Configuring QuickFIX/J

A QuickFIX/J acceptor or initiator can maintain as many FIX sessions as you would like. A FIX session is identified by a group of settings defined within the configuration section for a session (or inherited from the default section). The identification settings are:

Setting	Required?
BeginString	Y
SenderCompID	Y
SenderSubID	N
SenderLocationID	N
TargetCompID	Y
TargetSubID	N
TargetLocationID	N

The sender settings are your identification and the target settings are for the counterparty. A **SessionQualifier** can also be use to disambiguate otherwise identical sessions. *Session qualifier usage is not recommended. It is provided for compatibility with QuickFIX JNI and for the nonstandard FIX implementations where there are multiple sessions that would otherwise have the same identification without the qualifier.* A **SessionQualifier** can only be used with an initiator.

Each of the sessions can have several settings associated with them. Some of these settings may not be known at compile time and are therefore passed around in a class called SessionSettings.

The SessionSettings class has the ability to pull settings out of any input stream such as a file stream. You can also simply pass it a filename. If you decide to write your own components, (storage for a particular database, a new kind of connector etc...), you may also use the session settings to store settings for your custom component.

A settings file is set up with two types of heading, a [DEFAULT] and a [SESSION] heading. [SESSION] tells QuickFIX/J that a new Session is being defined. [DEFAULT] is a place that you can define settings which will be inherited by sessions that don't explicitly define them. If you do not provide a setting that QuickFIX/J needs, it will throw a ConfigError telling you what setting is missing or improperly formatted.

These are the settings you can associate with a session based on the default components provided with QuickFIX, followed by an example.

QuickFIX Settings

ID	Description	Valid Values	Default
Session			
BeginString	Version of FIX this session should use	FIX.4.4 FIX.4.3 FIX.4.2 FIX.4.1 FIX.4.0 FIXT.1.1 (which then requires DefaultApplVerID, see below)	
SenderCompID	Your compID as associated with this FIX session	case-sensitive alpha-numeric string	
SenderSubID	(Optional) Your subID as associated with this FIX session	case-sensitive alpha-numeric string	
SenderLocationID	(Optional) Your locationID as associated with this FIX session	case-sensitive alpha-numeric string	
TargetCompID	Counterparty's compID as associated with this FIX session	case-sensitive alpha-numeric string	
TargetSubID	(Optional) Counterparty's subID as associated with this FIX session	case-sensitive alpha-numeric string	
TargetLocationID	(Optional) Counterparty's locationID as associated with this FIX session	case-sensitive alpha-numeric string	
SessionQualifier	Additional qualifier to disambiguate otherwise identical sessions. This can only be used with initiator sessions. **Note:** See Special notes for Oracle.	case-sensitive alpha-numeric string	

DefaultApplVerID	Required only for FIXT 1.1 (and newer). Ignored for earlier transport versions. Specifies the default application version ID for the session. This can either be the ApplVerID enum (see the ApplVerID field) the beginString for the default version.	String. Examples:
		`# Enum. FIX 5.0 over FIXT 1.1`
		`DefaultApplVerID=7`
		`# BeginString: FIX 5.0 over FIXT 1.1`
		`DefaultApplVerID=FIX.5.0`
		`# BeginString: FIX 4.2 over FIXT 1.1`
		`DefaultApplVerID=FIX.4.2`
			No default. Required for FIXT 1.1

ConnectionType	Defines if session will act as an acceptor or an initiator	initiator acceptor	
TimeZone	Time zone for this session; if specified, the session start and end will be converted from this zone to UTC.	Time zone ID (America/New_York, Asia/Tokyo, Europe/London, etc.)	
StartTime	Time of day that this FIX session becomes activated	time in the format of HH:MM:SS [timezone]. The time zone is optional. The TimeZone setting will be used, if set, or UTC will be used by default. The timezone string should be one that the Java TimeZone class can resolve. For example, "15:00:00 US/Central".	

EndTime	Time of day that this FIX session becomes deactivated	time in the format of HH:MM:SS [timezone]. The time zone is optional. The TimeZone setting will be used, if set, or UTC will be used by default. The timezone string should be one that the Java TimeZone class can resolve. For example, "09:00:00 US/Eastern".	
StartDay	For week long sessions, the starting day of week for the session. Use in combination with StartTime. Incompatible with Weekdays	Day of week in the default locale (e.g. Monday, mon, lundi, lun. etc.)	
EndDay	For week long sessions, the ending day of week for the session. Use in combination with EndTime. Incompatible with Weekdays	Day of week in the default locale (e.g. Monday, mon, lundi, lun. etc.)	
Weekdays	For daily sessions that are active on specific days of the week. Use in combination with StartTime and EndTime. Incompatible with StartDay and EndDay. If StartTime is before EndTime then the day corresponds to the StartTime.	Comma-delimited list of days of the week in the default locale (e.g. "Sun,Mon,Tue", "Dimanche,Lundi,Mardi" etc.)	
NonStopSession	If set the session will *never* reset. This is effectively the same as setting 00:00:00 as StartTime and EndTime.	Y N	N
TimeStampPrecision	Determines precision for timestamps in (Orig)SendingTime fields. Only available for FIX.4.2 and greater. NB: This configuration is only considered for messages that are sent out. QuickFIX/J is able to receive UtcTimestamp fields with up to picosecond precision. Please note however that only up to nanosecond precision will be stored, i.e. the picoseconds will be truncated.	One of • SECONDS • MILLIS • MICROS • NANOS	MILLIS
ClosedResendInterval	Use actual end of sequence gap for resend requests rather than using "infinity" as the end sequence of the gap. Not recommended by the FIX specification, but needed for some counterparties.	Y N	N
Validation			
UseDataDictionary	Tell session whether or not to expect a data dictionary. You should always use a DataDictionary if you are using repeating groups.	Y N	Y
DataDictionary	XML definition file for validating incoming FIX messages. If no DataDictionary is supplied, only basic message validation will be done This setting should only be used with FIX transport versions old than FIXT 1.1. See TransportDataDictionary and ApplicationDataDictionary for FIXT 1.1 settings.	Valid XML data dictionary file, QuickFIX/J comes with the following defaults in the `etc` directory: FIXT11.xml, FIX50.xml, FIX44.xml, FIX43.xml, FIX42.xml, FIX41.xml, FIX40.xml.	If DataDictionary is not specified and UserDataDictionary=Y, then QuickFIX/J will look for a default dictionary based on the session's BeginString (e.g., FIX.4.2 = FIX42.xml). The DataDictionary file search strategy is to use a URL, then the file system, and then the thread context classloader (if any), and then the DataDictionary instance's classloader. Default data dictionary files are included in the QuickFIX/J jar file.
TransportDataDictionary	XML definition file for validating admin (transport) messages. This setting is only valid for the FIXT 1.1 (or newer) sessions. See DataDictionary for older transport versions (FIX 4.0-4.4) and for additional information.	Valid XML data dictionary file path.	If no dictionary path is supplied, an attempt will be made to load a default transport dictionary.

AppDataDictionary	XML definition file for validating application messages. This setting is only valid for the FIXT 1.1 (or newer) sessions.
	
	See DataDictionary for older transport versions (FIX 4.0-4.4) and for additional information.
	
	This setting supports the possibility of a custom application data dictionary for each session. This setting would only be used with FIXT 1.1 and new transport protocols. This setting can be used as a prefix to specify multiple application dictionaries for the FIXT transport. For example:
	
	`DefaultApplVerID=FIX.4.2`
	`# For default application version ID`
	`AppDataDictionary=FIX42.xml`
	`# For nondefault application version ID`
	`# Use beginString suffix for app version`
	`AppDataDictionary.FIX.4.4=FIX44.xml`
	
	This would use FIX42.xml for the default application version ID and FIX44.xml for any FIX 4.4 messages.
		Valid XML data dictionary file path.	If no dictionary path is supplied, an attempt will be made to load a dictionary using the DefaultApplVerID for the session.

ValidateFieldsOutOfOrder	If set to N, fields that are out of order (i.e. body fields in the header, or header fields in the body) will not be rejected. Useful for connecting to systems which do not properly order fields.	Y N	Y
ValidateFieldsHaveValues	If set to N, fields without values (empty) will not be rejected. Useful for connecting to systems which improperly send empty tags.	Y N	Y
ValidateUserDefinedFields	If set to N, user defined fields (field with tag >= 5000) will not be rejected if they are not defined in the data dictionary, or are present in messages they do not belong to.	Y N	Y
ValidateUnorderedGroupFields	Session validation setting for enabling whether field ordering is * validated. Values are "Y" or "N". Default is "Y".	Y N	Y
ValidateIncomingMessage	Allow to bypass the message validation (against the dictionary). Default is "Y".	Y N	Y
ValidateSequenceNumbers	Check the next expected target SeqNum against the received SeqNum. Default is "Y". If enabled and a mismatch is detected, apply the following logic: • if lower than expected SeqNum , logout • if higher, send a resend request If not enabled and a mismatch is detected, nothing is done. Must be enabled for EnableNextExpectedMsgSeqNum to work.	Y N	Y
ValidateChecksum	If ValidateChecksum is set to N, checksum validation will not be executed on messages. This setting cannot be set to N together with RejectGarbledMessage set to Y, in this case Config Error will be thrown.	Y N	Y
AllowUnknownMsgFields	If set to Y, non user defined fields (field with tag < 5000) will not be rejected if they are not defined in the data dictionary, or are present in messages they do not belong to.	Y N	N

CheckCompID	If set to Y, messages must be received from the counterparty with the correct SenderCompID and TargetCompID. Some systems will send you different CompIDs by design, so you must set this to N.	Y N	Y
CheckLatency	If set to Y, messages must be received from the counterparty within a defined number of seconds (see MaxLatency). It is useful to turn this off if a system uses localtime for its timestamps instead of GMT.	Y N	Y
MaxLatency	If CheckLatency is set to Y, this defines the number of seconds latency allowed for a message to be processed.	positive integer	120
RejectGarbledMessage	If RejectGarbledMessage is set to Y, garbled messages will be rejected (with a generic error message in 58/Text field) instead of ignored. This is only working for messages that pass the FIX decoder and reach the engine. Messages that cannot be considered a real FIX message (i.e. not starting with 8=FIX or not ending with 10=xxx) will be ignored in any case. See Invalid vs Garbled Messages for further explanation.	Y N	N
RejectInvalidMessage	If RejectInvalidMessage is set to N, only a warning will be logged on reception of message that fails data dictionary validation. See Invalid vs Garbled Messages for further explanation.	Y N	Y
RejectMessageOnUnhandledException	If this configuration is enabled, an uncaught Exception or Error in the application's message processing will lead to a (BusinessMessage)Reject being sent to the counterparty and the incoming message sequence number will be incremented. If disabled (default), the problematic incoming message is discarded and the message sequence number is not incremented. Processing of the next valid message will cause detection of a sequence gap and a ResendRequest will be generated.	Y N	N
RequiresOrigSendingTime	If RequiresOrigSendingTime is set to N, PossDup messages lacking that field will not be rejected.	Y N	Y
Initiator			
ReconnectInterval	Time between reconnection attempts in seconds. Only used for initiators	positive integer	30
HeartBtInt	Heartbeat interval in seconds. Only used for initiators.	positive integer	
LogonTimeout	Number of seconds to wait for a logon response before disconnecting.	positive integer	10
LogoutTimeout	Number of seconds to wait for a logout response before disconnecting.	positive integer	2
SocketConnectPort	Socket port for connecting to a session. Only used with a SocketInitiator	positive integer	
SocketConnectHost	Host to connect to. Only used with a SocketInitiator	valid IP address in the format of x.x.x.x or a domain name	
SocketConnectProtocol	Specifies the initiator communication protocol. The SocketConnectHost is not used with the VM_PIPE protocol, but the SocketConnectPort is significant and must match the acceptor configuration.	"TCP" or "VM_PIPE".	"TCP"
SocketConnectPort<n>	Alternate socket port(s) for connecting to a session for failover or load balancing, where **n** is a positive integer, i.e. SocketConnectPort1, SocketConnectPort2, etc. Must be consecutive and have a matching SocketConnectHost<n>	positive integer	
SocketConnectHost<n>	Alternate socket host(s) for connecting to a session for failover or load balancing, where **n** is a positive integer, i.e. SocketConnectHost1, SocketConnectHost2, etc. Must be consecutive and have a matching SocketConnectPort<n> Connection list iteration rules: • Connections are tried one after another until one is successful: SocketConnectHost:SocketConnectPort, SocketConnectHost1:SocketConnectPort1, etc. • Next connection attempt after a successful connection will start at first defined connection again: SocketConnectHost:SocketConnectPort.	valid IP address in the format of x.x.x.x or a domain name	
SocketLocalPort	Bind the local socket to this port. Only used with a SocketInitiator.	positive integer	If unset the socket will be bound to a free port from the ephemeral port range.
SocketLocalHost	Bind the local socket to this host. Only used with a SocketInitiator.	valid IP address in the format of x.x.x.x or a domain name	If unset the socket will be bound to all local interfaces.
DynamicSession	Leave the corresponding session disconnected until AbstractSocketInitiator.createDynamicSession is called	Y N	N
Acceptor			
SocketAcceptPort	Socket port for listening to incoming connections. Only used with a SocketAcceptor	positive integer, valid open socket port.	
SocketAcceptAddress	Local IP address to for binding accept port.	A hostname or IP address parsable by `java.net.InetAddress`.	Accept connections on any network interface.
SocketAcceptProtocol	Specifies the acceptor communication protocol. The SocketAcceptAddress is not used with the VM_PIPE protocol, but the SocketAcceptPort is significant and must match the initiator configuration.	"TCP" or "VM_PIPE".	"TCP"
AllowedRemoteAddresses	List of remote IP addresses which are allowed to connect to this acceptor.	comma-separated list of hostnames or IP addresses parseable by `java.net.InetAddress`	empty, ie all remote addresses are allowed
AcceptorTemplate	Designates a template Acceptor session. See Dynamic Acceptor Sessions	Y N	N
Secure Communication Options			
SocketUseSSL	Enables SSL usage for QFJ acceptor or initiator.	Y N	N
SocketKeyStore	KeyStore to use with SSL	File path	
SocketKeyStorePassword	KeyStore password		
KeyManagerFactoryAlgorithm	Algorithm used when generating an instance of KeyManagerFactory		SunX509
KeyStoreType	KeyStore type		JKS
SocketTrustStore	TrustStore to use with SSL	File path	
SocketTrustStorePassword	TrustStore password		
TrustManagerFactoryAlgorithm	Algorithm used when generating an instance of TrustManagerFactory		PKIX
TrustStoreType	TrustStore type		JKS
NeedClientAuth	Configures the SSL engine to require client authentication. This option is only useful to acceptors.	Y N	N
EnabledProtocols	Protocols enabled for use with the SSL engine.		Java supported protocols
CipherSuites	Cipher suites enabled for use with the SSL engine.		Java default cipher suites

UseSNI	Configures the SSL engine to use Server Name Indication. This option is only useful to initiators.	Y N	N
Socks Proxy Options (Initiator only)			
ProxyType	Proxy type	http socks	
ProxyVersion	Proxy HTTP or Socks version to use	For socks: 4, 4a or 5 For http: 1.0 or 1.1	For socks: For http: 1.0
ProxyHost	Proxy server hostname or IP	valid IP address in the format of x.x.x.x or a domain name	
ProxyPort	Proxy server port	positive integer	
ProxyUser	Proxy user		
ProxyPassword	Proxy password		
ProxyDomain	Proxy domain (For http proxy)		
ProxyWorkstation	Proxy workstation (For http proxy)		
Socket Options (Acceptor or Initiator)			
Acceptor and Initiator socket options can be set in either defaults or per-session settings.			
SocketKeepAlive	When the keepalive option is set for a TCP socket and no data has been exchanged across the socket in either direction for 2 hours (NOTE: the actual value is implementation dependent), TCP automatically sends a keepalive probe to the peer. This probe is a TCP segment to which the peer must respond. One of three responses is expected: 1. The peer responds with the expected ACK. The application is not notified (since everything is OK). TCP will send another probe following another 2 hours of inactivity. 2. The peer responds with an RST, which tells the local TCP that the peer host has crashed and rebooted. The socket is closed. 3. There is no response from the peer. The socket is closed. The purpose of this option is to detect if the peer host crashes.	Y N	
SocketOobInline	When the OOBINLINE option is set, any TCP urgent data received on the socket will be received through the socket input stream. When the option is disabled (which is the default) urgent data is silently discarded.	Y N	
SocketReceiveBufferSize	Set a hint the size of the underlying buffers used by the platform for incoming network I/O. When used in set, this is a suggestion to the kernel from the application about the size of buffers to use for the data to be received over the socket.	Integer value.	
SocketReuseAddress	Sets SO_REUSEADDR for a socket. This is used only for MulticastSockets in java, and it is set by default for MulticastSockets.	Y N	
SocketSendBufferSize	Set a hint the size of the underlying buffers used by the platform for outgoing network I/O. When used in set, this is a suggestion to the kernel from the application about the size of buffers to use for the data to be sent over the socket.	Integer value.	
SocketLinger	Specify a linger-on-close timeout. This option disables/enables immediate return from a **close()** of a TCP Socket. Enabling this option with a non-zero Integer *timeout* means that a **close()** will block pending the transmission and acknowledgement of all data written to the peer, at which point the socket is closed *gracefully*. Upon reaching the linger timeout, the socket is closed *forcefully*, with a TCP RST. Enabling the option with a timeout of zero does a forceful close immediately. If the specified timeout value exceeds 65,535 it will be reduced to 65,535.	Integer value.	
SocketTcpNoDelay	Disable Nagle's algorithm for this connection. Written data to the network is not buffered pending acknowledgement of previously written data.	Y N	Y
SocketTrafficClass	Sets traffic class or type-of-service octet in the IP header for packets sent from this Socket. As the underlying network implementation may ignore this value applications should consider it a hint. The tc **must** be in the range 0 ≤ tc ≤ 255 or an IllegalArgumentException will be thrown. Notes: for Internet Protocol v4 the value consists of an octet with precedence and TOS fields as detailed in RFC 1349. The TOS field is bitset created by bitwise-or'ing values such the following :- • `IPTOS_LOWCOST (0x02)` • `IPTOS_RELIABILITY (0x04)` • `IPTOS_THROUGHPUT (0x08)` • `IPTOS_LOWDELAY (0x10)` The last low order bit is always ignored as this corresponds to the MBZ (must be zero) bit. Setting bits in the precedence field may result in a SocketException indicating that the operation is not permitted.	An integer value or a set of string options separated by "|" (e.g., "IPTOS_LOWCOST|IPTOS_LOWDELAY")	
SocketSynchronousWrites	Write messages synchronously. This is not generally recommended as it may result in performance degradation. The MINA communication layer is asynchronous by design, but this option will override that behavior if needed.	Y N	N
SocketSynchronousWriteTimeout	The time in milliseconds to wait for a write to complete.	Integer.	30000 ms (30 seconds) if SocketSynchronousWrites is "Y".
MaxScheduledWriteRequests	Number of scheduled write requests on which session is forcefully disconnected.	positive Integer.	0 (disabled)
Storage			
Note: Unlike in QuickFIX JNI, database-specific classes (MySQLStore, etc.) are not included in QuickFIX/J. Use the JDBC support instead. The message store and logging schema are simple and should be easily adapted to any JDBC-supported database.			
PersistMessages	If set to N, no messages will be persisted. This will force QFJ to always send GapFills instead of resending messages. Use this if you know you never want to resend a message. Useful for market data streams.	Y N	Y
FileStorePath	Directory to store sequence number and message files. Only used with FileStoreFactory.	valid directory for storing files, must have write access	
FileStoreMaxCachedMsgs	Maximum number of message index entries to cache in memory.	Integer. A zero will not cache any entries.	10000
FileStoreSync	Whether the FileStore syncs to the hard drive on every write. It's safer to sync, but it's also much slower.	Y N	N

JdbcDataSourceName	JNDI name for the JDBC data source. This technique for finding the data source can be used as an alternative to specifying the driver details. It allows better integration with application servers and servlet containers that are already configured with JDBC data sources.	JNDI name of the data source. Configuration of the initial context must be done by an application server, through a property file or through system properties. See JNDI documentation for more information.	
JdbcDriver	JDBC driver for JDBC logger. Also used for JDBC log.	Class name for the JDBC driver. Specify driver properties directly will cause the creation of a Proxool data source that supports connection pooling. If you are using a database with it's own pooling data source (e.g., Oracle) then use the `setDataSource()` method on the Jdbc-related factories to set the data source directly.	
JdbcURL	JDBC database URL. Also used for JDBC log.	Depends on the JDBC database driver.	
JdbcUser	JDBC user. Also used for JDBC log.		
JdbcPassword	JDBC password. Also used for JDBC log.		
JdbcStoreMessagesTableName	Table name for messages table.	A valid SQL table name.	messages
JdbcStoreSessionsTableName	Table name for sessions table.	A valid SQL table name.	sessions
JdbcLogHeartBeats	Controls filtering of heartbeats for message logging (both in and out).	Y N	N
JdbcLogIncomingTable	The name of the JDBC log incoming table.	valid table name	messages_log
JdbcLogOutgoingTable	The name of the JDBC log outgoing table.	valid table name	messages_log
JdbcLogEventTable	The name of the JDBC log events table.	valid table name	event_log
JdbcSessionIdDefaultPropertyValue	The default value for Session ID bean properties is an empty string. Oracle treats this as a SQL NULL and that causes problems. This configuration setting allows you to set the default value for unspecified Session ID properties.	Any nonempty string.	"" (empty string)
JdbcMaxActiveConnection	Specifies the maximum number of connections to the database.	Any number	32
JdbcMaxActiveTime	Specifies if the housekeeper comes across a thread that has been active for longer than this (milliseconds) then it will kill it. So make sure you set this to a number bigger than your slowest expected response!	Any number	5000
JdbcMaxConnectionLifeTime	Specifies the maximum amount of time that a connection exists for before it is killed (milliseconds).	Any number	28800000
JdbcSimultaneousBuildThrottle	Specifies the maximum number of connections we can be building at any one time. That is, the number of new connections that have been requested but aren't yet available for use. Because connections can be built using more than one thread (for instance, when they are built on demand) and it takes a finite time between deciding to build the connection and it becoming available we need some way of ensuring that a lot of threads don't all decide to build a connection at once. (We could solve this in a smarter way - and indeed we will one day)	Any number	32
Logging			
FileLogPath	Directory to store logs. Only used with FileLogFactory.	valid directory for storing files, must have write access	
FileLogHeartbeats	Controls logging of heartbeat messages.	Y N	N
FileIncludeMilliseconds	Controls whether milliseconds are included in log time stamps.	Y N	N
FileIncludeTimeStampForMessages	Controls whether time stamps are included on message log entries.	Y N	N
SLF4JLogEventCategory	Log category for logged events.	Depends on log engine. The SLF4J adapter for JDK 1.4 logging is included by default. An adapter for Log4J and the Log4J JAR are in the lib/optional directory. See slf4j.org for other options. The SLF4J category options support Session ID variables in the category names. The variables are: • ${fixMajorVersion} • ${fixMinorVersion} • ${senderCompID} • ${targetCompID} • ${qualifier} For example, a category value "${senderCompID}.events" (without the quotes) would become "BANZAI.events" in the log file if BANZAI is the senderCompID for the session. This can be used with advanced logging libraries like Log4J to create sophisticated session-specific logging policies.	quickfixj.event
SLF4JLogIncomingMessageCategory	Log category for incoming messages.	Depends on log engine. See "SL4JLogEventCategory".	quickfixj.msg.incoming
SLF4JLogOutgoingMessageCategory	Log category for outgoing messages.	Depends on log engine. See "SL4JLogEventCategory".	quickfixj.msg.outgoing
SLF4JLogPrependSessionID	Controls whether session ID is prepended to log message.	Y N	Y
SLF4JLogHeartbeats	Controls whether heartbeats are logged.	Y N	N
JdbcDriver	JDBC driver for JDBC logger. Also used for JDBC message store.	Classname for the JDBC driver.	
JdbcURL	JDBC database URL. Also used for JDBC message store.	Depends on the JDBC database driver.	
JdbcUser	JDBC user. Also used for JDBC message store.		
JdbcPassword	JDBC password. Also used for JDBC message store.		
ScreenLogShowEvents	Log events to screen.	Y N	Y
ScreenLogShowIncoming	Log incoming messages to screen.	Y N	Y
ScreenLogShowOutgoing	Log outgoing messages to screen.	Y N	Y
ScreenLogShowHeartBeats	Filter heartbeats from output (both incoming and outgoing)	Y N	N
Miscellaneous			
LogonTag	Tag/value pair which will be set on sent or received Logon message.	<tag>=<value>, where "tag" has to be a positive integer and "value" a String	

		Example: LogonTag=553=foo	
LogonTag<n>	Additional tag/value pairs which will be set on sent or received Logon message, where **n** is a positive integer, i.e. LogonTag1, LogonTag2, etc. Must be consecutive.	<tag>=<value>, where "tag" has to be a positive integer and "value" a String Example: LogonTag=553=user LogonTag1=554=password	
RefreshOnLogon	Refresh the session state when a Logon is received. This allows a simple form of failover when the message store data is persistent. The option will be ignored for message stores that are not persistent (e.g., MemoryStore).	Y N	N
ResetOnLogon	Determines if sequence numbers should be reset before sending/receiving a logon request.	Y N	N
ResetOnLogout	Determines if sequence numbers should be reset to 1 after a normal logout termination.	Y N	N
ResetOnDisconnect	Determines if sequence numbers should be reset to 1 after an abnormal termination.	Y N	N
ResetOnError	Session setting for doing an automatic reset when an error occurs. A reset means disconnect, sequence numbers reset, store cleaned and reconnect, as for a daily reset.	Y N	N
DisconnectOnError	Session setting for doing an automatic disconnect when an error occurs.	Y N	N
EnableLastMsgSeqNumProcessed	Add tag LastMsgSeqNumProcessed in the header (optional tag 369).	Y N	N
EnableNextExpectedMsgSeqNum	Add tag NextExpectedMsgSeqNum (optional tag 789) on the sent Logon message and use value of tag 789 on received Logon message to synchronize session. This should not be enabled for FIX versions < 4.4. Only works when ValidateSequenceNumbers is enabled.	Y N	N
ResendRequestChunkSize	Setting to limit the size of a resend request in case of missing messages. This is useful when the remote FIX engine does not allow to ask for more than n message for a ResendRequest. E.g. if the ResendRequestChunkSize is set to 5 and a gap of 7 messages is detected, a first resend request will be sent for 5 messages. When this gap has been filled, another resend request for 2 messages will be sent. If the ResendRequestChunkSize is set to 0, only one ResendRequest for all the missing messages will be sent.	any positive integer	0 (disables splitting)
ContinueInitializationOnError	Continue initializing sessions if an error occurs. Useful when having multiple sessions per connector and misconfigured session(s) should not prevent the connector from starting.	Y N	N
SendRedundantResendRequests	Allows sending of redundant resend requests.	Y N	N
TestRequestDelayMultiplier	Fraction of the heartbeat interval which defines the additional time to wait if a TestRequest sent after a missing heartbeat times out (final coefficient value is equal to TestRequestDelayMultiplier + 1.0).	any non-negative value	0.5
HeartBeatTimeoutMultiplier	Fraction of the heartbeat interval which defines the additional time to wait since the last message was received before disconnecting (final coefficient value is equal to HeartBeatTimeoutMultiplier + 1.0).	any non-negative value	1.4
DisableHeartBeatCheck	Heartbeat detection is disabled. A disconnect due to a missing heartbeat will never occur.	Y N	N
ForceResendWhenCorruptedStore	Fill in heartbeats on resend when reading from message store fails.	Y N	N

Rejecting Invalid vs Garbled Messages

There are mainly two settings that influence QFJ's rejection behaviour:

- RejectInvalidMessage
- RejectGarbledMessage

While the first applies to messages that fail data dictionary validation, the latter applies to messages that fail basic validity checks on the FIX protocol level.

Setting RejectInvalidMessage

If RejectInvalidMessage is set to

- Y, the problematic message will be rejected (this is the default setting).
- N, only a warning will be logged on reception of a message that fails data dictionary validation. The message will then be handed over to the application level code.

Setting RejectGarbledMessage

If RejectGarbledMessage is set to

- Y, garbled messages will be rejected (with a generic error message in 58/Text field) instead of ignored.
- N, garbled messages will be ignored and the sequence number not be incremented (this is the default setting).

Information on garbled messages

In FIX it is legal to ignore a message under certain circumstances. Since FIX is an optimistic protocol it expects that some errors are transient and will correct themselves with the next message transmission. Therefore the sequence number is not incremented and a resend request is issued on the next received message that has a higher sequence number than expected.

In the case that the error is not transient, the default behaviour is not optimal because not consuming a message sequence number can lead to follow-up problems since QFJ will wait for the message to be resent and queue all subsequent messages until the resend request has been satisfied (i.e. infinite resend loop).

What constitutes a garbled message (taken from the FIX protocol specification):

- BeginString (tag #8) is not the first tag in a message or is not of the format 8=FIXT.n.m.
- BodyLength (tag #9) is not the second tag in a message or does not contain the correct byte count.
- MsgType (tag #35) is not the third tag in a message.
- Checksum (tag #10) is not the last tag or contains an incorrect value.

If the MsgSeqNum(tag #34) is missing a logout message should be sent terminating the FIX Connection, as this indicates a serious application error that is likely only circumvented by software modification.

You have the possibility to adapt QFJ's behaviour for some of the cases mentioned above.

- If an incoming message does neither start with the BeginString tag nor does it end with the Checksum tag, the message cannot be passed to the session and will be discarded by the FIX decoder right away.
- Examples where the message will be rejected instead of ignored when RejectGarbledMessage=Y:
  - incorrect checksum
  - repeating group count field contains no valid integer
  - no SOH delimiter found in field
  - missing MsgType
  - invalid tags, e.g. 49foo=bar

Sample Settings File

Here is a typical settings file you might find in a firm that wants to connect to several ECNs.

```
# default settings for sessions
[DEFAULT]
ConnectionType=initiator
ReconnectInterval=60
SenderCompID=TW

# session definition
[SESSION]
# inherit ConnectionType, ReconnectInterval and SenderCompID from default
BeginString=FIX.4.1
TargetCompID=ARCA
StartTime=12:30:00
EndTime=23:30:00
HeartBtInt=20
SocketConnectPort=9823
SocketConnectHost=123.123.123.123
DataDictionary=somewhere/FIX41.xml

[SESSION]
BeginString=FIX.4.0
TargetCompID=ISLD
StartTime=12:00:00
EndTime=23:00:00
HeartBtInt=30
SocketConnectPort=8323
SocketConnectHost=23.23.23.23
DataDictionary=somewhere/FIX40.xml

[SESSION]
BeginString=FIX.4.2
TargetCompID=INCA
StartTime=12:30:00
EndTime=21:30:00
# overide default setting for RecconnectInterval
ReconnectInterval=30
HeartBtInt=30
SocketConnectPort=6523
SocketConnectHost=3.3.3.3
# (optional) alternate connection ports and hosts to cycle through on failover
SocketConnectPort1=8392
SocketConnectHost1=8.8.8.8
SocketConnectPort2=2932
SocketConnectHost2=12.12.12.12
DataDictionary=somewhere/FIX42.xml
```