**C/C++ XMPP/Jabber 客户端类库对比/点评**



[CodingSir](https://blog.csdn.net/educast) 2014-06-16 11:22:58  5083  收藏 1

================ gloox ================ Ans. 老牌库，推荐 gloox是一个稳定功能完整的XMPP客户端库, 使用ANSI C++编写, 使用它编写标准客户非常容易, 并能将Jabber/XMPP功能集成到现有的应用程序中, gloox遵循GNU GPL许可协议, 支持商业许可. 特点: 1. 易于使用 2. 可扩展性好 3. 平台无关性(支持多种操作系统) 4. 完整

================

gloox

================

Ans. 老牌库，推荐

gloox是一个稳定功能完整的XMPP客户端库, 使用ANSI C++编写, 使用它编写标准客户非常容易, 并能将Jabber/XMPP功能集成到现有的应用程序中, gloox遵循GNU GPL许可协议, 支持商业许可.

特点:

1. 易于使用

2. 可扩展性好

3. 平台无关性(支持多种操作系统)

4. 完整的开发文档

5. 完全支持XMPP核心协议及几乎所有的XMPP IM协议

6. 支持TLS和SASL安全认证

7. 支持名册管理

8. 支持隐私名单

9. 支持SRV查找

10.支持服务器认证方式

11.支持xml:lang

================

(X)Iris

================

Ans. Qt的，慎用

Iris是基于Qt/C++库编写的XMPP/Jabber协议库,目前它仍然处于开发阶段，但已经支持了许多重要的功能。设计目标是全力支持 XMPP协议的正式草案, 库功能强大且易于使用的，由于大量使用Qt的结构和数据类型, 对于创建客户端,服务器和其他组件非常有用.

特点:

1. 支持XMPP核心协议包括:

1) DNS服务查询

2) SSL/TLS安全认证

3) SASL加密认证

5) 早期的jabber:iq:auth login

6) 资源绑定

7) 支持HTTP连接及Socket5,http代理

8) 高级对象流处理

2. 支持的XMPP-IM协议

1) 消息交换

2) 出席广播与接收

3) 名册管理

4) 订阅功能

3. JEP扩展

1) 版本时间要求

2) 服务发现

3) Jabber账号注册

4) 密码更改

5) 代理注册

6) VCards

7) 基本组聊天

8) S5B直接连接

9) 文件传输

开发计划:

1. 完成支持XMPP核心协议的基本服务

2. 确保XMPP的核心完全匹配草案

3. 编写XMPP的核心文档API

4. 撰写完整XMPP的即时通讯的API

5. 其他重要规格：x:data， MUC等等

================

(X)QXmpp

================

Ans. Qt的，慎用

QXmpp是一个跨平台的C++ XMPP客户端开发库, 它是基于Qt库开发的.

QXmpp是非常直观，易于使用, Qt是唯一的第三方依赖库, 用户需要一个有C ++和Qt的基础知识(Qt的信号与槽和数据类型)及开发经验。基础的TCP套接字和XMPP协议的RFC （ RFC3920和RFC3921 ）已被封装到类和函数 必要时可以扩展, 所以，用户不需要处理这些细节。

说明文档比较少, 没找到!!!

================

(X)oajabber

================

Ans. 很久没更新，单人开发，慎用

oajabber是一个支持XMPP/Jabber协议的C++库, 可扩展及灵活性好,它的目标是实现最完整的XMPP/Jabber协议的C++库, 这在很大程度上依赖于有oapr对线程和网络的处理及Xerces-C对XML的解析.

================

(X)iksemel

================

Ans. 接口太复杂，慎用

iksemel是一个为Jabber应用而设计的XML(可扩展标记语言)解析器库。

1. 高度便携，可在所有的POSIX环境和Windows(via mingw)环境下构建

2. 适用于嵌入式系统。代码小，高度模块化，可以削减不必要的部分。内存使用量低，解析器速度相当快。

3. 提供的SAX,DOM和XMPP协议分析器。

4. 支持TLS和SASL安全认证

5. 仅支持UTF-8编码

6. 完整的文档及示例

================

?loudmouth

================

Ans. C的，接口太多，慎用

loudmouth是一个用C/Glib编写的轻量级jabber客户端库, 易于使用和扩展, 可跨平台.

================

\*\*\* libstrophe

================

Ans. C的，有C++封装，可以考虑

libstrophe是一个XMPP协议的客户端和组件通信的C库。我们的目标是建立一个库，它是便携式的，可用的快捷，可靠，有据可查，并充分实现的XMPP规范

================

\*Libjingle

================

Ans. GTalk的库，可以考虑

　　Libjingle - Google Talk Voice及 P2P 的交互操作函数库

　　Libjingle是Google提供的C++组件集，它为Google Talk的点对点通讯与语音呼叫功能提供交互操作性。组件包包括了Jingle和Jingle-Audio的google实现的源代码，它们是XMPP标准的推荐扩展，目前试验版可用。

　　我们发布此代码是为了让消费者们在实时通讯工具的选择和操作性更多些。Google的源码发布基于Berkeley协议，也就是说你可以把它们合并到商业或非商业的软件中并发布。

　　此外关于Google Talk的互操作性，函数库里有许多通用组件如P2P栈，它可用来创建各样的通信和协作应用程序。我们希望看到更多组织能够使用这项技术建立创新的的应用程序。

　　下面是函数库各个组件的描述。你可以使用它们中的任一或是全部组件。

　　base -低级可移植的功能函数

　　p2p - p2p栈，包括了与XMPP挂钩的客户端和基础p2p函数。

　　session - 电话呼叫指令以及可变点对点流

　　third\_party - 用于其他功能的非google组件

　　xmllite -XML解析器

　　xmpp - XMPP engine

===============

\*libpurple

===============

Ans. Pidgin的库，推荐

libpurple is intended to be the core of an IM program. When using libpurple, you'll basically be writing a UI for this core chunk of code. Pidgin is a GTK+ frontend to libpurple, Finch is an ncurses frontend, and Adium is a Cocoa frontend.

# [C/C++ XMPP/Jabber 客户端类库对比/点评 （转）](https://www.cnblogs.com/happykoukou/p/5803005.html)

### 原文转自 <http://blog.csdn.net/educast/article/details/31359835>

### 1、gloox

Ans. 老牌库，推荐

gloox是一个稳定功能完整的XMPP客户端库, 使用ANSI C++编写, 使用它编写标准客户非常容易, 并能将Jabber/XMPP功能集成到现有的应用程序中, gloox遵循GNU GPL许可协议, 支持商业许可.

特点:

1. 易于使用

2. 可扩展性好

3. 平台无关性(支持多种操作系统)

4. 完整的开发文档

5. 完全支持XMPP核心协议及几乎所有的XMPP IM协议

6. 支持TLS和SASL安全认证

7. 支持名册管理

8. 支持隐私名单

9. 支持SRV查找

10.支持服务器认证方式

11.支持xml:lang

### 2、Iris

Ans. Qt的，慎用

Iris是基于Qt/C++库编写的XMPP/Jabber协议库,目前它仍然处于开发阶段，但已经支持了许多重要的功能。设计目标是全力支持 XMPP协议的正式草案, 库功能强大且易于使用的，由于大量使用Qt的结构和数据类型, 对于创建客户端,服务器和其他组件非常有用.

特点:

1. 支持XMPP核心协议包括:

1) DNS服务查询

2) SSL/TLS安全认证

3) SASL加密认证

5) 早期的jabber:iq:auth login

6) 资源绑定

7) 支持HTTP连接及Socket5,http代理

8) 高级对象流处理

2. 支持的XMPP-IM协议

1) 消息交换

2) 出席广播与接收

3) 名册管理

4) 订阅功能

3. JEP扩展

1) 版本时间要求

2) 服务发现

3) Jabber账号注册

4) 密码更改

5) 代理注册

6) VCards

7) 基本组聊天

8) S5B直接连接

9) 文件传输

开发计划:

1. 完成支持XMPP核心协议的基本服务

2. 确保XMPP的核心完全匹配草案

3. 编写XMPP的核心文档API

4. 撰写完整XMPP的即时通讯的API

5. 其他重要规格：x:data， MUC等等

### 3、QXmpp

Ans. Qt的，慎用

QXmpp是一个跨平台的C++ XMPP客户端开发库, 它是基于Qt库开发的.

QXmpp是非常直观，易于使用, Qt是唯一的第三方依赖库, 用户需要一个有C ++和Qt的基础知识(Qt的信号与槽和数据类型)及开发经验。基础的TCP套接字和XMPP协议的RFC （ RFC3920和RFC3921 ）已被封装到类和函数 必要时可以扩展, 所以，用户不需要处理这些细节。

说明文档比较少, 没找到!!!

### 4、oajabber

Ans. 很久没更新，单人开发，慎用

oajabber是一个支持XMPP/Jabber协议的C++库, 可扩展及灵活性好,它的目标是实现最完整的XMPP/Jabber协议的C++库, 这在很大程度上依赖于有oapr对线程和网络的处理及Xerces-C对XML的解析.

### 5、iksemel

Ans. 接口太复杂，慎用

iksemel是一个为Jabber应用而设计的XML(可扩展标记语言)解析器库。

1. 高度便携，可在所有的POSIX环境和Windows(via mingw)环境下构建

2. 适用于嵌入式系统。代码小，高度模块化，可以削减不必要的部分。内存使用量低，解析器速度相当快。

3. 提供的SAX,DOM和XMPP协议分析器。

4. 支持TLS和SASL安全认证

5. 仅支持UTF-8编码

6. 完整的文档及示例

### 6、loudmouth

Ans. C的，接口太多，慎用

loudmouth是一个用C/Glib编写的轻量级jabber客户端库, 易于使用和扩展, 可跨平台.

### 7、libstrophe

Ans. C的，有C++封装，可以考虑

libstrophe是一个XMPP协议的客户端和组件通信的C库。我们的目标是建立一个库，它是便携式的，可用的快捷，可靠，有据可查，并充分实现的XMPP规范

### 8、Libjingle

Ans. GTalk的库，可以考虑

　　Libjingle - Google Talk Voice及 P2P 的交互操作函数库

　　Libjingle是Google提供的C++组件集，它为Google Talk的点对点通讯与语音呼叫功能提供交互操作性。组件包包括了Jingle和Jingle-Audio的google实现的源代码，它们是XMPP标准的推荐扩展，目前试验版可用。

　　我们发布此代码是为了让消费者们在实时通讯工具的选择和操作性更多些。Google的源码发布基于Berkeley协议，也就是说你可以把它们合并到商业或非商业的软件中并发布。

　　此外关于Google Talk的互操作性，函数库里有许多通用组件如P2P栈，它可用来创建各样的通信和协作应用程序。我们希望看到更多组织能够使用这项技术建立创新的的应用程序。

　　下面是函数库各个组件的描述。你可以使用它们中的任一或是全部组件。

　　base -低级可移植的功能函数

　　p2p - p2p栈，包括了与XMPP挂钩的客户端和基础p2p函数。

　　session - 电话呼叫指令以及可变点对点流

　　third\_party - 用于其他功能的非google组件

　　xmllite -XML解析器

　　xmpp - XMPP engine

### 9、libpurple

Ans. Pidgin的库，推荐

libpurple is intended to be the core of an IM program. When using libpurple, you'll basically be writing a UI for this core chunk of code. Pidgin is a GTK+ frontend to libpurple, Finch is an ncurses frontend, and Adium is a Cocoa frontend.

分类: [音视频](https://www.cnblogs.com/happykoukou/category/852863.html)

【转】常见XMPP客户端c/c++开发库

[](http://www.360doc.cn/userhome.aspx?userid=3700464)

[techres](http://www.360doc.cn/userhome.aspx?userid=3700464)

阅1851转32014-02-25

[分享保存](javascript:void(0);)

 XMPP客户端开发库

gloox

gloox是一个稳定功能完整的XMPP客户端库, 使用ANSI C++编写, 使用它编写标准客户非常容易, 并能将Jabber/XMPP功能集成到现有的应用程序中, gloox遵循GNU GPL许可协议, 支持商业许可.

特点:  
    1. 易于使用  
    2. 可扩展性好  
    3. 平台无关性(支持多种**[操作系统](http://www.360doc.cn/outlink.html?url=http://garylee.blog.51cto.com/3280377/\\%27" \l "\\'%22)**)  
    4. 完整的开发文档  
    5. 完全支持XMPP核心协议及几乎所有的XMPP IM协议  
    6. 支持TLS和SASL安全认证  
    7. 支持名册管理  
    8. 支持隐私名单  
    9. 支持SRV查找  
    10.支持服务器认证方式  
    11.支持xml:lang

Iris

Iris是基于Qt/C++库编写的XMPP/Jabber协议库,目前它仍然处于开发阶段，但已经支持了许多重要的功能。设计目标是全力支持XMPP协议的正式草案, 库功能强大且易于使用的，由于大量使用Qt的结构和数据类型, 对于创建客户端,服务器和**[其他](http://www.360doc.cn/outlink.html?url=http://garylee.blog.51cto.com/3280377/\\%27" \l "\\'%22)**组件非常有用.

特点:  
    1. 支持XMPP核心协议包括:  
  1) DNS服务查询  
  2) SSL/TLS安全认证  
  3) SASL加密认证  
  5) 早期的jabber:iq:auth login  
   6) 资源绑定  
  7) 支持HTTP连接及Socket5,http代理  
   8) 高级对象流处理  
    2. 支持的XMPP-IM协议  
  1) 消息交换  
  2) 出席广播与接收  
  3) 名册管理  
  4) 订阅功能  
    3. JEP扩展  
  1) 版本时间要求  
  2) 服务发现  
  3) Jabber账号注册  
  4) 密码更改  
  5) 代理注册  
  6) VCards  
  7) 基本组聊天  
  8) S5B直接连接  
  9) 文件传输

开发计划:  
    1. 完成支持XMPP核心协议的基本服务  
    2. 确保XMPP的核心完全匹配草案  
    3. 编写XMPP的核心文档API  
    4. 撰写完整XMPP的即时通讯的API  
    5. 其他重要规格：x:data， MUC等等

QXmpp

QXmpp是一个跨平台的C++ XMPP客户端开发库, 它是基于Qt库开发的.  
QXmpp是非常直观，易于使用, Qt是唯一的第三方依赖库, 用户需要一个有C ++和Qt的基础知识(Qt的信号与槽和数据类型)及开发经验。基础的TCP套接字和XMPP协议的RFC （ RFC3920和RFC3921 ）已被封装到类和函数 必要时可以扩展, 所以，用户不需要处理这些细节。

说明文档比较少, 没找到!!!

oajabber

oajabber是一个支持XMPP/Jabber协议的C++库, 可扩展及灵活性好,它的目标是实现最完整的XMPP/Jabber协议的C++库, 这在很大程度上依赖于有oapr对线程和网络的处理及Xerces-C对XML的解析.

iksemel

iksemel是一个为Jabber应用而设计的XML(可扩展标记语言)解析器库。  
   
    1. 高度便携，可在所有的POSIX环境和**[Windows](http://www.360doc.cn/outlink.html?url=http://garylee.blog.51cto.com/3280377/\\%27" \l "\\'%22)**(via mingw)环境下构建  
    2. 适用于嵌入式系统。代码小，高度模块化，可以削减不必要的部分。内存使用量低，解析器速度相当快。  
    3. 提供的SAX,DOM和XMPP协议分析器。  
    4. 支持TLS和SASL安全认证  
    5. 仅支持UTF-8编码  
    6. 完整的文档及示例

loudmouth

loudmouth是一个用C/Glib编写的轻量级jabber客户端库, 易于使用和扩展, 可跨平台.

libstrophe  
libstrophe是一个XMPP协议的客户端和组件通信的C库。我们的目标是建立一个库，它是便携式的，可用的快捷，可靠，有据可查，并充分实现的XMPP规范

libpurple

是Pidgin 的底层IM协议的封装库，其中就包括XMPP协议的封装



XMPP客户端开发库

2011-06-30 17:34:26

标签：[休闲](http://www.360doc.cn/outlink.html?url=http://blog.51cto.com/tag-%E4%BC%91%E9%97%B2.html) [XMPP](http://www.360doc.cn/outlink.html?url=http://blog.51cto.com/tag-XMPP.html) [职场](http://www.360doc.cn/outlink.html?url=http://blog.51cto.com/tag-%E8%81%8C%E5%9C%BA.html)

gloox

gloox是一个稳定功能完整的XMPP客户端库, 使用ANSI C++编写, 使用它编写标准客户非常容易, 并能将Jabber/XMPP功能集成到现有的应用程序中, gloox遵循GNU GPL许可协议, 支持商业许可.

特点:  
    1. 易于使用  
    2. 可扩展性好  
    3. 平台无关性(支持多种**[操作系统](http://www.360doc.cn/outlink.html?url=http://garylee.blog.51cto.com/3280377/\\%27" \l "\\'%22)**)  
    4. 完整的开发文档  
    5. 完全支持XMPP核心协议及几乎所有的XMPP IM协议  
    6. 支持TLS和SASL安全认证  
    7. 支持名册管理  
    8. 支持隐私名单  
    9. 支持SRV查找  
    10.支持服务器认证方式  
    11.支持xml:lang

Iris

Iris是基于Qt/C++库编写的XMPP/Jabber协议库,目前它仍然处于开发阶段，但已经支持了许多重要的功能。设计目标是全力支持XMPP协议的正式草案, 库功能强大且易于使用的，由于大量使用Qt的结构和数据类型, 对于创建客户端,服务器和**[其他](http://www.360doc.cn/outlink.html?url=http://garylee.blog.51cto.com/3280377/\\%27" \l "\\'%22)**组件非常有用.

特点:  
    1. 支持XMPP核心协议包括:  
  1) DNS服务查询  
  2) SSL/TLS安全认证  
  3) SASL加密认证  
  5) 早期的jabber:iq:auth login  
   6) 资源绑定  
  7) 支持HTTP连接及Socket5,http代理  
   8) 高级对象流处理  
    2. 支持的XMPP-IM协议  
  1) 消息交换  
  2) 出席广播与接收  
  3) 名册管理  
  4) 订阅功能  
    3. JEP扩展  
  1) 版本时间要求  
  2) 服务发现  
  3) Jabber账号注册  
  4) 密码更改  
  5) 代理注册  
  6) VCards  
  7) 基本组聊天  
  8) S5B直接连接  
  9) 文件传输

开发计划:  
    1. 完成支持XMPP核心协议的基本服务  
    2. 确保XMPP的核心完全匹配草案  
    3. 编写XMPP的核心文档API  
    4. 撰写完整XMPP的即时通讯的API  
    5. 其他重要规格：x:data， MUC等等

QXmpp

QXmpp是一个跨平台的C++ XMPP客户端开发库, 它是基于Qt库开发的.  
QXmpp是非常直观，易于使用, Qt是唯一的第三方依赖库, 用户需要一个有C ++和Qt的基础知识(Qt的信号与槽和数据类型)及开发经验。基础的TCP套接字和XMPP协议的RFC （ RFC3920和RFC3921 ）已被封装到类和函数 必要时可以扩展, 所以，用户不需要处理这些细节。

说明文档比较少, 没找到!!!

oajabber

oajabber是一个支持XMPP/Jabber协议的C++库, 可扩展及灵活性好,它的目标是实现最完整的XMPP/Jabber协议的C++库, 这在很大程度上依赖于有oapr对线程和网络的处理及Xerces-C对XML的解析.

iksemel

iksemel是一个为Jabber应用而设计的XML(可扩展标记语言)解析器库。  
   
    1. 高度便携，可在所有的POSIX环境和**[Windows](http://www.360doc.cn/outlink.html?url=http://garylee.blog.51cto.com/3280377/\\%27" \l "\\'%22)**(via mingw)环境下构建  
    2. 适用于嵌入式系统。代码小，高度模块化，可以削减不必要的部分。内存使用量低，解析器速度相当快。  
    3. 提供的SAX,DOM和XMPP协议分析器。  
    4. 支持TLS和SASL安全认证  
    5. 仅支持UTF-8编码  
    6. 完整的文档及示例

loudmouth

loudmouth是一个用C/Glib编写的轻量级jabber客户端库, 易于使用和扩展, 可跨平台.

libstrophe  
libstrophe是一个XMPP协议的客户端和组件通信的C库。我们的目标是建立一个库，它是便携式的，可用的快捷，可靠，有据可查，并充分实现的XMPP规范

libpurple

是Pidgin 的底层IM协议的封装库，其中就包括XMPP协议的封装



XMPP客户端开发库

2011-06-30 17:34:26

标签：[休闲](http://www.360doc.cn/outlink.html?url=http://blog.51cto.com/tag-%E4%BC%91%E9%97%B2.html) [XMPP](http://www.360doc.cn/outlink.html?url=http://blog.51cto.com/tag-XMPP.html) [职场](http://www.360doc.cn/outlink.html?url=http://blog.51cto.com/tag-%E8%81%8C%E5%9C%BA.html)

gloox

gloox是一个稳定功能完整的XMPP客户端库, 使用ANSI C++编写, 使用它编写标准客户非常容易, 并能将Jabber/XMPP功能集成到现有的应用程序中, gloox遵循GNU GPL许可协议, 支持商业许可.

特点:  
    1. 易于使用  
    2. 可扩展性好  
    3. 平台无关性(支持多种**[操作系统](http://www.360doc.cn/outlink.html?url=http://garylee.blog.51cto.com/3280377/\\%27" \l "\\'%22)**)  
    4. 完整的开发文档  
    5. 完全支持XMPP核心协议及几乎所有的XMPP IM协议  
    6. 支持TLS和SASL安全认证  
    7. 支持名册管理  
    8. 支持隐私名单  
    9. 支持SRV查找  
    10.支持服务器认证方式  
    11.支持xml:lang

Iris

Iris是基于Qt/C++库编写的XMPP/Jabber协议库,目前它仍然处于开发阶段，但已经支持了许多重要的功能。设计目标是全力支持XMPP协议的正式草案, 库功能强大且易于使用的，由于大量使用Qt的结构和数据类型, 对于创建客户端,服务器和**[其他](http://www.360doc.cn/outlink.html?url=http://garylee.blog.51cto.com/3280377/\\%27" \l "\\'%22)**组件非常有用.

特点:  
    1. 支持XMPP核心协议包括:  
  1) DNS服务查询  
  2) SSL/TLS安全认证  
  3) SASL加密认证  
  5) 早期的jabber:iq:auth login  
   6) 资源绑定  
  7) 支持HTTP连接及Socket5,http代理  
   8) 高级对象流处理  
    2. 支持的XMPP-IM协议  
  1) 消息交换  
  2) 出席广播与接收  
  3) 名册管理  
  4) 订阅功能  
    3. JEP扩展  
  1) 版本时间要求  
  2) 服务发现  
  3) Jabber账号注册  
  4) 密码更改  
  5) 代理注册  
  6) VCards  
  7) 基本组聊天  
  8) S5B直接连接  
  9) 文件传输

开发计划:  
    1. 完成支持XMPP核心协议的基本服务  
    2. 确保XMPP的核心完全匹配草案  
    3. 编写XMPP的核心文档API  
    4. 撰写完整XMPP的即时通讯的API  
    5. 其他重要规格：x:data， MUC等等

QXmpp

QXmpp是一个跨平台的C++ XMPP客户端开发库, 它是基于Qt库开发的.  
QXmpp是非常直观，易于使用, Qt是唯一的第三方依赖库, 用户需要一个有C ++和Qt的基础知识(Qt的信号与槽和数据类型)及开发经验。基础的TCP套接字和XMPP协议的RFC （ RFC3920和RFC3921 ）已被封装到类和函数 必要时可以扩展, 所以，用户不需要处理这些细节。

说明文档比较少, 没找到!!!

oajabber

oajabber是一个支持XMPP/Jabber协议的C++库, 可扩展及灵活性好,它的目标是实现最完整的XMPP/Jabber协议的C++库, 这在很大程度上依赖于有oapr对线程和网络的处理及Xerces-C对XML的解析.

iksemel

iksemel是一个为Jabber应用而设计的XML(可扩展标记语言)解析器库。  
   
    1. 高度便携，可在所有的POSIX环境和**[Windows](http://www.360doc.cn/outlink.html?url=http://garylee.blog.51cto.com/3280377/\\%27" \l "\\'%22)**(via mingw)环境下构建  
    2. 适用于嵌入式系统。代码小，高度模块化，可以削减不必要的部分。内存使用量低，解析器速度相当快。  
    3. 提供的SAX,DOM和XMPP协议分析器。  
    4. 支持TLS和SASL安全认证  
    5. 仅支持UTF-8编码  
    6. 完整的文档及示例

loudmouth

loudmouth是一个用C/Glib编写的轻量级jabber客户端库, 易于使用和扩展, 可跨平台.

libstrophe  
libstrophe是一个XMPP协议的客户端和组件通信的C库。我们的目标是建立一个库，它是便携式的，可用的快捷，可靠，有据可查，并充分实现的XMPP规范

libpurple

是Pidgin 的底层IM协议的封装库，其中就包括XMPP协议的封装



XMPP客户端开发库

2011-06-30 17:34:26

标签：[休闲](http://www.360doc.cn/outlink.html?url=http://blog.51cto.com/tag-%E4%BC%91%E9%97%B2.html) [XMPP](http://www.360doc.cn/outlink.html?url=http://blog.51cto.com/tag-XMPP.html) [职场](http://www.360doc.cn/outlink.html?url=http://blog.51cto.com/tag-%E8%81%8C%E5%9C%BA.html)

gloox

gloox是一个稳定功能完整的XMPP客户端库, 使用ANSI C++编写, 使用它编写标准客户非常容易, 并能将Jabber/XMPP功能集成到现有的应用程序中, gloox遵循GNU GPL许可协议, 支持商业许可.

特点:  
    1. 易于使用  
    2. 可扩展性好  
    3. 平台无关性(支持多种**[操作系统](http://www.360doc.cn/outlink.html?url=http://garylee.blog.51cto.com/3280377/\\%27" \l "\\'%22)**)  
    4. 完整的开发文档  
    5. 完全支持XMPP核心协议及几乎所有的XMPP IM协议  
    6. 支持TLS和SASL安全认证  
    7. 支持名册管理  
    8. 支持隐私名单  
    9. 支持SRV查找  
    10.支持服务器认证方式  
    11.支持xml:lang

Iris

Iris是基于Qt/C++库编写的XMPP/Jabber协议库,目前它仍然处于开发阶段，但已经支持了许多重要的功能。设计目标是全力支持XMPP协议的正式草案, 库功能强大且易于使用的，由于大量使用Qt的结构和数据类型, 对于创建客户端,服务器和**[其他](http://www.360doc.cn/outlink.html?url=http://garylee.blog.51cto.com/3280377/\\%27" \l "\\'%22)**组件非常有用.

特点:  
    1. 支持XMPP核心协议包括:  
  1) DNS服务查询  
  2) SSL/TLS安全认证  
  3) SASL加密认证  
  5) 早期的jabber:iq:auth login  
   6) 资源绑定  
  7) 支持HTTP连接及Socket5,http代理  
   8) 高级对象流处理  
    2. 支持的XMPP-IM协议  
  1) 消息交换  
  2) 出席广播与接收  
  3) 名册管理  
  4) 订阅功能  
    3. JEP扩展  
  1) 版本时间要求  
  2) 服务发现  
  3) Jabber账号注册  
  4) 密码更改  
  5) 代理注册  
  6) VCards  
  7) 基本组聊天  
  8) S5B直接连接  
  9) 文件传输

开发计划:  
    1. 完成支持XMPP核心协议的基本服务  
    2. 确保XMPP的核心完全匹配草案  
    3. 编写XMPP的核心文档API  
    4. 撰写完整XMPP的即时通讯的API  
    5. 其他重要规格：x:data， MUC等等

QXmpp

QXmpp是一个跨平台的C++ XMPP客户端开发库, 它是基于Qt库开发的.  
QXmpp是非常直观，易于使用, Qt是唯一的第三方依赖库, 用户需要一个有C ++和Qt的基础知识(Qt的信号与槽和数据类型)及开发经验。基础的TCP套接字和XMPP协议的RFC （ RFC3920和RFC3921 ）已被封装到类和函数 必要时可以扩展, 所以，用户不需要处理这些细节。

说明文档比较少, 没找到!!!

oajabber

oajabber是一个支持XMPP/Jabber协议的C++库, 可扩展及灵活性好,它的目标是实现最完整的XMPP/Jabber协议的C++库, 这在很大程度上依赖于有oapr对线程和网络的处理及Xerces-C对XML的解析.

iksemel

iksemel是一个为Jabber应用而设计的XML(可扩展标记语言)解析器库。  
   
    1. 高度便携，可在所有的POSIX环境和**[Windows](http://www.360doc.cn/outlink.html?url=http://garylee.blog.51cto.com/3280377/\\%27" \l "\\'%22)**(via mingw)环境下构建  
    2. 适用于嵌入式系统。代码小，高度模块化，可以削减不必要的部分。内存使用量低，解析器速度相当快。  
    3. 提供的SAX,DOM和XMPP协议分析器。  
    4. 支持TLS和SASL安全认证  
    5. 仅支持UTF-8编码  
    6. 完整的文档及示例

loudmouth

loudmouth是一个用C/Glib编写的轻量级jabber客户端库, 易于使用和扩展, 可跨平台.

libstrophe  
libstrophe是一个XMPP协议的客户端和组件通信的C库。我们的目标是建立一个库，它是便携式的，可用的快捷，可靠，有据可查，并充分实现的XMPP规范

libpurple

是Pidgin 的底层IM协议的封装库，其中就包括XMPP协议的封装

# [agsXMPP .NET的XMPP协议包](https://www.oschina.net/p/agsxmpp)

收藏36

[评论1](https://www.oschina.net/p/agsxmpp#comments)

分享

[纠错](https://www.oschina.net/p/agsxmpp/edit)

授权协议[GPL](http://www.oschina.net/question/12_2826)

开发语言[C#](https://www.oschina.net/project/lang/194/csharp) [.NET](https://www.oschina.net/project/lang/20/dotnet)

操作系统Windows

软件类型开源软件

所属分类[程序开发](https://www.oschina.net/project/tag/12/development)、 [网络工具包](https://www.oschina.net/project/tag/145/networklib)

开源组织无

地区不详

投 递 者[红薯](https://my.oschina.net/javayou" \t "_blank)

适用人群未知

收录时间2009-07-21

[使用agsXMPP查询聊天室内联系人](https://my.oschina.net/yunfound/blog/134204)

[agsXMPP + Openfore 即时通讯开发（一）【agsXMPP 连接 Openfire】](https://my.oschina.net/knife2013/blog/681030)

[agsXMPP + Openfire 即时通讯开发（二） 【文件传输】](https://my.oschina.net/knife2013/blog/681064)

[C#连接基于Java开发IM——Openfire](https://my.oschina.net/u/4311024/blog/3643290)

[[2008年10月15日] RichEdit富文本聊天控件 SDK，采用openfire xmpp实现即时通讯](https://my.oschina.net/zencodex/blog/4682256)

[C#开源项目大全](https://my.oschina.net/u/4319585/blog/3870235)

[软件首页](https://www.oschina.net/home/login?goto_page=https%3A%2F%2Fwww.oschina.net%2Fp%2Fagsxmpp" \o "agsXMPP 网站" \t "_blank)

[软件文档](https://www.oschina.net/home/login?goto_page=https%3A%2F%2Fwww.oschina.net%2Fp%2Fagsxmpp" \o "agsXMPP 文档" \t "_blank)

[官方下载](https://www.oschina.net/home/login?goto_page=https%3A%2F%2Fwww.oschina.net%2Fp%2Fagsxmpp" \o "agsXMPP 下载" \t "_blank)

* **概览**
* 资讯
* 博客
* 问答

### 软件简介

agsXMPP致力于创建一个轻量、快速的跨平台类库，用于XMPP协议。

通过下面的三项技术，agsXMPP达到了这个目标。

1. 异步套接字
2. 与工厂模式结合的快速XML解析器
3. 自有的轻量级XML Dom，作为所有agsXMPP协议类的基础

示例代码：

XmppClientConnection xmpp = new XmppClientConnection("jabber.org");  
xmpp.Open("myusername", "mysecret");  
xmpp.OnLogin += delegate(object o) {  
      xmpp.Send(new Message("test@jabber.org", MessageType.chat, "Hello, how are you?"));  
};

**.Net开源IM项目：agsXMPP（C#）**



[devie](https://blog.csdn.net/devie) 2012-04-24 15:25:28  7673  收藏

分类专栏： [即时通讯](https://blog.csdn.net/devie/category_1132583.html) 文章标签： [.net](https://so.csdn.net/so/search/s.do?q=.net&t=blog&o=vip&s=&l=&f=&viparticle=) [c#](https://www.csdn.net/tags/MtTakg1sNDQ5LWJsb2cO0O0O.html) [im](https://so.csdn.net/so/search/s.do?q=im&t=blog&o=vip&s=&l=&f=&viparticle=) [xmpp](https://www.csdn.net/tags/MtjaMg2sNzA5NjQtYmxvZwO0O0OO0O0O.html) [string](https://www.csdn.net/tags/MtTaEg0sMDc0OTEtYmxvZwO0O0OO0O0O.html) [服务器](https://www.csdn.net/tags/MtTaEg0sNDcxOTgtYmxvZwO0O0OO0O0O.html)

[[](https://blog.csdn.net/devie/category_1132583.html)即时通讯](https://blog.csdn.net/devie/category_1132583.html)专栏收录该内容

0 篇文章0 订阅

订阅专栏

先传个实例：  
  
using System;  
using System.Text;  
using System.Threading;  
  
using agsXMPP;  
using agsXMPP.protocol;  
using agsXMPP.protocol.client;  
  
namespace ConsoleClient  
{  
    class Program  
    {  
        static bool \_bWait;  
          
        static void Main(string[] args)  
        {  
            XmppClientConnection xmppCon = new XmppClientConnection();  
  
            Console.Title = "Console Client";  
  
            // read the jid from the console  
            PrintHelp("Enter you Jid ( [user@server.com](mailto:user@server.com) ): ");  
            Jid jid = new Jid(Console.ReadLine());  
  
            PrintHelp(String.Format("Enter password for '{0}': ", jid.ToString()));  
  
            xmppCon.Password = Console.ReadLine();  
            xmppCon.Username = jid.User;  
            xmppCon.Server = jid.Server;  
            xmppCon.AutoAgents = false;  
            xmppCon.AutoPresence = true;  
            xmppCon.AutoRoster = true;  
            xmppCon.AutoResolveConnectServer = true;  
  
            // Connect to the server now   
            // !!! this is asynchronous !!!  
            try  
            {  
                xmppCon.OnRosterStart += new ObjectHandler(xmppCon\_OnRosterStart);  
                xmppCon.OnRosterItem += new XmppClientConnection.RosterHandler(xmppCon\_OnRosterItem);  
                xmppCon.OnRosterEnd += new ObjectHandler(xmppCon\_OnRosterEnd);  
                xmppCon.OnPresence += new PresenceHandler(xmppCon\_OnPresence);  
                xmppCon.OnMessage += new MessageHandler(xmppCon\_OnMessage);  
                xmppCon.OnLogin += new ObjectHandler(xmppCon\_OnLogin);  
  
                xmppCon.Open();  
  
            }  
            catch (Exception e)  
            {  
                Console.WriteLine(e.Message);  
            }  
  
            Wait("Login to server, please wait");  
              
            PrintCommands();  
  
            bool bQuit = false;  
  
            while (!bQuit)  
            {  
                string command = Console.ReadLine();  
                string[] commands = command.Split(' ');  
  
                switch (commands[0].ToLower())  
                {  
                    case "help":  
                        PrintCommands();  
                        break;  
                    case "quit":  
                        bQuit = true;  
                        break;  
                    case "msg":  
                        string msg = command.Substring(command.IndexOf(commands[2]));  
                        xmppCon.Send(new Message(new Jid(commands[1]), MessageType.chat, msg));  
                        break;  
                    case "status":  
                        switch (commands[1])  
                        {  
                            case "online":  
                                xmppCon.Show = ShowType.NONE;  
                                break;  
                            case "away":  
                                xmppCon.Show = ShowType.away;  
                                break;  
                            case "xa":  
                                xmppCon.Show = ShowType.xa;  
                                break;  
                            case "chat":  
                                xmppCon.Show = ShowType.chat;  
                                break;  
                        }  
                        string status = command.Substring(command.IndexOf(commands[2]));  
                        xmppCon.Status = status;  
                        xmppCon.SendMyPresence();  
                        break;  
                }  
            }  
  
            // close connection  
            xmppCon.Close();  
        }  
  
        private static void PrintCommands()  
        {  
            PrintHelp("You are logged in to the server now.");  
            PrintHelp("");  
            PrintHelp("Available commands are:");  
            PrintHelp("msg toJid text");  
            PrintHelp("status show{online, away, xa, chat} status");  
            PrintHelp("help");  
            PrintHelp("quit");  
            PrintHelp("");  
            PrintHelp("Examples:");  
            PrintHelp("msg  [test@server.com](mailto:test@server.com)  Hello World");  
            PrintHelp("msg  [test@server.com](mailto:test@server.com) /Office Hello World");  
            PrintHelp("status chat free for chat");  
            PrintHelp("");  
        }  
  
        private static void Wait(string statusMessage)  
        {  
            int i = 0;  
            \_bWait = true;  
  
            while (\_bWait)  
            {  
                i++;  
                if (i == 60)  
                    \_bWait = false;  
  
                Thread.Sleep(500);  
            }  
        }  
  
        static void xmppCon\_OnLogin(object sender)  
        {  
            Console.WriteLine();  
            PrintEvent("Logged in to server");  
        }  
  
        static void xmppCon\_OnRosterEnd(object sender)  
        {  
            \_bWait = false;  
            Console.WriteLine();  
            PrintInfo("All contacts received");  
        }  
  
        static void xmppCon\_OnRosterItem(object sender, agsXMPP.protocol.iq.roster.RosterItem item)  
        {  
            PrintInfo(String.Format("Got contact: {0}", item.Jid));  
        }  
  
        static void xmppCon\_OnRosterStart(object sender)  
        {  
            PrintEvent("Getting contacts now");  
        }  
  
        static void xmppCon\_OnPresence(object sender, Presence pres)  
        {  
            PrintInfo(String.Format("Got presence from: {0}", pres.From.ToString()));  
            PrintInfo(String.Format("type: {0}", pres.Type.ToString()));  
            PrintInfo(String.Format("status: {0}", pres.Status));  
            PrintInfo("");  
        }  
  
        static void xmppCon\_OnMessage(object sender, Message msg)  
        {  
  
            if (msg.Body != null)  
            {  
                PrintEvent(String.Format("Got message from: {0}", msg.From.ToString()));  
                PrintEvent("message: " + msg.Body);  
                PrintInfo("");  
            }  
        }        
  
        static void PrintEvent(string msg)  
        {  
            ConsoleColor current = Console.BackgroundColor;  
  
            Console.BackgroundColor = ConsoleColor.DarkGreen;  
            Console.WriteLine(msg);  
  
            Console.BackgroundColor = current;  
        }  
  
        static void PrintInfo(string msg)  
        {  
            ConsoleColor current = Console.BackgroundColor;  
  
            Console.BackgroundColor = ConsoleColor.Red;  
            Console.WriteLine(msg);  
  
            Console.BackgroundColor = current;  
        }  
  
        static void PrintHelp(string msg)  
        {  
            ConsoleColor current = Console.ForegroundColor;  
  
            Console.ForegroundColor = ConsoleColor.Yellow;  
            Console.WriteLine(msg);  
  
            Console.ForegroundColor = current;  
        }  
    }  
}  
  
Jabber Software：Jabber-NET、agsXMPP与Wilefire   
本篇介绍两个使用.NET技术，确切的说是使用C#写的Jabber Code Libraries – Jabber.NET、agsXMPP，以及一个Java写的跨平台Jabber Server – Wilefire。  
  
前言：  
  
即将完成Jabber Protocal(XMPP) : Core的翻译，在接下来的学习中将结合使用一个Jabber代码库和一个Jabber服务器，做些XMPP实现的分析与Practice。   
  
Jabber-NET  
  
Jabber-NET是一个使用.NET技术连接到Jabber服务器的类库集。它目前没有任何实现服务器端的计划，如果你兴趣于领导一个分项目来实现服务器端，那么就可以将其从你的选择中排除。它是用C#写的，但是也可以用.NET的其它语言（如VB.NET）进行引用，可以用于组件也可以用于客户端。同时当你探究时你会发现深藏其中的好东西，如Trees、命令行处理等，而且要比也是使用.NET技术的JabberCOM来的简单。  
  
事实上，Jabber-NET似乎已经被搁置，最近一次跟新也是在半年以前，跟新了一些新Xep的支持，以及对vs2005的支持。另外它的文档极少，而且除了邮件列表里有些声响，没有支持该项目的开发Forum或是Blog。也因此官方没有整理好的源文件及编译好的类库可下载。不过仅从中做一些参考，在它的CVS上还是可以进行的。  
  
在Jabber Code Libraries中的介绍：  
  
Code Libraries中罗列了用于 Jabber/XMPP 开发的一些知名类库。  
  
Library     
Language     
License     
Client     
Component     
Server     
  
Jabber-Net  
C#  
JOSL  
Yes  
Yes  
--  
  
  
有兴趣研究Jabber-Net的朋友可以下载我从CVS中整理编译后的类库：Jabber-NET Setup  
  
agsXMPP  
  
agsXMPP 是用C#写的，用于XMPP协议开发的SDK， 该SDK以基于"AG-Software shared source licence"的开源软件形式发布。目前版本0.1。  
  
不同于Jabber-NET，它作为一个SDK同时支持Client、Component和Server的开发；有着比较良好的文档、邮件列表及开发社区支持。官方页面：agsXMPP SDK，Forum: agsXMPP SDK。  
  
下面是一个简单示例，展示了采用agsXMPP登录XMPP服务器，发送一条信息给另一个用户的方便性。仅用三行代码： XmppClientConnection xmpp = new XmppClientConnection("jabber.org");  
xmpp.Open("myUsername", "mySecret");  
// Wait for the OnLogin event and send your message  
xmpp.Send( new Message("test@jabber.org", MessageType.chat, "Hello, how are you?"));  
在Jabber Code Libraries中的介绍：  
  
Library     
Language     
License     
Client     
Component     
Server     
  
agsXMPP  
C#  
Shared Source  
Yes  
Yes  
Yes  
  
  
  
Wildfire  
  
Wildfire就非常知名了，它不是类库，而是一个Java写的跨平台的Jabber服务器实现。它的安装、管理、定制、与其它应用的整合以及联合其它网络都非常简单，甚至做为你的自己的应用程序的平台也是如此方便。它有着强大的Jive软件社区支持 - Jive Software：instant messaging。  
  
Wildfire支持中文，目前版本为3.1，我们可以基于GPL许可证来使用。  
  
在Jabber Servers中的介绍：  
  
Servers中罗列了我们可以利用的，来自许多开源项目和商业用途的 Jabber/XMPP 服务器实现。  
  
Server     
Feature Score     
License     
Platforms     
  
Wildfire  
98%  
GPL or Proprietary  
AIX, HP-UX, Linux, MacOS X, Solaris, Windows  
  
  
  
  
更多的Jabber服务器介绍可以参见iso1600的：常用开源Jabber(XMPP) IM服务器介绍

[[](https://blog.csdn.net/devie)**devie**](https://blog.csdn.net/devie)

[关注](javascript:;)

* 1
* [[](https://blog.csdn.net/devie/article/details/7493988#commentBox)2](https://blog.csdn.net/devie/article/details/7493988#commentBox)
* [[](javascript:;)0](javascript:;)
* [](javascript:;)
* [](javascript:;)

专栏目录

[agsXMPP 开源IM全套代码](http://download.csdn.net/download/aoe168/9512699)

05-07

[最新agsXMPP全套代码，更新时间：2016.5.7， 开源的IM框架，目的是简化客户端的工作，将主要任务集中在服务端统一处理。](http://download.csdn.net/download/aoe168/9512699" \t "_blank)

[C#利用API发送和接收消息Demo](http://download.csdn.net/download/lybwwp/4781678)

11-18

[C#利用windows的API发送和接收消息Demo， 实现一个两个程序之间消息的通信，一个发送，另一个程序自动接收。 本代码测试全通过 .](http://download.csdn.net/download/lybwwp/4781678" \t "_blank)

[](https://blog.csdn.net/sanyan33588)

窗体顶端



表情包

窗体底端

* [](https://blog.csdn.net/jiangzh121)

[jiangzh121](https://blog.csdn.net/jiangzh121)**:**嗯，有没有源码呢6 年前回复



* [](https://blog.csdn.net/sqlite_me)

[sqlite\_me](https://blog.csdn.net/sqlite_me)**:**现在都1.1.0了7 年前回复

# 基于agsxmpp开发的即时聊天问题

[[](https://blog.csdn.net/fxy6781349)fxy6781349](https://blog.csdn.net/fxy6781349)2012-05-07 03:55:06

基于agsxmpp文件发送文件问题，发送文字已经解决 不知道发送附件是怎么处理的 我把  
agsXMPP.protocol.client.Message msg = new agsXMPP.protocol.client.Message();  
msg.From = new Jid(con.Username, Config.Server, "resourse");  
msg.To = new Jid(username, Config.Server, "resourse");  
msg.Type = MessageType.normal;  
msg.Body = fi.Name;  
msg.Attributes.Add("isfile", "1");  
msg.Attributes.Add("file", Convert.ToBase64String(bytes));  
msg.Attributes.Add("filesize", size.ToString());  
msg.Attributes.Add("filename", fi.Name);  
con.Send(msg);  
可以接收文件 但只能接收几百KB的 所以估计发送文件方式不是这样 请教各位 希望帮助！  
  
注：不使用自己定义的socket等方式传送文件，使用agsxmpp发送文件。

204点赞收藏7

分享

 写回复

**7 条回复**

 切换为时间正序

当前发帖距今超过3年，不再开放新的回复

发表回复

[](https://blog.csdn.net/fxy6781349)

[fxy6781349](https://blog.csdn.net/fxy6781349) 2012-05-07

公司最近做局域网即时聊天 用到xmpp 以前不熟悉这个 做到发送文件时倒是卡住了~~~

赞

[](https://blog.csdn.net/HttpResponse)

[别看等级看疗效](https://blog.csdn.net/HttpResponse) 2012-05-07

友情up下吧。也很喜欢xmpp,只是一直没有时间了解。

赞

[](https://blog.csdn.net/fxy6781349)

[fxy6781349](https://blog.csdn.net/fxy6781349) 2012-05-07

好像xmpp协议最大只支持3071字节 所以文件大一点就会超出这个字节 估计发送附件是其他的办法 大神们啊

赞

[](https://blog.csdn.net/fxy6781349)

[fxy6781349](https://blog.csdn.net/fxy6781349) 2012-05-07

呵呵 谢谢你的回答 其他解决了 就是发送附件不知道什么方法 在网上看到一些写法 但不是很明白 希望有熟悉的朋友能帮助

回复 赞

[](https://blog.csdn.net/saybookcat)

[saybookcat](https://blog.csdn.net/saybookcat) 2012-05-07

只研究过 asgXMPP的协议构建规则，无能为力。。。

赞

[](https://blog.csdn.net/fxy6781349)

[fxy6781349](https://blog.csdn.net/fxy6781349) 2012-05-07

希望有熟悉agsxmpp 或者有什么好的办法解决的 希望能说说您的看法....

[agsXMPP分析：agsXMPP简要指南 （转）](https://www.cnblogs.com/HappyQQ/archive/2008/01/13/1036646.html)

如果您还不知道agsXMPP是什么，您可以先阅读我的这篇随笔：[Jabber Software：Jabber-NET、agsXMPP与Wilefire](http://www.cnblogs.com/hunts/archive/2006/12/07/585014.html" \t "_blank)  
**agsXMPP Software Design**

agsXMPP致力于创建一个轻量、快速的跨平台类库，用于XMPP协议。

通过下面的三项技术，agsXMPP达到了这个目标。

1. 异步套接字
2. 与工厂模式结合的快速XML解析器
3. 自有的轻量级XML Dom，作为所有agsXMPP协议类的基础

我们为什么不直接所用Microsoft的System.Xml命名空间里的类呢？

因为我们决定创建自己的轻量级的Xml Dom，能够飞快地运作，特别是在像PPC's和Smartphones这样的嵌入式设备上。

XmlTextReader有利于SAX-like的解析。但是Microsoft在.NET1.1的SP1中做了下改变，这使得我们不能够再使用它来解析网络流。所以我们需要另外的XML解析器。

**类库的惊人之处在哪？**

一旦从套接字接收到数据，该数据便由sax-like XML解析器解析。此解析器使用工厂模式来创建agsXMPP协议类相关的元素。

**示例：**

套接字接收到一条信息，将比特流推送至解析器。XML解析器探测到隶属于jabber:client命名空间中名字为**message**的开标签符。在元素创建前，解析器在工厂散列表中做个查找。这样就创建了agsXMPP.protocol.client.Message类的一个实例。如果表中不存在name/namespace的绑定，则会创建agsXMPP.Xml.Element的一个实例。

所有的XMPP协议类都派生自agsXMPP.Xml.Element。他们都是在内存中保持XML树的'abstract'元素。所有的属性都是'realtime properties'。在我们要读取消息体，调用消息体属性时，类将会实时查找<body/>元素。

**Creating your own packet types**

下面的小例子中我们将要创建一个我们自己的扩展，通过XMPP网络发送天气信息。最简单的方法是将天气的数据信息嵌入到一个message节中。嵌入了天气信息的新XML message如下所示：

<message xmlns="jabber:client" to="romeo@montage.net">  
    <weather xmlns="agsoftware:weather">  
        <humidity>90</humidity>  
        <temperature>57</temperature>  
    </weather>  
</message>

我们要给这个新的协议创建一个新的命名空间和3个新元素：weather、humidity和temperature

首先，我们给我们定制的XML元素创建一个新的类weather.cs，派生自agsXMPP.Xml.Dom.Element。

using System;  
using agsXMPP.Xml.Dom;  
  
namespace MiniClient  
{  
    public class Weather : Element  
    {  
        public Weather()  
        {  
            this.TagName = "weather";  
            this.Namespace = "agsoftware:weather";  
        }  
        public Weather(int humidity, int temperature) : this()  
        {  
            this.Humidity = humidity;  
            this.Temperature = temperature;  
        }  
        public int Humidity  
        {  
            get { return GetTagInt("humidity"); }  
            set { SetTag("humidity", value.ToString()); }  
        }  
        public int Temperature  
        {  
            get { return GetTagInt("temperature"); }  
            set { SetTag("temperature", value.ToString()); }  
        }  
    }  
}

然后在元素工厂中注册这个新类。如果不注册，在解析XML流时XML解析器就不会weather对象。我们通过下面的代码注册该类：

agsXMPP.Factory.ElementFactory.AddElementType("weather",  
"agsoftware:weather", typeof(Weather));

我们在使用agsXMPP处理其它事件时，应该先注册我们自己的元素。

现在我们能够创建自己的weather message，然后发送：

Weather weather = new Weather(90, 57);  
Jid to = new Jid("romeo@montage.net");  
Message msg = new Message();  
msg.To = to;  
// Add our weather Element  
msg.AddChild(weather);  
// Send the message  
XmppCon.Send(msg);

接收此message的另一个应用程序可以像OnMessage handler那样访问到我们的定制数据：

private void XmppCon\_OnMessage(object sender, Message msg)  
{  
    if (msg.HasTag(typeof(Weather)))  
    {  
        Weather weather = msg.SelectSingleElement(typeof(Weather)) as Weather;  
        Console.WriteLine(weather.Temperature.ToString());  
        Console.WriteLine(weather.Humidity.ToString());  
    }  
}

现在我们创建了我们第一个自己的（信息）包，包含了一个message节。接下去我们要创建基于Iqs的小型weather service。

service的raw xml protocol：

Romeo向weather service请求了他城市的天气信息，zip code为'74080'：

<iq from='romeo@montagne.net' to='weather.mortagne.net' type='get' id='agsXMPP\_1'>  
    <query xmlns='agsoftware:weather'>  
        <zip>74080</zip>  
    </query>  
</iq>

接着weather service查找该zip code的天气数据，然后返回结果给Romeo：

<iq to='romeo@montagne.net' from='weather.mortagne.net' type='result' id='agsXMPP\_1'>  
    <query xmlns='agsoftware:weather'>  
        <humidity>90</humidity >  
        <temperature>57</temperature>  
        <zip>74080</zip>  
    </query>  
</iq>

Weather.cs的源代码：

using System;  
using agsXMPP.Xml;  
using agsXMPP.Xml.Dom;  
  
public class Weather : Element  
{  
    public Weather()  
    {  
        this.TagName = "query";  
        this.Namespace = "agsoftware:weather";  
    }  
    public int Humidity  
    {  
        get { return GetTagInt("humidity"); }  
        set { SetTag("humidity", value.ToString()); }  
    }  
    public int Temperature  
    {  
        get { return GetTagInt("temperature"); }  
        set { SetTag("temperature", value.ToString()); }  
    }  
    public int Zip  
    {  
        get { return GetTagInt("zip"); }  
        set { SetTag("zip", value.ToString()); }  
    }  
}

WeatherIq.cs的源代码：

using System;  
using agsXMPP;  
using agsXMPP.protocol.client;  
  
public class WeatherIq : IQ  
{  
    private Weather m\_Weather = new Weather();  
  
    public WeatherIq()  
    {  
        base.Query = m\_Weather;  
        this.GenerateId();  
    }  
  
    public WeatherIq(IqType type)  
        : this()  
    {  
        this.Type = type;  
    }  
  
    public WeatherIq(IqType type, Jid to)  
        : this(type)  
    {  
        this.To = to;  
    }  
  
    public WeatherIq(IqType type, Jid to, Jid from)  
        : this(type, to)  
    {  
        this.From = from;  
    }  
  
    public new Weather Query  
    {  
        get  
        {  
            return m\_Weather;  
        }  
    }  
}

当然我们要在工厂里注册Weather对象，使用下面代码：

agsXMPP.Factory.ElementFactory.AddElementType("weather",  
"agsoftware:weather", typeof(Weather));

Romeo的客户端使用WeatherIq对象创建了请求Iq包：

WeatherIq wIq = new WeatherIq(IqType.get);  
wIq.To = new Jid("weather.mortagne.net");  
wIq.From = new Jid("romeo@montagne.net");  
wIq.Query.zip = 74080;  
// Send the message  
XmppCon.Send(wIq);

weather service在Iq handler中接收此请求，然后把响应发回给Romeo。在weather service中获得请求信息，然后编辑：

private void XmppCon\_OnIq(object sender, agsXMPP.Xml.Dom.Node e)  
{  
    IQ iq = e as IQ;  
    Element query = iq.Query;  
  
    if (query != null)  
    {  
        if (query.GetType() == typeof(Weather))  
        {  
            // its a Weather IQ  
            Weather weather = query as Weather;  
            if (iq.Type == IqType.get)  
            {  
                Console.WriteLine(weather.Zip.ToString());  
                // read the zip code and lookup the weather  
                // data for this zip code  
                // . . . .  
                iq.SwitchDirection();  
                iq.Type = IqType.result;  
                weather.Humidity = 90;  
                weather.Temperature = 57;  
                // Send the result back to romeo  
                XmppCon.Send(iq);  
            }  
        }  
    }  
}

Romeo接收到响应，然后通过我们的weather对象的属性访问到天气数据。

成功者找方法，失败者找借口！

## [agsXMPP](https://www.cnblogs.com/sherlockhua/archive/2008/04/13/1151458.html)

Posted on 2008-04-13 18:59  [少林](https://www.cnblogs.com/sherlockhua/)  阅读(1812)  评论(0)  [编辑](https://i.cnblogs.com/EditPosts.aspx?postid=1151458)  [收藏](javascript:void(0))  [举报](javascript:void(0))

 一直想写一个类似qq的即时通信程序,由于某些原因,一直没有时间.所以拖到了今天,还丝毫没有进展.现在，好不容易有了时间,趁大学毕业之前,赶紧把这个梦想实现吧.今天到网上收集了一些IM的资料,还是有很大的收获吧.决定采用agsxmpp基础上进行开发,而agrscmpp采用是XMPP协议,看样子,对XMPP协议也要有的比较清楚的了解.部分资料如下.

     agsXMPP is a SDK / library for the e**X**tensible **M**essaging and **P**resence **P**rotocol (XMPP) protocol written in managed C# dedicated to .NET and Mono technologies. The SDK is released as open source under a [dual license](http://www.ag-software.de/index.php?page=license).  
  
Our SDK could be used for XMPP **client**, **server** and **component** development.  
  
Here is a small sample how easy you can login to a XMPP Server and send a simple chat-message to another user in only 3 lines of code with agsXMPP:

    XmppClientConnection xmpp = new XmppClientConnection("jabber.org");      
    xmpp.Open("myusername", "mysecret");  
    xmpp.OnLogin += delegate(object o) { xmpp.Send(new Message("test@jabber.org", MessageType.chat,     "Hello,     how are you?")); };

官方网站:http://www.ag-software.de/index.php?page=agsxmpp-sdk  
             http://www.xmpp.org/

基于QT的IM(jabber)库和客户端

cymlife 2011-06-27 11:27:00 10539 收藏 5

分类专栏： QT跨平台开发 文章标签： qt im xmpp 跨平台 聊天 服务器

版权

QT跨平台开发

专栏收录该内容

0 篇文章0 订阅

订阅专栏

本文来自http://blog.csdn.net/cymlife 原创，引用必须注明出处！

基于QT的IM(jabber)库和客户端

这段时间一直在研究跨平台的IM(jabber)客户端，经过大量的下载、编译、测试、评估，作此记录。

使用QT作为表现层，源于QT提供了丰富的控件，以及强大的跨平台能力。jabber是一个基于XMPP协议的IM应用，解决了做什么和怎么做的问题。

使用QT来开发的jabber xmpp库，主要是依据XMPP协议，和jabber服务器进行交互，在IM系统中起到通信协议层的作用。xmpp的客户端，必须要实现XMPP RFC3920(XMPP CORE，最新规范为RFC 6120)和RFC3921(XMPP IM，最新规范为RFC 6121)，以及其他辅助的XMPP Extensions协议。

jabber xmpp库

1. gloox

http://camaya.net/gloox/

纯C++的跨平台实现，全功能，易于使用。不过自1.0之后就没有更新了。

2. qxmpp

http://code.google.com/p/qxmpp/

纯QT实现，易于使用。目前版本是0.3，一直在更新。

qxmpp实现了很多XMPP Extensions协议，比如文件传输，vcard，avatar等功能。

3. jreen

http://gitorious.org/jreen

纯QT实现，易于使用。一直在更新。

4. psi(libiris)

http://psi-im.org/

psi从0.14版本之后就没有更新了。

5. psi-dev

http://code.google.com/p/psi-dev/

一群热心的开发人员继续对psi进行维护和升级，目前版本是0.15。

6. libpurple

http://pidgin.im/download/

支持多种IM的协议库，最新版本是2.9.0。

jabber xmpp QT-IM客户端

1. vacuum-im

http://code.google.com/p/vacuum-im/

纯QT实现，易于使用。目前版本是1.1，一直在更新。

vacuum实现了很多XMPP Extensions协议，比如文件传输，vcard，avatar，组聊等功能；采用插件方式，易于扩展。

2. qutecom

http://trac.qutecom.org/

qutecom的IM库采用libpurple，集成了sip电话的功能。最新稳定版是2.2.1，开发版是3.0 alpha。

qutecom使用了大量的开源库，包括：boost，ffmpeg，libosip2等等，编译过程复杂，编译速度极慢。

3. qutim

http://qutim.org/

qutim是支持多协议、跨平台（包括mobile）的IM，其中jabber协议使用jreen。系统采用插件功能，易于扩展。

4. kadu

http://www.kadu.net/w/English:Main\_Page

kadu支持的IM协议包括Gadu-Gadu和jabber（使用libiris），程序结构清晰，界面友好，易于使用。

5. jabbin

http://sourceforge.net/projects/jabbin/

jabbin的IM协议使用libiris，使用libjingle库提供语音功能。jabbin的代码还使用了很多QT3的功能，有点老旧，因为QT5都快出来了。

6. kopete

http://kopete.kde.org/

kopete是KDE平台支持多协议的IM，其中jabber协议使用libiris。从0.50.80之后就是打打补丁而已。

IM客户端功能

一个完整的IM客户端，功能还是很复杂的，基本包括：

通讯协议层：和IM服务器交互；

用户管理：组织机构（分组，企业通讯录），名片管理(vcard)，头像显示(avatar)，用户状态，用户查找等；

文字聊天功能：聊天方式包括海聊，组聊，私聊；表情管理；贴图功能；拼写检查；聊天风格；

音视频聊天功能：使用google的webrtc；

聊天历史记录：查询；

文件传输：在线与离线文件传输；

界面风格：吸引用户的首要因素。

————————————————

版权声明：本文为CSDN博主「cymlife」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/cymlife/article/details/6569824