|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PircBot.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-all.html) | [**Help**](http://docs.google.com/help-doc.html) | |  |  | ***PircBot Java IRC Bot*** |
| [**PREV CLASS**](http://docs.google.com/org/jibble/pircbot/OutputThread.html)   [**NEXT CLASS**](http://docs.google.com/org/jibble/pircbot/Queue.html) | [**FRAMES**](http://docs.google.com/index.html)    [**NO FRAMES**](http://docs.google.com/PircBot.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#id.3znysh7)| [CONSTR](#id.tyjcwt)| [METHOD](#id.3dy6vkm) | DETAIL: [FIELD](#id.4d34og8)| [CONSTR](#id.17dp8vu)| [METHOD](#id.26in1rg) |

**org.jibble.pircbot**

Class PircBot

[java.lang.Object](http://java.sun.com/j2se/1.4/docs/api/java/lang/Object.html)  
 **org.jibble.pircbot.PircBot**

**All Implemented Interfaces:** [ReplyConstants](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html)public abstract class **PircBot**extends [Object](http://java.sun.com/j2se/1.4/docs/api/java/lang/Object.html)implements [ReplyConstants](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html)

PircBot is a Java framework for writing IRC bots quickly and easily.

It provides an event-driven architecture to handle common IRC events, flood protection, DCC support, ident support, and more. The comprehensive logfile format is suitable for use with pisg to generate channel statistics.

Methods of the PircBot class can be called to send events to the IRC server that it connects to. For example, calling the sendMessage method will send a message to a channel or user on the IRC server. Multiple servers can be supported using multiple instances of PircBot.

To perform an action when the PircBot receives a normal message from the IRC server, you would override the onMessage method defined in the PircBot class. All on*XYZ* methods in the PircBot class are automatically called when the event *XYZ* happens, so you would override these if you wish to do something when it does happen.

Some event methods, such as onPing, should only really perform a specific function (i.e. respond to a PING from the server). For your convenience, such methods are already correctly implemented in the PircBot and should not normally need to be overridden. Please read the full documentation for each method to see which ones are already implemented by the PircBot class.

Please visit the PircBot homepage at <http://www.jibble.org/pircbot.php> for full revision history, a beginners guide to creating your first PircBot and a list of some existing Java IRC bots and clients that use the PircBot framework.

**Version:** 1.5.0 (Build time: Mon Dec 14 20:07:17 2009) **Author:** Paul James Mutton, <http://www.jibble.org/>

|  |  |
| --- | --- |
| **Field Summary** |  |
| static [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) | [**VERSION**](http://docs.google.com/org/jibble/pircbot/PircBot.html#VERSION)            The definitive version number of this release of PircBot. |

|  |
| --- |
| **Fields inherited from interface org.jibble.pircbot.**[**ReplyConstants**](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html) |
| [ERR\_ALREADYREGISTRED](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_ALREADYREGISTRED), [ERR\_BADCHANMASK](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_BADCHANMASK), [ERR\_BADCHANNELKEY](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_BADCHANNELKEY), [ERR\_BANNEDFROMCHAN](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_BANNEDFROMCHAN), [ERR\_CANNOTSENDTOCHAN](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_CANNOTSENDTOCHAN), [ERR\_CANTKILLSERVER](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_CANTKILLSERVER), [ERR\_CHANNELISFULL](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_CHANNELISFULL), [ERR\_CHANOPRIVSNEEDED](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_CHANOPRIVSNEEDED), [ERR\_ERRONEUSNICKNAME](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_ERRONEUSNICKNAME), [ERR\_FILEERROR](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_FILEERROR), [ERR\_INVITEONLYCHAN](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_INVITEONLYCHAN), [ERR\_KEYSET](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_KEYSET), [ERR\_NEEDMOREPARAMS](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NEEDMOREPARAMS), [ERR\_NICKCOLLISION](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NICKCOLLISION), [ERR\_NICKNAMEINUSE](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NICKNAMEINUSE), [ERR\_NOADMININFO](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NOADMININFO), [ERR\_NOLOGIN](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NOLOGIN), [ERR\_NOMOTD](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NOMOTD), [ERR\_NONICKNAMEGIVEN](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NONICKNAMEGIVEN), [ERR\_NOOPERHOST](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NOOPERHOST), [ERR\_NOORIGIN](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NOORIGIN), [ERR\_NOPERMFORHOST](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NOPERMFORHOST), [ERR\_NOPRIVILEGES](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NOPRIVILEGES), [ERR\_NORECIPIENT](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NORECIPIENT), [ERR\_NOSERVICEHOST](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NOSERVICEHOST), [ERR\_NOSUCHCHANNEL](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NOSUCHCHANNEL), [ERR\_NOSUCHNICK](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NOSUCHNICK), [ERR\_NOSUCHSERVER](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NOSUCHSERVER), [ERR\_NOTEXTTOSEND](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NOTEXTTOSEND), [ERR\_NOTONCHANNEL](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NOTONCHANNEL), [ERR\_NOTOPLEVEL](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NOTOPLEVEL), [ERR\_NOTREGISTERED](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_NOTREGISTERED), [ERR\_PASSWDMISMATCH](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_PASSWDMISMATCH), [ERR\_SUMMONDISABLED](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_SUMMONDISABLED), [ERR\_TOOMANYCHANNELS](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_TOOMANYCHANNELS), [ERR\_TOOMANYTARGETS](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_TOOMANYTARGETS), [ERR\_UMODEUNKNOWNFLAG](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_UMODEUNKNOWNFLAG), [ERR\_UNKNOWNCOMMAND](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_UNKNOWNCOMMAND), [ERR\_UNKNOWNMODE](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_UNKNOWNMODE), [ERR\_USERNOTINCHANNEL](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_USERNOTINCHANNEL), [ERR\_USERONCHANNEL](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_USERONCHANNEL), [ERR\_USERSDISABLED](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_USERSDISABLED), [ERR\_USERSDONTMATCH](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_USERSDONTMATCH), [ERR\_WASNOSUCHNICK](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_WASNOSUCHNICK), [ERR\_WILDTOPLEVEL](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_WILDTOPLEVEL), [ERR\_YOUREBANNEDCREEP](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_YOUREBANNEDCREEP), [ERR\_YOUWILLBEBANNED](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#ERR_YOUWILLBEBANNED), [RPL\_ADMINEMAIL](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_ADMINEMAIL), [RPL\_ADMINLOC1](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_ADMINLOC1), [RPL\_ADMINLOC2](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_ADMINLOC2), [RPL\_ADMINME](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_ADMINME), [RPL\_AWAY](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_AWAY), [RPL\_BANLIST](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_BANLIST), [RPL\_CHANNELMODEIS](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_CHANNELMODEIS), [RPL\_CLOSEEND](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_CLOSEEND), [RPL\_CLOSING](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_CLOSING), [RPL\_ENDOFBANLIST](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_ENDOFBANLIST), [RPL\_ENDOFINFO](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_ENDOFINFO), [RPL\_ENDOFLINKS](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_ENDOFLINKS), [RPL\_ENDOFMOTD](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_ENDOFMOTD), [RPL\_ENDOFNAMES](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_ENDOFNAMES), [RPL\_ENDOFSERVICES](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_ENDOFSERVICES), [RPL\_ENDOFSTATS](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_ENDOFSTATS), [RPL\_ENDOFUSERS](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_ENDOFUSERS), [RPL\_ENDOFWHO](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_ENDOFWHO), [RPL\_ENDOFWHOIS](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_ENDOFWHOIS), [RPL\_ENDOFWHOWAS](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_ENDOFWHOWAS), [RPL\_INFO](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_INFO), [RPL\_INFOSTART](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_INFOSTART), [RPL\_INVITING](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_INVITING), [RPL\_ISON](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_ISON), [RPL\_KILLDONE](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_KILLDONE), [RPL\_LINKS](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_LINKS), [RPL\_LIST](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_LIST), [RPL\_LISTEND](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_LISTEND), [RPL\_LISTSTART](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_LISTSTART), [RPL\_LUSERCHANNELS](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_LUSERCHANNELS), [RPL\_LUSERCLIENT](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_LUSERCLIENT), [RPL\_LUSERME](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_LUSERME), [RPL\_LUSEROP](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_LUSEROP), [RPL\_LUSERUNKNOWN](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_LUSERUNKNOWN), [RPL\_MOTD](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_MOTD), [RPL\_MOTDSTART](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_MOTDSTART), [RPL\_MYPORTIS](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_MYPORTIS), [RPL\_NAMREPLY](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_NAMREPLY), [RPL\_NONE](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_NONE), [RPL\_NOTOPIC](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_NOTOPIC), [RPL\_NOUSERS](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_NOUSERS), [RPL\_NOWAWAY](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_NOWAWAY), [RPL\_REHASHING](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_REHASHING), [RPL\_SERVICE](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_SERVICE), [RPL\_SERVICEINFO](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_SERVICEINFO), [RPL\_SERVLIST](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_SERVLIST), [RPL\_SERVLISTEND](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_SERVLISTEND), [RPL\_STATSCLINE](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_STATSCLINE), [RPL\_STATSCOMMANDS](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_STATSCOMMANDS), [RPL\_STATSHLINE](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_STATSHLINE), [RPL\_STATSILINE](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_STATSILINE), [RPL\_STATSKLINE](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_STATSKLINE), [RPL\_STATSLINKINFO](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_STATSLINKINFO), [RPL\_STATSLLINE](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_STATSLLINE), [RPL\_STATSNLINE](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_STATSNLINE), [RPL\_STATSOLINE](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_STATSOLINE), [RPL\_STATSQLINE](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_STATSQLINE), [RPL\_STATSUPTIME](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_STATSUPTIME), [RPL\_STATSYLINE](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_STATSYLINE), [RPL\_SUMMONING](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_SUMMONING), [RPL\_TIME](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_TIME), [RPL\_TOPIC](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_TOPIC), [RPL\_TOPICINFO](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_TOPICINFO), [RPL\_TRACECLASS](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_TRACECLASS), [RPL\_TRACECONNECTING](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_TRACECONNECTING), [RPL\_TRACEHANDSHAKE](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_TRACEHANDSHAKE), [RPL\_TRACELINK](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_TRACELINK), [RPL\_TRACELOG](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_TRACELOG), [RPL\_TRACENEWTYPE](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_TRACENEWTYPE), [RPL\_TRACEOPERATOR](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_TRACEOPERATOR), [RPL\_TRACESERVER](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_TRACESERVER), [RPL\_TRACEUNKNOWN](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_TRACEUNKNOWN), [RPL\_TRACEUSER](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_TRACEUSER), [RPL\_UMODEIS](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_UMODEIS), [RPL\_UNAWAY](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_UNAWAY), [RPL\_USERHOST](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_USERHOST), [RPL\_USERS](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_USERS), [RPL\_USERSSTART](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_USERSSTART), [RPL\_VERSION](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_VERSION), [RPL\_WHOISCHANNELS](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_WHOISCHANNELS), [RPL\_WHOISCHANOP](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_WHOISCHANOP), [RPL\_WHOISIDLE](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_WHOISIDLE), [RPL\_WHOISOPERATOR](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_WHOISOPERATOR), [RPL\_WHOISSERVER](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_WHOISSERVER), [RPL\_WHOISUSER](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_WHOISUSER), [RPL\_WHOREPLY](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_WHOREPLY), [RPL\_WHOWASUSER](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_WHOWASUSER), [RPL\_YOUREOPER](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html#RPL_YOUREOPER) |

|  |  |
| --- | --- |
| **Constructor Summary** |  |
| [**PircBot**](http://docs.google.com/org/jibble/pircbot/PircBot.html#PircBot())()            Constructs a PircBot with the default settings. |

|  |  |
| --- | --- |
| **Method Summary** |  |
| void | [**ban**](http://docs.google.com/org/jibble/pircbot/PircBot.html#ban(java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostmask)            Bans a user from a channel. |
| void | [**changeNick**](http://docs.google.com/org/jibble/pircbot/PircBot.html#changeNick(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) newNick)            Attempt to change the current nick (nickname) of the bot when it is connected to an IRC server. |
| void | [**connect**](http://docs.google.com/org/jibble/pircbot/PircBot.html#connect(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostname)            Attempt to connect to the specified IRC server. |
| void | [**connect**](http://docs.google.com/org/jibble/pircbot/PircBot.html#connect(java.lang.String,%20int))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostname, int port)            Attempt to connect to the specified IRC server and port number. |
| void | [**connect**](http://docs.google.com/org/jibble/pircbot/PircBot.html#connect(java.lang.String,%20int,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostname, int port, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) password)            Attempt to connect to the specified IRC server using the supplied password. |
| protected  [DccChat](http://docs.google.com/org/jibble/pircbot/DccChat.html) | [**dccAcceptChatRequest**](http://docs.google.com/org/jibble/pircbot/PircBot.html#dccAcceptChatRequest(java.lang.String,%20long,%20int))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, long address, int port)  **Deprecated.** *As of PircBot 1.2.0, use* [*onIncomingChatRequest(DccChat)*](http://docs.google.com/org/jibble/pircbot/PircBot.html#onIncomingChatRequest(org.jibble.pircbot.DccChat)) |
| protected  void | [**dccReceiveFile**](http://docs.google.com/org/jibble/pircbot/PircBot.html#dccReceiveFile(java.io.File,%20long,%20int,%20int))([File](http://java.sun.com/j2se/1.4/docs/api/java/io/File.html) file, long address, int port, int size)  **Deprecated.** *As of PircBot 1.2.0, use* [*onIncomingFileTransfer(DccFileTransfer)*](http://docs.google.com/org/jibble/pircbot/PircBot.html#onIncomingFileTransfer(org.jibble.pircbot.DccFileTransfer)) |
| [DccChat](http://docs.google.com/org/jibble/pircbot/DccChat.html) | [**dccSendChatRequest**](http://docs.google.com/org/jibble/pircbot/PircBot.html#dccSendChatRequest(java.lang.String,%20int))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) nick, int timeout)            Attempts to establish a DCC CHAT session with a client. |
| [DccFileTransfer](http://docs.google.com/org/jibble/pircbot/DccFileTransfer.html) | [**dccSendFile**](http://docs.google.com/org/jibble/pircbot/PircBot.html#dccSendFile(java.io.File,%20java.lang.String,%20int))([File](http://java.sun.com/j2se/1.4/docs/api/java/io/File.html) file, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) nick, int timeout)            Sends a file to another user. |
| void | [**deOp**](http://docs.google.com/org/jibble/pircbot/PircBot.html#deOp(java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) nick)            Removes operator privilidges from a user on a channel. |
| void | [**deVoice**](http://docs.google.com/org/jibble/pircbot/PircBot.html#deVoice(java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) nick)            Removes voice privilidges from a user on a channel. |
| void | [**disconnect**](http://docs.google.com/org/jibble/pircbot/PircBot.html#disconnect())()            This method disconnects from the server cleanly by calling the quitServer() method. |
| void | [**dispose**](http://docs.google.com/org/jibble/pircbot/PircBot.html#dispose())()            Disposes of all thread resources used by this PircBot. |
| boolean | [**equals**](http://docs.google.com/org/jibble/pircbot/PircBot.html#equals(java.lang.Object))([Object](http://java.sun.com/j2se/1.4/docs/api/java/lang/Object.html) o)            Returns true if and only if the object being compared is the exact same instance as this PircBot. |
| [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html)[] | [**getChannels**](http://docs.google.com/org/jibble/pircbot/PircBot.html#getChannels())()            Returns an array of all channels that we are in. |
| [InetAddress](http://java.sun.com/j2se/1.4/docs/api/java/net/InetAddress.html) | [**getDccInetAddress**](http://docs.google.com/org/jibble/pircbot/PircBot.html#getDccInetAddress())()            Returns the InetAddress used when sending DCC chat or file transfers. |
| int[] | [**getDccPorts**](http://docs.google.com/org/jibble/pircbot/PircBot.html#getDccPorts())()            Returns the set of port numbers to be used when sending a DCC chat or file transfer. |
| [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) | [**getEncoding**](http://docs.google.com/org/jibble/pircbot/PircBot.html#getEncoding())()            Returns the encoding used to send and receive lines from the IRC server, or null if not set. |
| [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) | [**getFinger**](http://docs.google.com/org/jibble/pircbot/PircBot.html#getFinger())()            Gets the internal finger message of the PircBot. |
| [InetAddress](http://java.sun.com/j2se/1.4/docs/api/java/net/InetAddress.html) | [**getInetAddress**](http://docs.google.com/org/jibble/pircbot/PircBot.html#getInetAddress())()            Returns the InetAddress used by the PircBot. |
| [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) | [**getLogin**](http://docs.google.com/org/jibble/pircbot/PircBot.html#getLogin())()            Gets the internal login of the PircBot. |
| int | [**getMaxLineLength**](http://docs.google.com/org/jibble/pircbot/PircBot.html#getMaxLineLength())()            Gets the maximum length of any line that is sent via the IRC protocol. |
| long | [**getMessageDelay**](http://docs.google.com/org/jibble/pircbot/PircBot.html#getMessageDelay())()            Returns the number of milliseconds that will be used to separate consecutive messages to the server from the outgoing message queue. |
| [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) | [**getName**](http://docs.google.com/org/jibble/pircbot/PircBot.html#getName())()            Gets the name of the PircBot. |
| [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) | [**getNick**](http://docs.google.com/org/jibble/pircbot/PircBot.html#getNick())()            Returns the current nick of the bot. |
| int | [**getOutgoingQueueSize**](http://docs.google.com/org/jibble/pircbot/PircBot.html#getOutgoingQueueSize())()            Gets the number of lines currently waiting in the outgoing message Queue. |
| [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) | [**getPassword**](http://docs.google.com/org/jibble/pircbot/PircBot.html#getPassword())()            Returns the last password that we used when connecting to an IRC server. |
| int | [**getPort**](http://docs.google.com/org/jibble/pircbot/PircBot.html#getPort())()            Returns the port number of the last IRC server that the PircBot tried to connect to. |
| [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) | [**getServer**](http://docs.google.com/org/jibble/pircbot/PircBot.html#getServer())()            Returns the name of the last IRC server the PircBot tried to connect to. |
| [User](http://docs.google.com/org/jibble/pircbot/User.html)[] | [**getUsers**](http://docs.google.com/org/jibble/pircbot/PircBot.html#getUsers(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel)            Returns an array of all users in the specified channel. |
| [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) | [**getVersion**](http://docs.google.com/org/jibble/pircbot/PircBot.html#getVersion())()            Gets the internal version of the PircBot. |
| protected  void | [**handleLine**](http://docs.google.com/org/jibble/pircbot/PircBot.html#handleLine(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) line)            This method handles events when any line of text arrives from the server, then calling the appropriate method in the PircBot. |
| int | [**hashCode**](http://docs.google.com/org/jibble/pircbot/PircBot.html#hashCode())()            Returns the hashCode of this PircBot. |
| void | [**identify**](http://docs.google.com/org/jibble/pircbot/PircBot.html#identify(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) password)            Identify the bot with NickServ, supplying the appropriate password. |
| long | [**ipToLong**](http://docs.google.com/org/jibble/pircbot/PircBot.html#ipToLong(byte[]))(byte[] address)            A convenient method that accepts an IP address represented by a byte[] of size 4 and returns this as a long representation of the same IP address. |
| boolean | [**isConnected**](http://docs.google.com/org/jibble/pircbot/PircBot.html#isConnected())()            Returns whether or not the PircBot is currently connected to a server. |
| void | [**joinChannel**](http://docs.google.com/org/jibble/pircbot/PircBot.html#joinChannel(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel)            Joins a channel. |
| void | [**joinChannel**](http://docs.google.com/org/jibble/pircbot/PircBot.html#joinChannel(java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) key)            Joins a channel with a key. |
| void | [**kick**](http://docs.google.com/org/jibble/pircbot/PircBot.html#kick(java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) nick)            Kicks a user from a channel. |
| void | [**kick**](http://docs.google.com/org/jibble/pircbot/PircBot.html#kick(java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) nick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) reason)            Kicks a user from a channel, giving a reason. |
| void | [**listChannels**](http://docs.google.com/org/jibble/pircbot/PircBot.html#listChannels())()            Issues a request for a list of all channels on the IRC server. |
| void | [**listChannels**](http://docs.google.com/org/jibble/pircbot/PircBot.html#listChannels(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) parameters)            Issues a request for a list of all channels on the IRC server. |
| void | [**log**](http://docs.google.com/org/jibble/pircbot/PircBot.html#log(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) line)            Adds a line to the log. |
| int[] | [**longToIp**](http://docs.google.com/org/jibble/pircbot/PircBot.html#longToIp(long))(long address)            A convenient method that accepts an IP address represented as a long and returns an integer array of size 4 representing the same IP address. |
| protected  void | [**onAction**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onAction(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sender, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) login, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) action)            This method is called whenever an ACTION is sent from a user. |
| protected  void | [**onChannelInfo**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onChannelInfo(java.lang.String,%20int,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, int userCount, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) topic)            After calling the listChannels() method in PircBot, the server will start to send us information about each channel on the server. |
| protected  void | [**onConnect**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onConnect())()            This method is called once the PircBot has successfully connected to the IRC server. |
| protected  void | [**onDccChatRequest**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onDccChatRequest(java.lang.String,%20java.lang.String,%20java.lang.String,%20long,%20int))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, long address, int port)  **Deprecated.** *As of PircBot 1.2.0, use* [*onIncomingChatRequest(DccChat)*](http://docs.google.com/org/jibble/pircbot/PircBot.html#onIncomingChatRequest(org.jibble.pircbot.DccChat)) |
| protected  void | [**onDccSendRequest**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onDccSendRequest(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20long,%20int,%20int))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) filename, long address, int port, int size)  **Deprecated.** *As of PircBot 1.2.0, use* [*onIncomingFileTransfer(DccFileTransfer)*](http://docs.google.com/org/jibble/pircbot/PircBot.html#onIncomingFileTransfer(org.jibble.pircbot.DccFileTransfer)) |
| protected  void | [**onDeop**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onDeop(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) recipient)            Called when a user (possibly us) gets operator status taken away. |
| protected  void | [**onDeVoice**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onDeVoice(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) recipient)            Called when a user (possibly us) gets voice status removed. |
| protected  void | [**onDisconnect**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onDisconnect())()            This method carries out the actions to be performed when the PircBot gets disconnected. |
| protected  void | [**onFileTransferFinished**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onFileTransferFinished(org.jibble.pircbot.DccFileTransfer,%20java.lang.Exception))([DccFileTransfer](http://docs.google.com/org/jibble/pircbot/DccFileTransfer.html) transfer, [Exception](http://java.sun.com/j2se/1.4/docs/api/java/lang/Exception.html) e)            This method gets called when a DccFileTransfer has finished. |
| protected  void | [**onFinger**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onFinger(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target)            This method is called whenever we receive a FINGER request. |
| protected  void | [**onIncomingChatRequest**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onIncomingChatRequest(org.jibble.pircbot.DccChat))([DccChat](http://docs.google.com/org/jibble/pircbot/DccChat.html) chat)            This method will be called whenever a DCC Chat request is received. |
| protected  void | [**onIncomingFileTransfer**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onIncomingFileTransfer(org.jibble.pircbot.DccFileTransfer))([DccFileTransfer](http://docs.google.com/org/jibble/pircbot/DccFileTransfer.html) transfer)            This method is called whenever a DCC SEND request is sent to the PircBot. |
| protected  void | [**onInvite**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onInvite(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) targetNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel)            Called when we are invited to a channel by a user. |
| protected  void | [**onJoin**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onJoin(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sender, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) login, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostname)            This method is called whenever someone (possibly us) joins a channel which we are on. |
| protected  void | [**onKick**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onKick(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) kickerNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) kickerLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) kickerHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) recipientNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) reason)            This method is called whenever someone (possibly us) is kicked from any of the channels that we are in. |
| protected  void | [**onMessage**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onMessage(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sender, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) login, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) message)            This method is called whenever a message is sent to a channel. |
| protected  void | [**onMode**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onMode(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) mode)            Called when the mode of a channel is set. |
| protected  void | [**onNickChange**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onNickChange(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) oldNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) login, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) newNick)            This method is called whenever someone (possibly us) changes nick on any of the channels that we are on. |
| protected  void | [**onNotice**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onNotice(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) notice)            This method is called whenever we receive a notice. |
| protected  void | [**onOp**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onOp(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) recipient)            Called when a user (possibly us) gets granted operator status for a channel. |
| protected  void | [**onPart**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onPart(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sender, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) login, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostname)            This method is called whenever someone (possibly us) parts a channel which we are on. |
| protected  void | [**onPing**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onPing(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) pingValue)            This method is called whenever we receive a PING request from another user. |
| protected  void | [**onPrivateMessage**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onPrivateMessage(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sender, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) login, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) message)            This method is called whenever a private message is sent to the PircBot. |
| protected  void | [**onQuit**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onQuit(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) reason)            This method is called whenever someone (possibly us) quits from the server. |
| protected  void | [**onRemoveChannelBan**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onRemoveChannelBan(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostmask)            Called when a hostmask ban is removed from a channel. |
| protected  void | [**onRemoveChannelKey**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onRemoveChannelKey(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) key)            Called when a channel key is removed. |
| protected  void | [**onRemoveChannelLimit**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onRemoveChannelLimit(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)            Called when the user limit is removed for a channel. |
| protected  void | [**onRemoveInviteOnly**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onRemoveInviteOnly(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)            Called when a channel has 'invite only' removed. |
| protected  void | [**onRemoveModerated**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onRemoveModerated(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)            Called when a channel has moderated mode removed. |
| protected  void | [**onRemoveNoExternalMessages**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onRemoveNoExternalMessages(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)            Called when a channel is set to allow messages from any user, even if they are not actually in the channel. |
| protected  void | [**onRemovePrivate**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onRemovePrivate(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)            Called when a channel is marked as not being in private mode. |
| protected  void | [**onRemoveSecret**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onRemoveSecret(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)            Called when a channel has 'secret' mode removed. |
| protected  void | [**onRemoveTopicProtection**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onRemoveTopicProtection(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)            Called when topic protection is removed for a channel. |
| protected  void | [**onServerPing**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onServerPing(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) response)            The actions to perform when a PING request comes from the server. |
| protected  void | [**onServerResponse**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onServerResponse(int,%20java.lang.String))(int code, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) response)            This method is called when we receive a numeric response from the IRC server. |
| protected  void | [**onSetChannelBan**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onSetChannelBan(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostmask)            Called when a user (possibly us) gets banned from a channel. |
| protected  void | [**onSetChannelKey**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onSetChannelKey(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) key)            Called when a channel key is set. |
| protected  void | [**onSetChannelLimit**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onSetChannelLimit(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20int))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, int limit)            Called when a user limit is set for a channel. |
| protected  void | [**onSetInviteOnly**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onSetInviteOnly(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)            Called when a channel is set to 'invite only' mode. |
| protected  void | [**onSetModerated**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onSetModerated(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)            Called when a channel is set to 'moderated' mode. |
| protected  void | [**onSetNoExternalMessages**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onSetNoExternalMessages(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)            Called when a channel is set to only allow messages from users that are in the channel. |
| protected  void | [**onSetPrivate**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onSetPrivate(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)            Called when a channel is marked as being in private mode. |
| protected  void | [**onSetSecret**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onSetSecret(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)            Called when a channel is set to be in 'secret' mode. |
| protected  void | [**onSetTopicProtection**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onSetTopicProtection(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)            Called when topic protection is enabled for a channel. |
| protected  void | [**onTime**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onTime(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target)            This method is called whenever we receive a TIME request. |
| protected  void | [**onTopic**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onTopic(java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) topic)  **Deprecated.** *As of 1.2.0, replaced by* [*onTopic(String,String,String,long,boolean)*](http://docs.google.com/org/jibble/pircbot/PircBot.html#onTopic(java.lang.String,%20java.lang.String,%20java.lang.String,%20long,%20boolean)) |
| protected  void | [**onTopic**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onTopic(java.lang.String,%20java.lang.String,%20java.lang.String,%20long,%20boolean))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) topic, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) setBy, long date, boolean changed)            This method is called whenever a user sets the topic, or when PircBot joins a new channel and discovers its topic. |
| protected  void | [**onUnknown**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onUnknown(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) line)            This method is called whenever we receive a line from the server that the PircBot has not been programmed to recognise. |
| protected  void | [**onUserList**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onUserList(java.lang.String,%20org.jibble.pircbot.User[]))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [User](http://docs.google.com/org/jibble/pircbot/User.html)[] users)            This method is called when we receive a user list from the server after joining a channel. |
| protected  void | [**onUserMode**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onUserMode(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) targetNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) mode)            Called when the mode of a user is set. |
| protected  void | [**onVersion**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onVersion(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target)            This method is called whenever we receive a VERSION request. |
| protected  void | [**onVoice**](http://docs.google.com/org/jibble/pircbot/PircBot.html#onVoice(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) recipient)            Called when a user (possibly us) gets voice status granted in a channel. |
| void | [**op**](http://docs.google.com/org/jibble/pircbot/PircBot.html#op(java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) nick)            Grants operator privilidges to a user on a channel. |
| void | [**partChannel**](http://docs.google.com/org/jibble/pircbot/PircBot.html#partChannel(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel)            Parts a channel. |
| void | [**partChannel**](http://docs.google.com/org/jibble/pircbot/PircBot.html#partChannel(java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) reason)            Parts a channel, giving a reason. |
| void | [**quitServer**](http://docs.google.com/org/jibble/pircbot/PircBot.html#quitServer())()            Quits from the IRC server. |
| void | [**quitServer**](http://docs.google.com/org/jibble/pircbot/PircBot.html#quitServer(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) reason)            Quits from the IRC server with a reason. |
| void | [**reconnect**](http://docs.google.com/org/jibble/pircbot/PircBot.html#reconnect())()            Reconnects to the IRC server that we were previously connected to. |
| void | [**sendAction**](http://docs.google.com/org/jibble/pircbot/PircBot.html#sendAction(java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) action)            Sends an action to the channel or to a user. |
| void | [**sendCTCPCommand**](http://docs.google.com/org/jibble/pircbot/PircBot.html#sendCTCPCommand(java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) command)            Sends a CTCP command to a channel or user. |
| void | [**sendInvite**](http://docs.google.com/org/jibble/pircbot/PircBot.html#sendInvite(java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) nick, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel)            Sends an invitation to join a channel. |
| void | [**sendMessage**](http://docs.google.com/org/jibble/pircbot/PircBot.html#sendMessage(java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) message)            Sends a message to a channel or a private message to a user. |
| void | [**sendNotice**](http://docs.google.com/org/jibble/pircbot/PircBot.html#sendNotice(java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) notice)            Sends a notice to the channel or to a user. |
| void | [**sendRawLine**](http://docs.google.com/org/jibble/pircbot/PircBot.html#sendRawLine(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) line)            Sends a raw line to the IRC server as soon as possible, bypassing the outgoing message queue. |
| void | [**sendRawLineViaQueue**](http://docs.google.com/org/jibble/pircbot/PircBot.html#sendRawLineViaQueue(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) line)            Sends a raw line through the outgoing message queue. |
| void | [**setAutoNickChange**](http://docs.google.com/org/jibble/pircbot/PircBot.html#setAutoNickChange(boolean))(boolean autoNickChange)            When you connect to a server and your nick is already in use and this is set to true, a new nick will be automatically chosen. |
| void | [**setDccInetAddress**](http://docs.google.com/org/jibble/pircbot/PircBot.html#setDccInetAddress(java.net.InetAddress))([InetAddress](http://java.sun.com/j2se/1.4/docs/api/java/net/InetAddress.html) dccInetAddress)            Sets the InetAddress to be used when sending DCC chat or file transfers. |
| void | [**setDccPorts**](http://docs.google.com/org/jibble/pircbot/PircBot.html#setDccPorts(int[]))(int[] ports)            Sets the choice of port numbers that can be used when sending a DCC chat or file transfer. |
| void | [**setEncoding**](http://docs.google.com/org/jibble/pircbot/PircBot.html#setEncoding(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) charset)            Sets the encoding charset to be used when sending or receiving lines from the IRC server. |
| protected  void | [**setFinger**](http://docs.google.com/org/jibble/pircbot/PircBot.html#setFinger(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) finger)            Sets the interal finger message. |
| protected  void | [**setLogin**](http://docs.google.com/org/jibble/pircbot/PircBot.html#setLogin(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) login)            Sets the internal login of the Bot. |
| void | [**setMessageDelay**](http://docs.google.com/org/jibble/pircbot/PircBot.html#setMessageDelay(long))(long delay)            Sets the number of milliseconds to delay between consecutive messages when there are multiple messages waiting in the outgoing message queue. |
| void | [**setMode**](http://docs.google.com/org/jibble/pircbot/PircBot.html#setMode(java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) mode)            Set the mode of a channel. |
| protected  void | [**setName**](http://docs.google.com/org/jibble/pircbot/PircBot.html#setName(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) name)            Sets the name of the bot, which will be used as its nick when it tries to join an IRC server. |
| void | [**setTopic**](http://docs.google.com/org/jibble/pircbot/PircBot.html#setTopic(java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) topic)            Set the topic for a channel. |
| void | [**setVerbose**](http://docs.google.com/org/jibble/pircbot/PircBot.html#setVerbose(boolean))(boolean verbose)            Sets the verbose mode. |
| protected  void | [**setVersion**](http://docs.google.com/org/jibble/pircbot/PircBot.html#setVersion(java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) version)            Sets the internal version of the Bot. |
| void | [**startIdentServer**](http://docs.google.com/org/jibble/pircbot/PircBot.html#startIdentServer())()            Starts an ident server (Identification Protocol Server, RFC 1413). |
| [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) | [**toString**](http://docs.google.com/org/jibble/pircbot/PircBot.html#toString())()            Returns a String representation of this object. |
| void | [**unBan**](http://docs.google.com/org/jibble/pircbot/PircBot.html#unBan(java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostmask)            Unbans a user from a channel. |
| void | [**voice**](http://docs.google.com/org/jibble/pircbot/PircBot.html#voice(java.lang.String,%20java.lang.String))([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel, [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) nick)            Grants voice privilidges to a user on a channel. |

|  |
| --- |
| **Methods inherited from class java.lang.**[**Object**](http://java.sun.com/j2se/1.4/docs/api/java/lang/Object.html) |
| [clone](http://java.sun.com/j2se/1.4/docs/api/java/lang/Object.html#clone()), [finalize](http://java.sun.com/j2se/1.4/docs/api/java/lang/Object.html#finalize()), [getClass](http://java.sun.com/j2se/1.4/docs/api/java/lang/Object.html#getClass()), [notify](http://java.sun.com/j2se/1.4/docs/api/java/lang/Object.html#notify()), [notifyAll](http://java.sun.com/j2se/1.4/docs/api/java/lang/Object.html#notifyAll()), [wait](http://java.sun.com/j2se/1.4/docs/api/java/lang/Object.html#wait()), [wait](http://java.sun.com/j2se/1.4/docs/api/java/lang/Object.html#wait(long)), [wait](http://java.sun.com/j2se/1.4/docs/api/java/lang/Object.html#wait(long,%20int)) |

|  |
| --- |
| **Field Detail** |

### VERSION

public static final [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) **VERSION**

The definitive version number of this release of PircBot. (Note: Change this before automatically building releases)

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#org.jibble.pircbot.PircBot.VERSION)

|  |
| --- |
| **Constructor Detail** |

### PircBot

public **PircBot**()

Constructs a PircBot with the default settings. Your own constructors in classes which extend the PircBot abstract class should be responsible for changing the default settings if required.

**Method Detail**

|  |
| --- |
| **Method Detail** |

### connect

public final void **connect**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostname)  
 throws [IOException](http://java.sun.com/j2se/1.4/docs/api/java/io/IOException.html),  
 [IrcException](http://docs.google.com/org/jibble/pircbot/IrcException.html),  
 [NickAlreadyInUseException](http://docs.google.com/org/jibble/pircbot/NickAlreadyInUseException.html)

Attempt to connect to the specified IRC server. The onConnect method is called upon success.

**Parameters:**hostname - The hostname of the server to connect to. **Throws:** [IOException](http://java.sun.com/j2se/1.4/docs/api/java/io/IOException.html) - if it was not possible to connect to the server. [IrcException](http://docs.google.com/org/jibble/pircbot/IrcException.html) - if the server would not let us join it. [NickAlreadyInUseException](http://docs.google.com/org/jibble/pircbot/NickAlreadyInUseException.html) - if our nick is already in use on the server.

### connect

public final void **connect**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostname,  
 int port)  
 throws [IOException](http://java.sun.com/j2se/1.4/docs/api/java/io/IOException.html),  
 [IrcException](http://docs.google.com/org/jibble/pircbot/IrcException.html),  
 [NickAlreadyInUseException](http://docs.google.com/org/jibble/pircbot/NickAlreadyInUseException.html)

Attempt to connect to the specified IRC server and port number. The onConnect method is called upon success.

**Parameters:**hostname - The hostname of the server to connect to.port - The port number to connect to on the server. **Throws:** [IOException](http://java.sun.com/j2se/1.4/docs/api/java/io/IOException.html) - if it was not possible to connect to the server. [IrcException](http://docs.google.com/org/jibble/pircbot/IrcException.html) - if the server would not let us join it. [NickAlreadyInUseException](http://docs.google.com/org/jibble/pircbot/NickAlreadyInUseException.html) - if our nick is already in use on the server.

### connect

public final void **connect**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostname,  
 int port,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) password)  
 throws [IOException](http://java.sun.com/j2se/1.4/docs/api/java/io/IOException.html),  
 [IrcException](http://docs.google.com/org/jibble/pircbot/IrcException.html),  
 [NickAlreadyInUseException](http://docs.google.com/org/jibble/pircbot/NickAlreadyInUseException.html)

Attempt to connect to the specified IRC server using the supplied password. The onConnect method is called upon success.

**Parameters:**hostname - The hostname of the server to connect to.port - The port number to connect to on the server.password - The password to use to join the server. **Throws:** [IOException](http://java.sun.com/j2se/1.4/docs/api/java/io/IOException.html) - if it was not possible to connect to the server. [IrcException](http://docs.google.com/org/jibble/pircbot/IrcException.html) - if the server would not let us join it. [NickAlreadyInUseException](http://docs.google.com/org/jibble/pircbot/NickAlreadyInUseException.html) - if our nick is already in use on the server.

### reconnect

public final void **reconnect**()  
 throws [IOException](http://java.sun.com/j2se/1.4/docs/api/java/io/IOException.html),  
 [IrcException](http://docs.google.com/org/jibble/pircbot/IrcException.html),  
 [NickAlreadyInUseException](http://docs.google.com/org/jibble/pircbot/NickAlreadyInUseException.html)

Reconnects to the IRC server that we were previously connected to. If necessary, the appropriate port number and password will be used. This method will throw an IrcException if we have never connected to an IRC server previously.

**Throws:** [IOException](http://java.sun.com/j2se/1.4/docs/api/java/io/IOException.html) - if it was not possible to connect to the server. [IrcException](http://docs.google.com/org/jibble/pircbot/IrcException.html) - if the server would not let us join it. [NickAlreadyInUseException](http://docs.google.com/org/jibble/pircbot/NickAlreadyInUseException.html) - if our nick is already in use on the server.**Since:** PircBot 0.9.9

### disconnect

public final void **disconnect**()

This method disconnects from the server cleanly by calling the quitServer() method. Providing the PircBot was connected to an IRC server, the onDisconnect() will be called as soon as the disconnection is made by the server.

**See Also:**[quitServer](http://docs.google.com/org/jibble/pircbot/PircBot.html#quitServer()), [quitServer](http://docs.google.com/org/jibble/pircbot/PircBot.html#quitServer(java.lang.String))

### setAutoNickChange

public void **setAutoNickChange**(boolean autoNickChange)

When you connect to a server and your nick is already in use and this is set to true, a new nick will be automatically chosen. This is done by adding numbers to the end of the nick until an available nick is found.

**Parameters:**autoNickChange - Set to true if you want automatic nick changes during connection.

### startIdentServer

public final void **startIdentServer**()

Starts an ident server (Identification Protocol Server, RFC 1413).

Most IRC servers attempt to contact the ident server on connecting hosts in order to determine the user's identity. A few IRC servers will not allow you to connect unless this information is provided.

So when a PircBot is run on a machine that does not run an ident server, it may be necessary to call this method to start one up.

Calling this method starts up an ident server which will respond with the login provided by calling getLogin() and then shut down immediately. It will also be shut down if it has not been contacted within 60 seconds of creation.

If you require an ident response, then the correct procedure is to start the ident server and then connect to the IRC server. The IRC server may then contact the ident server to get the information it needs.

The ident server will fail to start if there is already an ident server running on port 113, or if you are running as an unprivileged user who is unable to create a server socket on that port number.

If it is essential for you to use an ident server when connecting to an IRC server, then make sure that port 113 on your machine is visible to the IRC server so that it may contact the ident server.

**Since:** PircBot 0.9c

### joinChannel

public final void **joinChannel**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel)

Joins a channel.

**Parameters:**channel - The name of the channel to join (eg "#cs").

### joinChannel

public final void **joinChannel**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) key)

Joins a channel with a key.

**Parameters:**channel - The name of the channel to join (eg "#cs").key - The key that will be used to join the channel.

### partChannel

public final void **partChannel**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel)

Parts a channel.

**Parameters:**channel - The name of the channel to leave.

### partChannel

public final void **partChannel**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) reason)

Parts a channel, giving a reason.

**Parameters:**channel - The name of the channel to leave.reason - The reason for parting the channel.

### quitServer

public final void **quitServer**()

Quits from the IRC server. Providing we are actually connected to an IRC server, the onDisconnect() method will be called as soon as the IRC server disconnects us.

### quitServer

public final void **quitServer**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) reason)

Quits from the IRC server with a reason. Providing we are actually connected to an IRC server, the onDisconnect() method will be called as soon as the IRC server disconnects us.

**Parameters:**reason - The reason for quitting the server.

### sendRawLine

public final void **sendRawLine**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) line)

Sends a raw line to the IRC server as soon as possible, bypassing the outgoing message queue.

**Parameters:**line - The raw line to send to the IRC server.

### sendRawLineViaQueue

public final void **sendRawLineViaQueue**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) line)

Sends a raw line through the outgoing message queue.

**Parameters:**line - The raw line to send to the IRC server.

### sendMessage

public final void **sendMessage**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) message)

Sends a message to a channel or a private message to a user. These messages are added to the outgoing message queue and sent at the earliest possible opportunity.

Some examples: -

// Send the message "Hello!" to the channel #cs.  
 sendMessage("#cs", "Hello!");  
   
 // Send a private message to Paul that says "Hi".  
 sendMessage("Paul", "Hi");

You may optionally apply colours, boldness, underlining, etc to the message by using the Colors class.

**Parameters:**target - The name of the channel or user nick to send to.message - The message to send.**See Also:**[Colors](http://docs.google.com/org/jibble/pircbot/Colors.html)

### sendAction

public final void **sendAction**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) action)

Sends an action to the channel or to a user.

**Parameters:**target - The name of the channel or user nick to send to.action - The action to send.**See Also:**[Colors](http://docs.google.com/org/jibble/pircbot/Colors.html)

### sendNotice

public final void **sendNotice**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) notice)

Sends a notice to the channel or to a user.

**Parameters:**target - The name of the channel or user nick to send to.notice - The notice to send.

### sendCTCPCommand

public final void **sendCTCPCommand**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) command)

Sends a CTCP command to a channel or user. (Client to client protocol). Examples of such commands are "PING ", "FINGER", "VERSION", etc. For example, if you wish to request the version of a user called "Dave", then you would call sendCTCPCommand("Dave", "VERSION");. The type of response to such commands is largely dependant on the target client software.

**Parameters:**target - The name of the channel or user to send the CTCP message to.command - The CTCP command to send.**Since:** PircBot 0.9.5

### changeNick

public final void **changeNick**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) newNick)

Attempt to change the current nick (nickname) of the bot when it is connected to an IRC server. After confirmation of a successful nick change, the getNick method will return the new nick.

**Parameters:**newNick - The new nick to use.

### identify

public final void **identify**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) password)

Identify the bot with NickServ, supplying the appropriate password. Some IRC Networks (such as freenode) require users to *register* and *identify* with NickServ before they are able to send private messages to other users, thus reducing the amount of spam. If you are using an IRC network where this kind of policy is enforced, you will need to make your bot *identify* itself to NickServ before you can send private messages. Assuming you have already registered your bot's nick with NickServ, this method can be used to *identify* with the supplied password. It usually makes sense to identify with NickServ immediately after connecting to a server.

This method issues a raw NICKSERV command to the server, and is therefore safer than the alternative approach of sending a private message to NickServ. The latter approach is considered dangerous, as it may cause you to inadvertently transmit your password to an untrusted party if you connect to a network which does not run a NickServ service and where the untrusted party has assumed the nick "NickServ". However, if your IRC network is only compatible with the private message approach, you may typically identify like so:

sendMessage("NickServ", "identify PASSWORD");

**Parameters:**password - The password which will be used to identify with NickServ.

### setMode

public final void **setMode**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) mode)

Set the mode of a channel. This method attempts to set the mode of a channel. This may require the bot to have operator status on the channel. For example, if the bot has operator status, we can grant operator status to "Dave" on the #cs channel by calling setMode("#cs", "+o Dave"); An alternative way of doing this would be to use the op method.

**Parameters:**channel - The channel on which to perform the mode change.mode - The new mode to apply to the channel. This may include zero or more arguments if necessary.**See Also:**[op](http://docs.google.com/org/jibble/pircbot/PircBot.html#op(java.lang.String,%20java.lang.String))

### sendInvite

public final void **sendInvite**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) nick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel)

Sends an invitation to join a channel. Some channels can be marked as "invite-only", so it may be useful to allow a bot to invite people into it.

**Parameters:**nick - The nick of the user to invitechannel - The channel you are inviting the user to join.

### ban

public final void **ban**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostmask)

Bans a user from a channel. An example of a valid hostmask is "\*!\*compu@\*.18hp.net". This may be used in conjunction with the kick method to permanently remove a user from a channel. Successful use of this method may require the bot to have operator status itself.

**Parameters:**channel - The channel to ban the user from.hostmask - A hostmask representing the user we're banning.

### unBan

public final void **unBan**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostmask)

Unbans a user from a channel. An example of a valid hostmask is "\*!\*compu@\*.18hp.net". Successful use of this method may require the bot to have operator status itself.

**Parameters:**channel - The channel to unban the user from.hostmask - A hostmask representing the user we're unbanning.

### op

public final void **op**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) nick)

Grants operator privilidges to a user on a channel. Successful use of this method may require the bot to have operator status itself.

**Parameters:**channel - The channel we're opping the user on.nick - The nick of the user we are opping.

### deOp

public final void **deOp**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) nick)

Removes operator privilidges from a user on a channel. Successful use of this method may require the bot to have operator status itself.

**Parameters:**channel - The channel we're deopping the user on.nick - The nick of the user we are deopping.

### voice

public final void **voice**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) nick)

Grants voice privilidges to a user on a channel. Successful use of this method may require the bot to have operator status itself.

**Parameters:**channel - The channel we're voicing the user on.nick - The nick of the user we are voicing.

### deVoice

public final void **deVoice**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) nick)

Removes voice privilidges from a user on a channel. Successful use of this method may require the bot to have operator status itself.

**Parameters:**channel - The channel we're devoicing the user on.nick - The nick of the user we are devoicing.

### setTopic

public final void **setTopic**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) topic)

Set the topic for a channel. This method attempts to set the topic of a channel. This may require the bot to have operator status if the topic is protected.

**Parameters:**channel - The channel on which to perform the mode change.topic - The new topic for the channel.

### kick

public final void **kick**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) nick)

Kicks a user from a channel. This method attempts to kick a user from a channel and may require the bot to have operator status in the channel.

**Parameters:**channel - The channel to kick the user from.nick - The nick of the user to kick.

### kick

public final void **kick**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) nick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) reason)

Kicks a user from a channel, giving a reason. This method attempts to kick a user from a channel and may require the bot to have operator status in the channel.

**Parameters:**channel - The channel to kick the user from.nick - The nick of the user to kick.reason - A description of the reason for kicking a user.

### listChannels

public final void **listChannels**()

Issues a request for a list of all channels on the IRC server. When the PircBot receives information for each channel, it will call the onChannelInfo method, which you will need to override if you want it to do anything useful.

**See Also:**[onChannelInfo](http://docs.google.com/org/jibble/pircbot/PircBot.html#onChannelInfo(java.lang.String,%20int,%20java.lang.String))

### listChannels

public final void **listChannels**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) parameters)

Issues a request for a list of all channels on the IRC server. When the PircBot receives information for each channel, it will call the onChannelInfo method, which you will need to override if you want it to do anything useful.

Some IRC servers support certain parameters for LIST requests. One example is a parameter of ">10" to list only those channels that have more than 10 users in them. Whether these parameters are supported or not will depend on the IRC server software.

**Parameters:**parameters - The parameters to supply when requesting the list.**See Also:**[onChannelInfo](http://docs.google.com/org/jibble/pircbot/PircBot.html#onChannelInfo(java.lang.String,%20int,%20java.lang.String))

### dccSendFile

public final [DccFileTransfer](http://docs.google.com/org/jibble/pircbot/DccFileTransfer.html) **dccSendFile**([File](http://java.sun.com/j2se/1.4/docs/api/java/io/File.html) file,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) nick,  
 int timeout)

Sends a file to another user. Resuming is supported. The other user must be able to connect directly to your bot to be able to receive the file.

You may throttle the speed of this file transfer by calling the setPacketDelay method on the DccFileTransfer that is returned.

This method may not be overridden.

**Parameters:**file - The file to send.nick - The user to whom the file is to be sent.timeout - The number of milliseconds to wait for the recipient to acccept the file (we recommend about 120000). **Returns:**The DccFileTransfer that can be used to monitor this transfer.**Since:** 0.9c **See Also:**[DccFileTransfer](http://docs.google.com/org/jibble/pircbot/DccFileTransfer.html)

### dccReceiveFile

protected final void **dccReceiveFile**([File](http://java.sun.com/j2se/1.4/docs/api/java/io/File.html) file,  
 long address,  
 int port,  
 int size)

**Deprecated.** *As of PircBot 1.2.0, use* [*onIncomingFileTransfer(DccFileTransfer)*](http://docs.google.com/org/jibble/pircbot/PircBot.html#onIncomingFileTransfer(org.jibble.pircbot.DccFileTransfer))

Receives a file that is being sent to us by a DCC SEND request. Please use the onIncomingFileTransfer method to receive files.

### dccSendChatRequest

public final [DccChat](http://docs.google.com/org/jibble/pircbot/DccChat.html) **dccSendChatRequest**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) nick,  
 int timeout)

Attempts to establish a DCC CHAT session with a client. This method issues the connection request to the client and then waits for the client to respond. If the connection is successfully made, then a DccChat object is returned by this method. If the connection is not made within the time limit specified by the timeout value, then null is returned.

It is **strongly recommended** that you call this method within a new Thread, as it may take a long time to return.

This method may not be overridden.

**Parameters:**nick - The nick of the user we are trying to establish a chat with.timeout - The number of milliseconds to wait for the recipient to accept the chat connection (we recommend about 120000). **Returns:**a DccChat object that can be used to send and recieve lines of text. Returns **null** if the connection could not be made.**Since:** PircBot 0.9.8 **See Also:**[DccChat](http://docs.google.com/org/jibble/pircbot/DccChat.html)

### dccAcceptChatRequest

protected final [DccChat](http://docs.google.com/org/jibble/pircbot/DccChat.html) **dccAcceptChatRequest**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 long address,  
 int port)

**Deprecated.** *As of PircBot 1.2.0, use* [*onIncomingChatRequest(DccChat)*](http://docs.google.com/org/jibble/pircbot/PircBot.html#onIncomingChatRequest(org.jibble.pircbot.DccChat))

Attempts to accept a DCC CHAT request by a client. Please use the onIncomingChatRequest method to receive files.

### log

public void **log**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) line)

Adds a line to the log. This log is currently output to the standard output and is in the correct format for use by tools such as pisg, the Perl IRC Statistics Generator. You may override this method if you wish to do something else with log entries. Each line in the log begins with a number which represents the logging time (as the number of milliseconds since the epoch). This timestamp and the following log entry are separated by a single space character, " ". Outgoing messages are distinguishable by a log entry that has ">>>" immediately following the space character after the timestamp. DCC events use "+++" and warnings about unhandled Exceptions and Errors use "###".

This implementation of the method will only cause log entries to be output if the PircBot has had its verbose mode turned on by calling setVerbose(true);

**Parameters:**line - The line to add to the log.

### handleLine

protected void **handleLine**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) line)

This method handles events when any line of text arrives from the server, then calling the appropriate method in the PircBot. This method is protected and only called by the InputThread for this instance.

This method may not be overridden!

**Parameters:**line - The raw line of text from the server.

### onConnect

protected void **onConnect**()

This method is called once the PircBot has successfully connected to the IRC server.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Since:** PircBot 0.9.6

### onDisconnect

protected void **onDisconnect**()

This method carries out the actions to be performed when the PircBot gets disconnected. This may happen if the PircBot quits from the server, or if the connection is unexpectedly lost.

Disconnection from the IRC server is detected immediately if either we or the server close the connection normally. If the connection to the server is lost, but neither we nor the server have explicitly closed the connection, then it may take a few minutes to detect (this is commonly referred to as a "ping timeout").

If you wish to get your IRC bot to automatically rejoin a server after the connection has been lost, then this is probably the ideal method to override to implement such functionality.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

### onServerResponse

protected void **onServerResponse**(int code,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) response)

This method is called when we receive a numeric response from the IRC server.

Numerics in the range from 001 to 099 are used for client-server connections only and should never travel between servers. Replies generated in response to commands are found in the range from 200 to 399. Error replies are found in the range from 400 to 599.

For example, we can use this method to discover the topic of a channel when we join it. If we join the channel #test which has a topic of "I am King of Test" then the response will be "PircBot #test :I Am King of Test" with a code of 332 to signify that this is a topic. (This is just an example - note that overriding the onTopic method is an easier way of finding the topic for a channel). Check the IRC RFC for the full list of other command response codes.

PircBot implements the interface ReplyConstants, which contains contstants that you may find useful here.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**code - The three-digit numerical code for the response.response - The full response from the IRC server.**See Also:**[ReplyConstants](http://docs.google.com/org/jibble/pircbot/ReplyConstants.html)

### onUserList

protected void **onUserList**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [User](http://docs.google.com/org/jibble/pircbot/User.html)[] users)

This method is called when we receive a user list from the server after joining a channel.

Shortly after joining a channel, the IRC server sends a list of all users in that channel. The PircBot collects this information and calls this method as soon as it has the full list.

To obtain the nick of each user in the channel, call the getNick() method on each User object in the array.

At a later time, you may call the getUsers method to obtain an up to date list of the users in the channel.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The name of the channel.users - An array of User objects belonging to this channel.**Since:** PircBot 1.0.0 **See Also:**[User](http://docs.google.com/org/jibble/pircbot/User.html)

### onMessage

protected void **onMessage**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sender,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) login,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) message)

This method is called whenever a message is sent to a channel.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel to which the message was sent.sender - The nick of the person who sent the message.login - The login of the person who sent the message.hostname - The hostname of the person who sent the message.message - The actual message sent to the channel.

### onPrivateMessage

protected void **onPrivateMessage**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sender,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) login,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) message)

This method is called whenever a private message is sent to the PircBot.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**sender - The nick of the person who sent the private message.login - The login of the person who sent the private message.hostname - The hostname of the person who sent the private message.message - The actual message.

### onAction

protected void **onAction**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sender,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) login,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) action)

This method is called whenever an ACTION is sent from a user. E.g. such events generated by typing "/me goes shopping" in most IRC clients.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**sender - The nick of the user that sent the action.login - The login of the user that sent the action.hostname - The hostname of the user that sent the action.target - The target of the action, be it a channel or our nick.action - The action carried out by the user.

### onNotice

protected void **onNotice**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) notice)

This method is called whenever we receive a notice.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**sourceNick - The nick of the user that sent the notice.sourceLogin - The login of the user that sent the notice.sourceHostname - The hostname of the user that sent the notice.target - The target of the notice, be it our nick or a channel name.notice - The notice message.

### onJoin

protected void **onJoin**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sender,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) login,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostname)

This method is called whenever someone (possibly us) joins a channel which we are on.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel which somebody joined.sender - The nick of the user who joined the channel.login - The login of the user who joined the channel.hostname - The hostname of the user who joined the channel.

### onPart

protected void **onPart**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sender,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) login,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostname)

This method is called whenever someone (possibly us) parts a channel which we are on.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel which somebody parted from.sender - The nick of the user who parted from the channel.login - The login of the user who parted from the channel.hostname - The hostname of the user who parted from the channel.

### onNickChange

protected void **onNickChange**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) oldNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) login,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) newNick)

This method is called whenever someone (possibly us) changes nick on any of the channels that we are on.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**oldNick - The old nick.login - The login of the user.hostname - The hostname of the user.newNick - The new nick.

### onKick

protected void **onKick**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) kickerNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) kickerLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) kickerHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) recipientNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) reason)

This method is called whenever someone (possibly us) is kicked from any of the channels that we are in.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel from which the recipient was kicked.kickerNick - The nick of the user who performed the kick.kickerLogin - The login of the user who performed the kick.kickerHostname - The hostname of the user who performed the kick.recipientNick - The unfortunate recipient of the kick.reason - The reason given by the user who performed the kick.

### onQuit

protected void **onQuit**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) reason)

This method is called whenever someone (possibly us) quits from the server. We will only observe this if the user was in one of the channels to which we are connected.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**sourceNick - The nick of the user that quit from the server.sourceLogin - The login of the user that quit from the server.sourceHostname - The hostname of the user that quit from the server.reason - The reason given for quitting the server.

### onTopic

protected void **onTopic**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) topic)

**Deprecated.** *As of 1.2.0, replaced by* [*onTopic(String,String,String,long,boolean)*](http://docs.google.com/org/jibble/pircbot/PircBot.html#onTopic(java.lang.String,%20java.lang.String,%20java.lang.String,%20long,%20boolean))

This method is called whenever a user sets the topic, or when PircBot joins a new channel and discovers its topic.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel that the topic belongs to.topic - The topic for the channel.

### onTopic

protected void **onTopic**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) topic,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) setBy,  
 long date,  
 boolean changed)

This method is called whenever a user sets the topic, or when PircBot joins a new channel and discovers its topic.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel that the topic belongs to.topic - The topic for the channel.setBy - The nick of the user that set the topic.date - When the topic was set (milliseconds since the epoch).changed - True if the topic has just been changed, false if the topic was already there.

### onChannelInfo

protected void **onChannelInfo**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 int userCount,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) topic)

After calling the listChannels() method in PircBot, the server will start to send us information about each channel on the server. You may override this method in order to receive the information about each channel as soon as it is received.

Note that certain channels, such as those marked as hidden, may not appear in channel listings.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The name of the channel.userCount - The number of users visible in this channel.topic - The topic for this channel.**See Also:**[listChannels](http://docs.google.com/org/jibble/pircbot/PircBot.html#listChannels())

### onMode

protected void **onMode**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) mode)

Called when the mode of a channel is set.

You may find it more convenient to decode the meaning of the mode string by overriding the onOp, onDeOp, onVoice, onDeVoice, onChannelKey, onDeChannelKey, onChannelLimit, onDeChannelLimit, onChannelBan or onDeChannelBan methods as appropriate.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel that the mode operation applies to.sourceNick - The nick of the user that set the mode.sourceLogin - The login of the user that set the mode.sourceHostname - The hostname of the user that set the mode.mode - The mode that has been set.

### onUserMode

protected void **onUserMode**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) targetNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) mode)

Called when the mode of a user is set.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**targetNick - The nick that the mode operation applies to.sourceNick - The nick of the user that set the mode.sourceLogin - The login of the user that set the mode.sourceHostname - The hostname of the user that set the mode.mode - The mode that has been set.**Since:** PircBot 1.2.0

### onOp

protected void **onOp**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) recipient)

Called when a user (possibly us) gets granted operator status for a channel.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.recipient - The nick of the user that got 'opped'.**Since:** PircBot 0.9.5

### onDeop

protected void **onDeop**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) recipient)

Called when a user (possibly us) gets operator status taken away.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.recipient - The nick of the user that got 'deopped'.**Since:** PircBot 0.9.5

### onVoice

protected void **onVoice**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) recipient)

Called when a user (possibly us) gets voice status granted in a channel.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.recipient - The nick of the user that got 'voiced'.**Since:** PircBot 0.9.5

### onDeVoice

protected void **onDeVoice**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) recipient)

Called when a user (possibly us) gets voice status removed.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.recipient - The nick of the user that got 'devoiced'.**Since:** PircBot 0.9.5

### onSetChannelKey

protected void **onSetChannelKey**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) key)

Called when a channel key is set. When the channel key has been set, other users may only join that channel if they know the key. Channel keys are sometimes referred to as passwords.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.key - The new key for the channel.**Since:** PircBot 0.9.5

### onRemoveChannelKey

protected void **onRemoveChannelKey**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) key)

Called when a channel key is removed.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.key - The key that was in use before the channel key was removed.**Since:** PircBot 0.9.5

### onSetChannelLimit

protected void **onSetChannelLimit**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 int limit)

Called when a user limit is set for a channel. The number of users in the channel cannot exceed this limit.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.limit - The maximum number of users that may be in this channel at the same time.**Since:** PircBot 0.9.5

### onRemoveChannelLimit

protected void **onRemoveChannelLimit**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)

Called when the user limit is removed for a channel.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.**Since:** PircBot 0.9.5

### onSetChannelBan

protected void **onSetChannelBan**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostmask)

Called when a user (possibly us) gets banned from a channel. Being banned from a channel prevents any user with a matching hostmask from joining the channel. For this reason, most bans are usually directly followed by the user being kicked :-)

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.hostmask - The hostmask of the user that has been banned.**Since:** PircBot 0.9.5

### onRemoveChannelBan

protected void **onRemoveChannelBan**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) hostmask)

Called when a hostmask ban is removed from a channel.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.hostmask - **Since:** PircBot 0.9.5

### onSetTopicProtection

protected void **onSetTopicProtection**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)

Called when topic protection is enabled for a channel. Topic protection means that only operators in a channel may change the topic.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.**Since:** PircBot 0.9.5

### onRemoveTopicProtection

protected void **onRemoveTopicProtection**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)

Called when topic protection is removed for a channel.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.**Since:** PircBot 0.9.5

### onSetNoExternalMessages

protected void **onSetNoExternalMessages**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)

Called when a channel is set to only allow messages from users that are in the channel.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.**Since:** PircBot 0.9.5

### onRemoveNoExternalMessages

protected void **onRemoveNoExternalMessages**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)

Called when a channel is set to allow messages from any user, even if they are not actually in the channel.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.**Since:** PircBot 0.9.5

### onSetInviteOnly

protected void **onSetInviteOnly**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)

Called when a channel is set to 'invite only' mode. A user may only join the channel if they are invited by someone who is already in the channel.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.**Since:** PircBot 0.9.5

### onRemoveInviteOnly

protected void **onRemoveInviteOnly**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)

Called when a channel has 'invite only' removed.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.**Since:** PircBot 0.9.5

### onSetModerated

protected void **onSetModerated**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)

Called when a channel is set to 'moderated' mode. If a channel is moderated, then only users who have been 'voiced' or 'opped' may speak or change their nicks.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.**Since:** PircBot 0.9.5

### onRemoveModerated

protected void **onRemoveModerated**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)

Called when a channel has moderated mode removed.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.**Since:** PircBot 0.9.5

### onSetPrivate

protected void **onSetPrivate**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)

Called when a channel is marked as being in private mode.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.**Since:** PircBot 0.9.5

### onRemovePrivate

protected void **onRemovePrivate**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)

Called when a channel is marked as not being in private mode.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.**Since:** PircBot 0.9.5

### onSetSecret

protected void **onSetSecret**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)

Called when a channel is set to be in 'secret' mode. Such channels typically do not appear on a server's channel listing.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.**Since:** PircBot 0.9.5

### onRemoveSecret

protected void **onRemoveSecret**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname)

Called when a channel has 'secret' mode removed.

This is a type of mode change and is also passed to the onMode method in the PircBot class.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**channel - The channel in which the mode change took place.sourceNick - The nick of the user that performed the mode change.sourceLogin - The login of the user that performed the mode change.sourceHostname - The hostname of the user that performed the mode change.**Since:** PircBot 0.9.5

### onInvite

protected void **onInvite**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) targetNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel)

Called when we are invited to a channel by a user.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**targetNick - The nick of the user being invited - should be us!sourceNick - The nick of the user that sent the invitation.sourceLogin - The login of the user that sent the invitation.sourceHostname - The hostname of the user that sent the invitation.channel - The channel that we're being invited to.**Since:** PircBot 0.9.5

### onDccSendRequest

protected void **onDccSendRequest**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) filename,  
 long address,  
 int port,  
 int size)

**Deprecated.** *As of PircBot 1.2.0, use* [*onIncomingFileTransfer(DccFileTransfer)*](http://docs.google.com/org/jibble/pircbot/PircBot.html#onIncomingFileTransfer(org.jibble.pircbot.DccFileTransfer))

This method used to be called when a DCC SEND request was sent to the PircBot. Please use the onIncomingFileTransfer method to receive files, as it has better functionality and supports resuming.

### onDccChatRequest

protected void **onDccChatRequest**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 long address,  
 int port)

**Deprecated.** *As of PircBot 1.2.0, use* [*onIncomingChatRequest(DccChat)*](http://docs.google.com/org/jibble/pircbot/PircBot.html#onIncomingChatRequest(org.jibble.pircbot.DccChat))

This method used to be called when a DCC CHAT request was sent to the PircBot. Please use the onIncomingChatRequest method to accept chats, as it has better functionality.

### onIncomingFileTransfer

protected void **onIncomingFileTransfer**([DccFileTransfer](http://docs.google.com/org/jibble/pircbot/DccFileTransfer.html) transfer)

This method is called whenever a DCC SEND request is sent to the PircBot. This means that a client has requested to send a file to us. This abstract implementation performs no action, which means that all DCC SEND requests will be ignored by default. If you wish to receive the file, then you may override this method and call the receive method on the DccFileTransfer object, which connects to the sender and downloads the file.

Example:

public void onIncomingFileTransfer(DccFileTransfer transfer) {  
 // Use the suggested file name.  
 File file = transfer.getFile();  
 // Receive the transfer and save it to the file, allowing resuming.  
 transfer.receive(file, true);  
 }

**Warning:** Receiving an incoming file transfer will cause a file to be written to disk. Please ensure that you make adequate security checks so that this file does not overwrite anything important!

Each time a file is received, it happens within a new Thread in order to allow multiple files to be downloaded by the PircBot at the same time.

If you allow resuming and the file already partly exists, it will be appended to instead of overwritten. If resuming is not enabled, the file will be overwritten if it already exists.

You can throttle the speed of the transfer by calling the setPacketDelay method on the DccFileTransfer object, either before you receive the file or at any moment during the transfer.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**transfer - The DcccFileTransfer that you may accept.**Since:** PircBot 1.2.0 **See Also:**[DccFileTransfer](http://docs.google.com/org/jibble/pircbot/DccFileTransfer.html)

### onFileTransferFinished

protected void **onFileTransferFinished**([DccFileTransfer](http://docs.google.com/org/jibble/pircbot/DccFileTransfer.html) transfer,  
 [Exception](http://java.sun.com/j2se/1.4/docs/api/java/lang/Exception.html) e)

This method gets called when a DccFileTransfer has finished. If there was a problem, the Exception will say what went wrong. If the file was sent successfully, the Exception will be null.

Both incoming and outgoing file transfers are passed to this method. You can determine the type by calling the isIncoming or isOutgoing methods on the DccFileTransfer object.

**Parameters:**transfer - The DccFileTransfer that has finished.e - null if the file was transfered successfully, otherwise this will report what went wrong.**Since:** PircBot 1.2.0 **See Also:**[DccFileTransfer](http://docs.google.com/org/jibble/pircbot/DccFileTransfer.html)

### onIncomingChatRequest

protected void **onIncomingChatRequest**([DccChat](http://docs.google.com/org/jibble/pircbot/DccChat.html) chat)

This method will be called whenever a DCC Chat request is received. This means that a client has requested to chat to us directly rather than via the IRC server. This is useful for sending many lines of text to and from the bot without having to worry about flooding the server or any operators of the server being able to "spy" on what is being said. This abstract implementation performs no action, which means that all DCC CHAT requests will be ignored by default.

If you wish to accept the connection, then you may override this method and call the accept() method on the DccChat object, which connects to the sender of the chat request and allows lines to be sent to and from the bot.

Your bot must be able to connect directly to the user that sent the request.

Example:

public void onIncomingChatRequest(DccChat chat) {  
 try {  
 // Accept all chat, whoever it's from.  
 chat.accept();  
 chat.sendLine("Hello");  
 String response = chat.readLine();  
 chat.close();  
 }  
 catch (IOException e) {}  
 }

Each time this method is called, it is called from within a new Thread so that multiple DCC CHAT sessions can run concurrently.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**chat - A DccChat object that represents the incoming chat request.**Since:** PircBot 1.2.0 **See Also:**[DccChat](http://docs.google.com/org/jibble/pircbot/DccChat.html)

### onVersion

protected void **onVersion**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target)

This method is called whenever we receive a VERSION request. This abstract implementation responds with the PircBot's \_version string, so if you override this method, be sure to either mimic its functionality or to call super.onVersion(...);

**Parameters:**sourceNick - The nick of the user that sent the VERSION request.sourceLogin - The login of the user that sent the VERSION request.sourceHostname - The hostname of the user that sent the VERSION request.target - The target of the VERSION request, be it our nick or a channel name.

### onPing

protected void **onPing**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) pingValue)

This method is called whenever we receive a PING request from another user.

This abstract implementation responds correctly, so if you override this method, be sure to either mimic its functionality or to call super.onPing(...);

**Parameters:**sourceNick - The nick of the user that sent the PING request.sourceLogin - The login of the user that sent the PING request.sourceHostname - The hostname of the user that sent the PING request.target - The target of the PING request, be it our nick or a channel name.pingValue - The value that was supplied as an argument to the PING command.

### onServerPing

protected void **onServerPing**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) response)

The actions to perform when a PING request comes from the server.

This sends back a correct response, so if you override this method, be sure to either mimic its functionality or to call super.onServerPing(response);

**Parameters:**response - The response that should be given back in your PONG.

### onTime

protected void **onTime**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target)

This method is called whenever we receive a TIME request.

This abstract implementation responds correctly, so if you override this method, be sure to either mimic its functionality or to call super.onTime(...);

**Parameters:**sourceNick - The nick of the user that sent the TIME request.sourceLogin - The login of the user that sent the TIME request.sourceHostname - The hostname of the user that sent the TIME request.target - The target of the TIME request, be it our nick or a channel name.

### onFinger

protected void **onFinger**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceNick,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceLogin,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) sourceHostname,  
 [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) target)

This method is called whenever we receive a FINGER request.

This abstract implementation responds correctly, so if you override this method, be sure to either mimic its functionality or to call super.onFinger(...);

**Parameters:**sourceNick - The nick of the user that sent the FINGER request.sourceLogin - The login of the user that sent the FINGER request.sourceHostname - The hostname of the user that sent the FINGER request.target - The target of the FINGER request, be it our nick or a channel name.

### onUnknown

protected void **onUnknown**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) line)

This method is called whenever we receive a line from the server that the PircBot has not been programmed to recognise.

The implementation of this method in the PircBot abstract class performs no actions and may be overridden as required.

**Parameters:**line - The raw line that was received from the server.

### setVerbose

public final void **setVerbose**(boolean verbose)

Sets the verbose mode. If verbose mode is set to true, then log entries will be printed to the standard output. The default value is false and will result in no output. For general development, we strongly recommend setting the verbose mode to true.

**Parameters:**verbose - true if verbose mode is to be used. Default is false.

### setName

protected final void **setName**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) name)

Sets the name of the bot, which will be used as its nick when it tries to join an IRC server. This should be set before joining any servers, otherwise the default nick will be used. You would typically call this method from the constructor of the class that extends PircBot.

The changeNick method should be used if you wish to change your nick when you are connected to a server.

**Parameters:**name - The new name of the Bot.

### setLogin

protected final void **setLogin**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) login)

Sets the internal login of the Bot. This should be set before joining any servers.

**Parameters:**login - The new login of the Bot.

### setVersion

protected final void **setVersion**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) version)

Sets the internal version of the Bot. This should be set before joining any servers.

**Parameters:**version - The new version of the Bot.

### setFinger

protected final void **setFinger**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) finger)

Sets the interal finger message. This should be set before joining any servers.

**Parameters:**finger - The new finger message for the Bot.

### getName

public final [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) **getName**()

Gets the name of the PircBot. This is the name that will be used as as a nick when we try to join servers.

**Returns:**The name of the PircBot.

### getNick

public [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) **getNick**()

Returns the current nick of the bot. Note that if you have just changed your nick, this method will still return the old nick until confirmation of the nick change is received from the server.

The nick returned by this method is maintained only by the PircBot class and is guaranteed to be correct in the context of the IRC server.

**Returns:**The current nick of the bot.**Since:** PircBot 1.0.0

### getLogin

public final [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) **getLogin**()

Gets the internal login of the PircBot.

**Returns:**The login of the PircBot.

### getVersion

public final [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) **getVersion**()

Gets the internal version of the PircBot.

**Returns:**The version of the PircBot.

### getFinger

public final [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) **getFinger**()

Gets the internal finger message of the PircBot.

**Returns:**The finger message of the PircBot.

### isConnected

public final boolean **isConnected**()

Returns whether or not the PircBot is currently connected to a server. The result of this method should only act as a rough guide, as the result may not be valid by the time you act upon it.

**Returns:**True if and only if the PircBot is currently connected to a server.

### setMessageDelay

public final void **setMessageDelay**(long delay)

Sets the number of milliseconds to delay between consecutive messages when there are multiple messages waiting in the outgoing message queue. This has a default value of 1000ms. It is a good idea to stick to this default value, as it will prevent your bot from spamming servers and facing the subsequent wrath! However, if you do need to change this delay value (**not recommended**), then this is the method to use.

**Parameters:**delay - The number of milliseconds between each outgoing message.

### getMessageDelay

public final long **getMessageDelay**()

Returns the number of milliseconds that will be used to separate consecutive messages to the server from the outgoing message queue.

**Returns:**Number of milliseconds.

### getMaxLineLength

public final int **getMaxLineLength**()

Gets the maximum length of any line that is sent via the IRC protocol. The IRC RFC specifies that line lengths, including the trailing \r\n must not exceed 512 bytes. Hence, there is currently no option to change this value in PircBot. All lines greater than this length will be truncated before being sent to the IRC server.

**Returns:**The maximum line length (currently fixed at 512)

### getOutgoingQueueSize

public final int **getOutgoingQueueSize**()

Gets the number of lines currently waiting in the outgoing message Queue. If this returns 0, then the Queue is empty and any new message is likely to be sent to the IRC server immediately.

**Returns:**The number of lines in the outgoing message Queue.**Since:** PircBot 0.9.9

### getServer

public final [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) **getServer**()

Returns the name of the last IRC server the PircBot tried to connect to. This does not imply that the connection attempt to the server was successful (we suggest you look at the onConnect method). A value of null is returned if the PircBot has never tried to connect to a server.

**Returns:**The name of the last machine we tried to connect to. Returns null if no connection attempts have ever been made.

### getPort

public final int **getPort**()

Returns the port number of the last IRC server that the PircBot tried to connect to. This does not imply that the connection attempt to the server was successful (we suggest you look at the onConnect method). A value of -1 is returned if the PircBot has never tried to connect to a server.

**Returns:**The port number of the last IRC server we connected to. Returns -1 if no connection attempts have ever been made.**Since:** PircBot 0.9.9

### getPassword

public final [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) **getPassword**()

Returns the last password that we used when connecting to an IRC server. This does not imply that the connection attempt to the server was successful (we suggest you look at the onConnect method). A value of null is returned if the PircBot has never tried to connect to a server using a password.

**Returns:**The last password that we used when connecting to an IRC server. Returns null if we have not previously connected using a password.**Since:** PircBot 0.9.9

### longToIp

public int[] **longToIp**(long address)

A convenient method that accepts an IP address represented as a long and returns an integer array of size 4 representing the same IP address.

**Parameters:**address - the long value representing the IP address. **Returns:**An int[] of size 4.**Since:** PircBot 0.9.4

### ipToLong

public long **ipToLong**(byte[] address)

A convenient method that accepts an IP address represented by a byte[] of size 4 and returns this as a long representation of the same IP address.

**Parameters:**address - the byte[] of size 4 representing the IP address. **Returns:**a long representation of the IP address.**Since:** PircBot 0.9.4

### setEncoding

public void **setEncoding**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) charset)  
 throws [UnsupportedEncodingException](http://java.sun.com/j2se/1.4/docs/api/java/io/UnsupportedEncodingException.html)

Sets the encoding charset to be used when sending or receiving lines from the IRC server. If set to null, then the platform's default charset is used. You should only use this method if you are trying to send text to an IRC server in a different charset, e.g. "GB2312" for Chinese encoding. If a PircBot is currently connected to a server, then it must reconnect before this change takes effect.

**Parameters:**charset - The new encoding charset to be used by PircBot. **Throws:** [UnsupportedEncodingException](http://java.sun.com/j2se/1.4/docs/api/java/io/UnsupportedEncodingException.html) - If the named charset is not supported.**Since:** PircBot 1.0.4

### getEncoding

public [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) **getEncoding**()

Returns the encoding used to send and receive lines from the IRC server, or null if not set. Use the setEncoding method to change the encoding charset.

**Returns:**The encoding used to send outgoing messages, or null if not set.**Since:** PircBot 1.0.4

### getInetAddress

public [InetAddress](http://java.sun.com/j2se/1.4/docs/api/java/net/InetAddress.html) **getInetAddress**()

Returns the InetAddress used by the PircBot. This can be used to find the I.P. address from which the PircBot is connected to a server.

**Returns:**The current local InetAddress, or null if never connected.**Since:** PircBot 1.4.4

### setDccInetAddress

public void **setDccInetAddress**([InetAddress](http://java.sun.com/j2se/1.4/docs/api/java/net/InetAddress.html) dccInetAddress)

Sets the InetAddress to be used when sending DCC chat or file transfers. This can be very useful when you are running a bot on a machine which is behind a firewall and you need to tell receiving clients to connect to a NAT/router, which then forwards the connection.

**Parameters:**dccInetAddress - The new InetAddress, or null to use the default.**Since:** PircBot 1.4.4

### getDccInetAddress

public [InetAddress](http://java.sun.com/j2se/1.4/docs/api/java/net/InetAddress.html) **getDccInetAddress**()

Returns the InetAddress used when sending DCC chat or file transfers. If this is null, the default InetAddress will be used.

**Returns:**The current DCC InetAddress, or null if left as default.**Since:** PircBot 1.4.4

### getDccPorts

public int[] **getDccPorts**()

Returns the set of port numbers to be used when sending a DCC chat or file transfer. This is useful when you are behind a firewall and need to set up port forwarding. The array of port numbers is traversed in sequence until a free port is found to listen on. A DCC tranfer will fail if all ports are already in use. If set to null, *any* free port number will be used.

**Returns:**An array of port numbers that PircBot can use to send DCC transfers, or null if any port is allowed.**Since:** PircBot 1.4.4

### setDccPorts

public void **setDccPorts**(int[] ports)

Sets the choice of port numbers that can be used when sending a DCC chat or file transfer. This is useful when you are behind a firewall and need to set up port forwarding. The array of port numbers is traversed in sequence until a free port is found to listen on. A DCC tranfer will fail if all ports are already in use. If set to null, *any* free port number will be used.

**Parameters:**ports - The set of port numbers that PircBot may use for DCC transfers, or null to let it use any free port (default).**Since:** PircBot 1.4.4

### equals

public boolean **equals**([Object](http://java.sun.com/j2se/1.4/docs/api/java/lang/Object.html) o)

Returns true if and only if the object being compared is the exact same instance as this PircBot. This may be useful if you are writing a multiple server IRC bot that uses more than one instance of PircBot.

**Returns:**true if and only if Object o is a PircBot and equal to this.**Since:** PircBot 0.9.9

### hashCode

public int **hashCode**()

Returns the hashCode of this PircBot. This method can be called by hashed collection classes and is useful for managing multiple instances of PircBots in such collections.

**Returns:**the hash code for this instance of PircBot.**Since:** PircBot 0.9.9

### toString

public [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) **toString**()

Returns a String representation of this object. You may find this useful for debugging purposes, particularly if you are using more than one PircBot instance to achieve multiple server connectivity. The format of this String may change between different versions of PircBot but is currently something of the form Version{PircBot x.y.z Java IRC Bot - www.jibble.org} Connected{true} Server{irc.dal.net} Port{6667} Password{}

**Returns:**a String representation of this object.**Since:** PircBot 0.9.10

### getUsers

public final [User](http://docs.google.com/org/jibble/pircbot/User.html)[] **getUsers**([String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html) channel)

Returns an array of all users in the specified channel.

There are some important things to note about this method:-

* This method may not return a full list of users if you call it before the complete nick list has arrived from the IRC server.
* If you wish to find out which users are in a channel as soon as you join it, then you should override the onUserList method instead of calling this method, as the onUserList method is only called as soon as the full user list has been received.
* This method will return immediately, as it does not require any interaction with the IRC server.
* The bot must be in a channel to be able to know which users are in it.

**Parameters:**channel - The name of the channel to list. **Returns:**An array of User objects. This array is empty if we are not in the channel.**Since:** PircBot 1.0.0 **See Also:**[onUserList](http://docs.google.com/org/jibble/pircbot/PircBot.html#onUserList(java.lang.String,%20org.jibble.pircbot.User[]))

### getChannels

public final [String](http://java.sun.com/j2se/1.4/docs/api/java/lang/String.html)[] **getChannels**()

Returns an array of all channels that we are in. Note that if you call this method immediately after joining a new channel, the new channel may not appear in this array as it is not possible to tell if the join was successful until a response is received from the IRC server.

**Returns:**A String array containing the names of all channels that we are in.**Since:** PircBot 1.0.0

### dispose

public void **dispose**()

Disposes of all thread resources used by this PircBot. This may be useful when writing bots or clients that use multiple servers (and therefore multiple PircBot instances) or when integrating a PircBot with an existing program.

Each PircBot runs its own threads for dispatching messages from its outgoing message queue and receiving messages from the server. Calling dispose() ensures that these threads are stopped, thus freeing up system resources and allowing the PircBot object to be garbage collected if there are no other references to it.

Once a PircBot object has been disposed, it should not be used again. Attempting to use a PircBot that has been disposed may result in unpredictable behaviour.

**Since:** 1.2.2

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PircBot.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-all.html) | [**Help**](http://docs.google.com/help-doc.html) | |  |  | ***PircBot Java IRC Bot*** |
| [**PREV CLASS**](http://docs.google.com/org/jibble/pircbot/OutputThread.html)   [**NEXT CLASS**](http://docs.google.com/org/jibble/pircbot/Queue.html) | [**FRAMES**](http://docs.google.com/index.html)    [**NO FRAMES**](http://docs.google.com/PircBot.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#id.3znysh7)| [CONSTR](#id.tyjcwt)| [METHOD](#id.3dy6vkm) | DETAIL: [FIELD](#id.4d34og8)| [CONSTR](#id.17dp8vu)| [METHOD](#id.26in1rg) |