocalAddr>//本网的局域地址

<bureau\_code>8</bureau\_code>

<unit\_type>station</unit\_type>

<unit\_id>1</unit\_id>

</LocalAddr>

Net2的局域地址为：

<LocalAddr>//本网的局域地址

<bureau\_code>8</bureau\_code>

<unit\_type>station</unit\_type>

<unit\_id>2</unit\_id>

</LocalAddr>

则所有连接至Net1的应用程序发往8,1,2地址的数据，都会由Net2的NetRouter发往Net2局域网的应用程序。而反过来亦然。

* 1. 其次看OutNet广域网配置的作用

假设Net1的配置为：

<OutNet>//广域网配置，可配置单双网

<svrIP>191.168.99.98</svrIP>//本机服务ip

<svrPort>9001</svrPort>//本机服务port

<remote>

<!--必须配置，所要连往的服务器ip和端口号 -->

<server\_ip> 191.168.99.23 </server\_ip>

<server\_port> 9002 </server\_port>

</remote>

</OutNet>

Net2的配置为：

<OutNet>//广域网配置，可配置单双网

注意：IP、Port的数值与Net1中配置的remote中的IP、Port一致

<svrIP>191.168.99.23</svrIP>//本机服务ip

<svrPort>9002</svrPort>//本机服务port

<remote>

<!--必须配置，所要连往的服务器ip和端口号 -->

<server\_ip> 191.168.99.98 </server\_ip>

<server\_port> 9001 </server\_port>

</remote>

</OutNet>

则两个网的NetRouter可以互相连接和通信。若需多子网互联，需要加入相应个数的remote标签配置。

## log4cplus.properties

使用默认值。

## 双击NetRouter.exe启动总线服务

执行文件位于“NetRouter软件组\NetRouter”文件夹中。

# 应用开发

本节针对用户使用NetRouterClient接口库进行应用开发进行描述。具体分为配置文件描述、数据结构描述和接口描述。

## 配置文件描述

NetRouterClient的配置文件位于应用程序同目录下的config文件夹中（放在Debug目录下），有Configure.xml和log4cplus.properties两个。

### Configure.xml

如果不使用Configure.xml文件进行初始化，而是应用程序自己初始化的话，不需要对其进行配置，只需要保留这个文件即可(必须保留)，无需更改其配置。

<FCMP>

<server>//局域网总线服务地址，双网

<ip1>191.168.99.98</ip1>

<port1>9003</port1>

<ip2>191.168.99.98</ip2>

<port2>9005</port2>

</server>

<address>//逻辑地址标识

<!--必须配置，线路号-->

<bureau\_code>8</bureau\_code>

<!--必须配置，单位类型：center，station, station:0x01, center:0x02-->

<unit\_type>center</unit\_type> （这里填写数字2）

<!--必须配置，单位ID：调度台ID，站码，中心服务器的单位ID为0 -->

<unit\_id>0</unit\_id>

<!--必须配置，设备类型：svr，ddt，term，other, svr:0x01, ddr:0x02, term: 0x03, other: 0x04 -->

<dev\_type>svr</dev\_type>

<!--必须配置，设备ID -->

<dev\_id>3</dev\_id> （不同的机器配置不同的ID）

</address>

<in\_condition>//注册信息

<rec>

<protocol418\_condition>

<type\_func>0xff,0xff</type\_func>

</protocol418\_condition>

</rec>

</in\_condition>

<config>

</config>

</FCMP>

### log4cplus.properties

暂时使用默认配置。

## Java适配

### 数据结构描述

1. Address

每个通信客户端的逻辑通信地址设计如下：

Class Address{

1: byte bureauCode, //表示局码

2: byte unitType, //表示站或是中心

3: short unitId, //表示站或是中心的id

4: byte devType, //表示设备类型

5: short devId //表示设备id

}

1. SendMessage

Class SendMessage{

1: List<Address> destAddrs\_, //消息目的地址

2: String msg\_ //发送的消息

}

1. RecvMessage

Class RecvMessage{

1: Address srcAddrs\_, //消息源地址

2: String msg\_ //接收到的消息

}

### 接口描述

1. NetRouterClient类

用户程序通过这个类与本网的NetRouter进行交互。

|  |  |
| --- | --- |
| NetRouterClient（） | |
| 返回参数 |  |
| 参数列表 | String appName |
| 描述 | 构造函数，用此函数构造对象则需要对配置文件进行配置，不推荐。 |

|  |  |
| --- | --- |
| NetRouterClient（） | |
| 返回参数 |  |
| 参数列表 | * 1. String appName //应用的名称   2. String ip1 //网1服务的Ip地址   3. int port1 //网1服务的端口号   4. String ip2 //网2服务的ip地址   5. int port2 //网2服务的端口号   6. Address localAddr //本程序的逻辑通信地址   7. String reginfo //注册信息，可填空，默认注册全信息 |
| 描述 | 构造函数，用此函数构造对象则不需要对配置文件进行配置，推荐。  Reginfo的注册信息示例如下：  string reginfo =  "<in\_condition>\n"+  "<rec>\n"+  "<protocol418\_condition>\n"+  "<type\_func>0x0A,0x0B</type\_func>\n"+  "<type\_func>0x0A,0x0C</type\_func>\n"+  "</protocol418\_condition>\n"+  "</rec>\n"+  "</in\_condition>\n";  表示向NetRouter订阅类型为0A和子类型为0B、0C的数据。 |

|  |  |
| --- | --- |
| start（） | |
| 返回参数 | boolean |
| 参数列表 |  |
| 描述 | 开启通信服务，进行双网连接，成功返回true，失败返回false |

|  |  |
| --- | --- |
| sendMessage（） | |
| 返回参数 | boolean |
| 参数列表 | SendMessage sendMsg |
| 描述 | 发送消息，成功返回true，失败返回false |

|  |  |
| --- | --- |
| receiveMessage（） | |
| 返回参数 | boolean |
| 参数列表 | RecvMessage recvMsg |
| 描述 | 接收消息，成功返回true，失败返回false. |

|  |  |
| --- | --- |
| isNet1Connected（） | |
| 返回参数 | boolean |
| 参数列表 |  |
| 描述 | 检查网1是否连接，连接返回true，断开返回false. |

|  |  |
| --- | --- |
| isNet2Connected（） | |
| 返回参数 | boolean |
| 参数列表 |  |
| 描述 | 检查网2是否连接，连接返回true，断开返回false. |

|  |  |
| --- | --- |
| close（） | |
| 返回参数 | boolean |
| 参数列表 |  |
| 描述 | 断开网络连接，成功返回ture，失败返回false |

### 集成方法

在运行目录下放入config文件夹和NetRouterCppClient.dll。将NetRouterClient.jar集成到应用程序。

## C++适配

### 数据结构描述

1. NetAddress

每个通信客户端的逻辑通信地址设计如下：

Class NetAddress{

1: uinsigned char bureauCode, //表示局码

2: uinsigned char unitType, //表示站或是中心

3: uinsigned short unitId, //表示站或是中心的id

4: uinsigned char devType, //表示设备类型

5: uinsigned short devId //表示设备id

}

1. SentMessage

Class SentMessage{

1: vector<NetAddress> destAddrs\_, //消息目的地址

2: string msg\_ //发送的消息

}

1. RecvMessage

Class RecvMessage{

1: NetAddress srcAddrs\_, //消息源地址

2: string msg\_ //接收到的消息

}

### 接口描述

1. NetRouterClient类

用户程序通过这个类与本网的NetRouter进行交互。

|  |  |
| --- | --- |
| NetRouterClient（） | |
| 返回参数 |  |
| 参数列表 | string appName |
| 描述 | 构造函数，用此函数构造对象则需要对配置文件进行配置，不推荐。 |

|  |  |
| --- | --- |
| NetRouterClient（） | |
| 返回参数 |  |
| 参数列表 | 1. string appName //应用的名称 2. string ip1 //网1服务的Ip地址 3. int port1 //网1服务的端口号 4. string ip2 //网2服务的ip地址 5. int port2 //网2服务的端口号 6. NetAddress &localAddr //本程序的逻辑通信地址 7. string reginfo //注册信息，可不填，默认注册全信息 |
| 描述 | 构造函数，用此函数构造对象则不需要对配置文件进行配置，推荐。  Reginfo的注册信息示例如下：  string reginfo =  "<in\_condition>\n"  "<rec>\n"  "<protocol418\_condition>\n"  "<type\_func>0x0A,0x0B</type\_func>\n"  "<type\_func>0x0A,0x0C</type\_func>\n"  "</protocol418\_condition>\n"  "</rec>\n"  "</in\_condition>\n";  表示向NetRouter订阅类型为0A和子类型为0B、0C的数据。 |

|  |  |
| --- | --- |
| init（） | |
| 返回参数 | bool |
| 参数列表 |  |
| 描述 | 初始化，成功返回true，失败返回false |

|  |  |
| --- | --- |
| start（） | |
| 返回参数 | bool |
| 参数列表 |  |
| 描述 | 开启通信服务，进行双网连接，成功返回true，失败返回false |

|  |  |
| --- | --- |
| sendMessage（） | |
| 返回参数 | bool |
| 参数列表 | SentMessage &sendMsg |
| 描述 | 发送消息，成功返回true，失败返回false |

|  |  |
| --- | --- |
| receiveMessage（） | |
| 返回参数 | bool |
| 参数列表 | RecvMessage &recvMsg |
| 描述 | 接收消息，成功返回true，失败返回false. |

|  |  |
| --- | --- |
| isNet1Connected（） | |
| 返回参数 | bool |
| 参数列表 |  |
| 描述 | 检查网1是否连接，连接返回true，断开返回false. |

|  |  |
| --- | --- |
| isNet2Connected（） | |
| 返回参数 | bool |
| 参数列表 |  |
| 描述 | 检查网2是否连接，连接返回true，断开返回false. |

|  |  |
| --- | --- |
| close（） | |
| 返回参数 | bool |
| 参数列表 |  |
| 描述 | 断开网络连接，成功返回ture，失败返回false |

### 集成方法

在运行目录下放入config文件夹和NetRouterCppClient.dll。

## C#适配

由于C#调用C++动态库需要进行两种语言间参数的匹配，因此逻辑通信地址使用的是结构体。

### 数据结构描述

1. NETADDR

每个通信客户端的逻辑通信地址设计如下：

Struct NETADDR{

1: byte bureauCode, //表示局码

2: byte unitType, //表示站或是中心

3: UInt16 unitId, //表示站或是中心的id

4: byte devType, //表示设备类型

5: UInt16 devId //表示设备id

}

1. SENDMSG

Class SENDMSG{

1: List< NETADDR > destAddrs\_, //消息目的地址

2: string msg\_ //发送的消息

}

1. REVMSG

Class REVMSG{

1: NETADDR srcAddrs\_, //消息源地址

2: string msg\_ //接收到的消息

}

### 接口描述

1. NetRouterClientFactory类

简单工厂类，用户通过这类来生成具体的NetRouterClient对象。

|  |  |
| --- | --- |
| CreateNetRouterClient（） | |
| 返回参数 | INetRouterClient |
| 参数列表 | string appName |
| 描述 | 用此函数构造对象则需要对配置文件进行配置，不推荐。 |

|  |  |
| --- | --- |
| CreateNetRouterClient（） | |
| 返回参数 | INetRouterClient |
| 参数列表 | 1. string appName //应用的名称 2. string ip1 //网1服务的Ip地址 3. int port1 //网1服务的端口号 4. string ip2 //网2服务的ip地址 5. int port2 //网2服务的端口号 6. ref NETADDR localAddr //本程序的逻辑通信地址 7. string reginfo //注册信息，可不填，默认注册全信息 |
| 描述 | 用此函数构造对象则不需要对配置文件进行配置，推荐  Reginfo的注册信息示例如下：  string reginfo =  "<in\_condition>\n"+  "<rec>\n"+  "<protocol418\_condition>\n"+  "<type\_func>0x0A,0x0B</type\_func>\n"+  "<type\_func>0x0A,0x0C</type\_func>\n"+  "</protocol418\_condition>\n"+  "</rec>\n"+  "</in\_condition>\n";  表示向NetRouter订阅类型为0A和子类型为0B、0C的数据。 |

1. INetRouterClient接口

用户程序通过此接口类与本网的NetRouter进行交互。

|  |  |
| --- | --- |
| start（） | |
| 返回参数 | bool |
| 参数列表 |  |
| 描述 | 开启通信服务，进行双网连接，成功返回true，失败返回false |

|  |  |
| --- | --- |
| sendMessage（） | |
| 返回参数 | bool |
| 参数列表 | Ref SENDMSG sendMsg |
| 描述 | 发送消息，成功返回true，失败返回false |

|  |  |
| --- | --- |
| receiveMessage（） | |
| 返回参数 | bool |
| 参数列表 | Ref REVMSG recvMsg |
| 描述 | 接收消息，成功返回true，失败返回false. |

|  |  |
| --- | --- |
| isNet1Connected（） | |
| 返回参数 | bool |
| 参数列表 |  |
| 描述 | 检查网1是否连接，连接返回true，断开返回false. |

|  |  |
| --- | --- |
| isNet2Connected（） | |
| 返回参数 | bool |
| 参数列表 |  |
| 描述 | 检查网2是否连接，连接返回true，断开返回false. |

|  |  |
| --- | --- |
| close（） | |
| 返回参数 | bool |
| 参数列表 |  |
| 描述 | 断开网络连接，成功返回ture，失败返回false |

### 集成方法

在项目的Debug目录下放入config文件夹（config文件夹在“NetRouter软件组\NetRouterClient”文件夹中）。

NetRouterClient.dll是C#接口库，需要将其引入项目，NetRouterCppClient.dll为C++库，直接拷贝到Debug文件夹下便可。

### 引用配置

（1）需要将下面三个文件放在Debug目录下（文件处于“NetRouter软件组\NetRouterClient\CSharp”文件夹中）

NetRouterClient.dll

NetRouterCppClient.dll

NetRouterCppClient.lib

（2）工程项目中添加NetRouterClient.dll引用。