

ChatRev M

ChatRev M
Copyright©2004-2005 by Mark Schonewille
Nijmegen, the Netherlands
Date: 17 dec 2004



Read this file before using ChatRev 1.2M19. A modified license is included below.

ChatRev 1.2M19

ChatRev Client is an application that can connect to a ChatRev Server. It works like any other regular chatbox. Even though there are messengers such as MSN, AIM, ICQ, Gadu-Gadu, iChat, and others, there might still be occasions when you want to use a chatbox where you can meet a community. For example, you can keep this application open while you are developing your own software, to talk to your fellow-developers. It is also possible download and run a server and build your own community. ChatRev 1.2M19 is compatible with the ChatRev protocol. Major features include:

- on-line English dictionary using Babylon
- file transfer to and from other clients
- sending colourised text using menus or palette
- compatibility with the new forum format
- speech
- clickable web links
- automatic linking to Bugzilla
- graphical smileys
- optional interpreting of HTML tags
- an opportunity to sponsor the project

The stack and standalone versions of ChatRev 1.2M13 can be downloaded at <http://home.wanadoo.nl/mark.sch/software/main.html>. From there, follow the link to ChatRev.

Quickstart

If you open ChatRev for the first time, choose the default url <http://www.revolutionboard.de/bvg/ip.txt> from the IP URL menu. An IP number will appear in the IP menu. Enter 80 into the Port field. Also, enter a user name and a password and click on the Connect button. You will be asked to register. Fill out the form and click on Okay. Then click on Connect again.

There are additional components available from the ChatRev M homepage. The files smileys.sit and smileys.zip contain smiley pictures for that can be stored locally. Put the Smileys folder into the User Dat folder of the Rev Chat application. You can put the sound file glass.aif into the User Data as well and configure your setting to play this sound file upon message arrival.

To-do List

In ChatRev 1.2M19, a few interface issues still have to be solved. A general issue is the intention to modify the file transfer procedure to allow for cancelling the transfer. E-Mail links will be better recognised in M20. Addition of timestamps is considered. Modification to allow for better clickable links in MetaCard is considered but may not be implemented until Mark has a MetaCard license. The built-in documentation is incomplete, since newly added features are not discussed. A new help system is also on the to-do list.

ChatRev M License

You agree to this license by using ChatRev 1.2M19 or later, or by using or modifying the code or ideas implemented in ChatRev 1.2M19 or later.

This license applies to ChatRev 1.2m19 (and later, if later versions are not accompanied by a modified license).

This project is open source. You can change it as you like, but If you copy any code from this project, or use any ideas implemented in this project, for use in your own project, you must return your project to the public domain. You must include the complete text of this file.

Original client by Björnke von Gierke. This version by Mark Schonewille.
Copyright©2004 this version by Mark Schonewille, Nijmegen, the Netherlands
Portions copyright©2004 by Björnke von Gierke, Switzerland.
Portions copyright©2000-2004 by Runtime Revolution Limited Edinburgh, Scotland

Download this project at: <http://home.wanadoo.nl/mark.sch/software/main.html>. Björnke's client and server are available, at <http://chatrev.cjb.net> (this address will be accessible from the week before Christmas 2004 onwards).

Björnke runs the default server. His IP URL is: <http://www.revolutionboard.de/bvg/ip.txt>. The default port is 8008 (from the week before Christmas 2004 onwards). Mark might be running a server, sometimes. The IP URL is: <<http://home.wanadoo.nl/mark.sch/ip.txt>>. If the server is running, it will usually be on port 8080.

Reporting Bugs

To improve this client, it is very important that bugs are reported. If you encounter any problems with the client or have other comments, please send an e-mail to chatrev@schonewille.cjb.net. Include an extensive description of what you did before the problem occurred and add what you would have expected to happen instead. Also list the hardware and software you are using, such as type and model of computer, memory, free space on your hard disk, operating system, et cetera. Bug reports cannot be too long!

Sponsoring

It is now possible to make a donation to the ChatRev M project. Choose the Sponsor menu item from the Help menu to contribute through PayPal. The server project can also be sponsored. For more information about sponsoring the server, go to <http://chatrev.cjb.net>.

Protocol as of 15th December 2004

This is the protocol for ChatRev Server 1.2.6. If you make your own client, you need to comply with this protocol for compatibility with the ChatRev server and other ChatRev clients. Note that the port (80) mentioned in the protocol will change in the near future. The new port will probably be 8008.

Connecting to the chat

You need to know the IP of the chat server to connect to it. currently there is a file which contains the IP of my server. its url is "http://www.revolutionboard.de/bvg/ip.txt" The port used is 80, and you might find another server on port 8080. 8080 is the test server, and might be taken down any time, where 80 is the stable server and has no planned downtimes.

Every message is delimited by returns. Every message is ended with return. In the examples I give, "return" is written as "cr". Also in the examples is "comma" which means the comma character (.). The last mentioned delimiter is "tab" which means the tabulator. Text you send is in quotes (eg: "login", variables you need to fill in are plain (eg: yourMessageHere). All communication is case independent.

Registering

Registering is currently a simple matter, with few crosstalk. First you open the socket to the chatrev server.

Then you write your first message:

```
"register" & cr & nickname & comma & password & comma & "deprecate" & cr
```

The nickname can contain any chars, besides "tab" or ","(comma).
the password can contain any chars, besides ",",.

The server responses with one of these messages:

```
"taken" & cr
```

this means that the nick is already registered;

```
"empty nick" & cr
```

this means that item one (above: nickname) of your message was empty; or

```
"registered" & cr
```

meaning that your client has successfully registered the user nickname.

If you want to set the user modifiable data (mouseover, about, label), then you just send the appropriate messages now (described below).

after registering you send a disconnect message:

```
"disconnect" & cr
```

the socket to the server should now be closed by using the "close socket" command.

Login

First you open the socket to the chatrev server.

Then you send your login information:

```
"login" & cr & nickname & comma & password & cr
```

The server responds with one of these messages:

```
"login wrong" & cr
```

either the user was not found, or the password was wrong.

```
"logged in" & cr
```

You can now proceed with logging in (you are not finished yet). You now write your version information:

```
"version 1.2" & cr
```

The server should now send you one of these:

```
"version OK" & cr
```

If it is ANYTHING else, then you should disconnect, and show the received message to the user:

```
"please update to version 1.2.6, http://homepage.mac.com/bvg/chatrev1.2.6.rev" & cr
```

The server now proceeds to send your login information to everyone else that is connected. This contains: a "member" message, a "setmouse" message. The server also sends the same to you about every other logged in client. The other clients also get a "message" from the server, saying "Nickname: logged on at " and the server time. Where you get a "message" from the server saying "Nickname: you are logged in!" and the server time. When you have received that, then you are properly connected.

Status messages

All these messages can be received at any time after you have send the version number.

The setmouse message sets the label (previously called "MouseOver") of a user to a certain value. It goes like this:

```
"setmouse" & cr & Nickname & tab & label & cr
```

You now can set the label of the user Nickname. The member message indicates that a user has logged on. It goes like this:

```
"member" & cr & nickname & tab & label & cr
```

After this message you automatically get a message by the server that the person has logged on. Note that you can't rely on the label for identifying users, just use the nickname to do that. The unmember message indicates that a user has logged off It goes like this:

```
"unmember" & cr & Nickname & tab & label & cr
```

After this message you automatically get a message by the server that the person has logged off. Note that you can't rely on the label for identifying users, just use the nickname to do that. The error message indicates that you have done something bad (not following the protocol) so you get this message:

```
"error" & cr & theErrorMessage & cr
```

You should disconnect after this message, socket communication can not be guaranteed anymore. The kick message is there to kick too talkative people.

```
"kick" & cr
```

You should disconnect when receiving this message.

Chat messages

The chat message indicates a message that is send to everyone. Excluded are messages from the server which might get send to everyone, but are always "messages". Chat goes like this:

```
"chat" & cr & Nickname & cr & htmltext & cr
```

You receive this to show the text. Version 1.1 and 1.0 have submitted clear text here, but htmltext allows for multiple lines and formatting. The message message indicates a private message, only send to you (except server messages which are send to everyone). it goes like this:

```
"message" & cr & Nickname & cr & htmltext & cr
```

You receive this to show the text. Version 1.1 and 1.0 have submitted clear text here, but htmltext allows for multiple lines and formatting. The me message indicates a humorous action. it is send to everyone, like the chat message.

```
"me" & cr & Nickname & cr & plaintext & cr
```

The me message allows no formatting or htmltext. Messages you can write are

```
"chat" & cr & htmltext & cr
"message" & cr & htmltext & cr
"me" & cr & text & cr
"setabout" & cr & text & cr
"setMouse" & cr & text & cr
"setmail" & cr & text & cr
```

The text in the setmail message may not contain spaces. and should be a mail address

```
"getmail" & cr & Nickname & space & subject & cr
```

When you send this message you automatically get a response from the server:

```
"mail" & cr & mailadress & space & subject & cr
```

where subject is the same as you have send before.

```
"getAbout" & cr & Nickname & cr
```

Again this gives you a autoreply, this time a private message which contains the about of the user Nickname. Last but not least there is the ping message. you should write a pong message to the server when you encounter a ping, or you will be disconnected.

```
"ping" & cr
"pong" & cr
```

File Request

When a client want's to initiate a file transfer he has to send the server a file request message, with a user as target, the file path, and a short description:

```
fileRequest & cr & RemoteUser & cr & filepath & cr & filelength & comma & description & cr
```

The server sends a message to the appropriate users' client. It contains the username that wants to initiate a file transfer, the filepath, and the short description:

```
fileRequest & cr & ContactingUser & cr & filepath & cr & filelength & comma & description & cr
```

The contacted client now either declines the proposal or accepts it:

Whether he accepts it or not he sends the user that wants to establish the connection back to the server.

```
fileAccept & cr & ContactingUser & cr & filepath & cr & "any message" & cr
fileAccept & cr & ContactingUser & cr & filepath & cr & "accept" & cr
```

The server now relays this message to the first (the contacting) client.
The server also includes the ip if the file transfer was accepted.

```
fileAccept & cr & RemoteUser & cr & filepath & cr & "any message" & cr
fileAccept & cr & RemoteUser & cr & filepath & cr & theIP & cr
```

Forum Messages

To retrieve the threads of the forum, write

```
"threads" & cr & forumName & cr
```

to the server. ForumName can currently be either "General discussion" or "Support forum". The server replies with

```
"threads" & cr & messageLength & cr & threadsList
```

To send a message to a forum, write

```
"postReply" & cr & forumName & cr & threadID & cr & messageLength & cr & messageText
```

to the server.

Retrieve the text of a thread by sending

```
"getThread" & cr & forumName & cr & threadID & cr
```

to the server. The server returns the entire thread as follows:

```
"thread" & cr & messageLength & cr & messageText
```

You can make a new thread by writing this

```
"makeThread" & cr & forumName & cr & threadName & cr messageLength & cr & messageText
```

to the server.

(eof)