

# FIX Engine parameters

- Overview
- Engine parameters
  - EngineRoot
  - ListenPort
  - LogFileName
  - ListenAddress
  - LicenseFile
  - LogDirectory
  - ConnectAddress
  - NumberOfWorkers
  - OutOfSequenceMessagesStrategy
  - OutOfSequenceMessagesStrategy.QueueSize
- Session parameters
  - LogonTimeFrame
  - LogoutTimeFrame
  - IntradayLogoutTolerance
  - ReasonableTransmissionTime
  - ForceSeqNumReset
  - DuplicateResendRequestLimit
  - AdditionalParsersList
  - ThirdPartyRoutingIsEnabled
  - DelayedProcessing.DeliveryTriesInterval
  - DelayedProcessing.MaxDeliveryTries
  - Reconnect.Interval
  - Reconnect.MaxTries
  - MessageTimeToLive
  - CheckVersionOfOutgoingMessages
  - ResendMessagesBlockSize
  - ResetSeqNumAfter24hours
  - DictionariesFilesList
  - EncryptionConfigFile
  - TotalOutgoingStorageMemoryLimit
  - Debug.LogSessionExtraParameters
  - OutOfSequenceMessagesStrategy
  - OutOfSequenceMessagesStrategy.QueueSize
  - SocketOpPriority
  - AggressiveReceiveDelay
  - SSL
  - ParserVersion
  - Role
  - Host
  - Port
  - HBI
  - CustomLogonMessageFileName
- Storages
  - LogIncomingMessages
  - OutgoingMessagesStorageSize
  - ExtraSafeMode
  - TimestampsInLogs
  - EnableIncrementalLogFileCreation
  - Persistents.LogIncrementSize
  - Persistents.IdxIncrementSize
  - SplitPersistentMsgStorage.BackupDir
  - SplitPersistentMsgStorage.MaxSliceSize
  - MessageStorage.IndexRebuildEnabled
  - MessageStorage.RequireRejectFlagSupport
  - MessageStorage.TimestampUnit
- Unregistered Acceptors parameters
  - CreateUnregisteredAcceptorSession
  - UnregisteredAcceptor.CreateSession
  - UnregisteredAcceptor.IgnoreSeqNumTooLowAtLogon
  - UnregisteredAcceptor.maxMessagesAmountInBunch
  - UnregisteredAcceptor.RejectMessageWhileNoConnection
  - UnregisteredAcceptor.tcpBufferDisabled
  - UnregisteredAcceptor.SessionStorageType
- Message Validation parameters
  - MessageMustBeValidated
  - VerifyTagsValues
  - ProhibitUnknownTags
  - VerifyRepeatingGroupBounds
  - IgnoreUnknownFields
  - Validation.CheckRequiredGroupFields
  - AllowEmptyFieldValue
- Logging parameters
  - Log.File.Format

- Log.File.CreateUniqueFileName
- Log.Cycling
- Log.Cycling.Ignore
- Log.Cycling.BlockSize
- Log.Cycling.Multiplier
- Log.Device
- Log.EventLog.EventSource
- Log.File.AutoFlush
- Log.File.Backup.Time
- Log.File.Locked
- Log.File.Recreate
- Log.File.RootDir
- Log.File.Rotate
- Log.File.Rotate.SizeLimit
- Log.File.Rotate.FileLimit
- Log.File.Rotate.Dir
- Log.File.TimeZone
- Log.File.Name
- Log.DebugsOn
- Log.ErrorsOn
- Log.FatalsOn
- Log.NotelsOn
- Log.WarnsOn
- Log.TracelsOn
- Log.OpenLDAP.TracelsOn
- Affinity parameters
  - WorkerCpuAffinity
  - HelperCpuAffinity
  - CpuAffinity
  - RecvCpuAffinity
  - SendCpuAffinity
- Backup parameters
  - BackupDirectory
  - EnableAutoSwitchToBackupConnection
  - EnableCyclicSwitchBackupConnection
- Monitoring parameters
  - Monitoring.AdminSessionDef.DisableTCPBuffer
  - Monitoring.AdminSessionDef.EncryptMethod
  - Monitoring.AdminSessionDef.ForceSeqNumResetMode
  - Monitoring.AdminSessionDef.IgnoreSeqNumTooLowAtLogon
  - Monitoring.AdminSessionDef.IntradayLogoutToleranceMode
  - Monitoring.AdminSessionDef.MaxMessagesAmountInBunch
  - Monitoring.AdminSessionDef.Password
  - Monitoring.AdminSessionDef.SenderLocationID
  - Monitoring.AdminSessionDef.SocketOpPriority
  - Monitoring.AdminSessionDef.SourceIPAddress
  - Monitoring.AdminSessionDef.TargetLocationID
  - Monitoring.AdminSessionDef.TargetSubID
  - Monitoring.AdminSessionDef.Username
  - Monitoring.ListenPort
  - Measuring.Enable
  - Monitoring.Enable
- SSL parameters
  - ListenSSLPort
  - SSLCACertificate
  - SSLCertificate
  - SSLPrivateKey
  - SSLCheckPrivateKey
  - SSLProtocols
  - SSLRequireClientCertificate
  - SSLCertificatePassword
  - SSLPrivateKeyPassword
  - SSLValidatePeerCertificate
  - SSLCiphersList
- TCP Protection parameters
  - ProtectionTCP.Enabled
  - ProtectionTCP.WaitLogon
  - ProtectionTCP.SizeWaitHostMax
  - ProtectionTCP.SizeBufferMax
- Other parameters
  - Myricom.Dbl.TcpListenAddresses
  - Myricom.Dbl.UseSocketAdaptor
  - EnableDnsEntrySpoofingDetection
  - KeepMillisecondsInUTCTimeFields

# Overview

FIX Antenna reads the configuration parameters by default from the engine.properties file during initialization.

The user can also assign another name for the configuration file, such as the `Engine::FixEngine::init(const std::string& aPropertiesFileName = "")` call parameter.

For example:

```
Engine::FixEngine::init("FE.properties")
```

## Engine parameters

### EngineRoot

The top of the directory tree under which the engine's configuration and log files are kept.

DO NOT add a slash at the end of the directory path.

### ListenPort

**Optional.** Engine's listen port.

Supports multiple values delimited by comma.

### LogFileName

**Optional.** Engine's log file name. If this parameter is commented or empty, `LogFileName = "engine.log"` as default.

Actually represents the alias for the "Log.Engine.File.Name" parameter. If both parameters are specified simultaneously, then:

- If [Log.Engine.Device](#) is not specified, "LogFileName" is used;
- If [Log.Engine.Device](#) is specified, "Log.Engine.File.Name" is used;

### ListenAddress

**Optional.** Engine's local IP address to bind to.

It can be used on a multi-homed host for a FIX Engine that will only accept connect requests to one of its addresses. If this parameter is commented or empty, the engine will accept connections to any/all local addresses.

### LicenseFile

The license file path.

### LogDirectory

This property is the path of the directory in which the logs for all incoming (if `LogIncomingMessages` is set to "true") and outgoing FIX messages are stored.

It is possible to specify a path related to the EngineRoot directory. For example, if `LogDirectory` is set to "logs" then the real path is /logs. The specified directory must exist.

### ConnectAddress

**Optional.** Engine's local IP address to send messages from.

It is used only for multi-homed hosts. If this parameter is commented or empty, the engine will send IP datagrams from any/all local addresses.

### NumberOfWorkers

A number of threads that serve FIX sessions (heartbeat generation/check, message rejecting, message delay delivering, etc). On Linux the recommended value is 2.

On other platforms the recommended value is 10. The value must be integer and greater than zero.



Changing this value will impact upon the performance of FIX Engine.

The most efficient parameter value should be distinguished during testing.



Workers are used for the following operations:

- Process incoming connections (for acceptors) until incoming Logon message is delivered
- Initiate session reconnect (for acceptors) when logout is received and Application marks session to connect again
- Process outgoing connections (for initiators) until TCP connection is established and outgoing Logon is sent in case async connect is used
- Deliver rejected messages back to Application if EnableMessageRejecting = true and message cannot be sent on time
- Deliver delayed messages to Application when first call of process() returns false

## OutOfSequenceMessagesStrategy

**Optional.** This parameter sets strategy for processing out-of-sequence messages. Parameter is available since **FIX Antenna 2.23.0**

Valid values:

- RequestAlways - send resend request with BeginSeqNo (7) = <out-of-sequence message number>, EndSeqNo (16) = 0 on any out-of-sequence message.
- RequestOnce - send the only resend request on the gap. Duplicates of resend requests for the same interval are not sent.
- IgnoreGap - send resend request on gap and process out-of-sequence messages.
- Queue - use temporary queue for storing out-of-sequence messages and process them after the gap is closed.



Is not supported for FAST sessions

Default value: RequestAlways.

## OutOfSequenceMessagesStrategy.QueueSize

**Optional.** The size of temporary queue for processing out-of-sequence messages (measured in a count of messages). The parameter is used if parameter OutOfSequenceMessagesStrategy=Queue. Parameter is available since **FIX Antenna 2.23.0**

Default value is 2000.

A queue should be cleared after exceeding the limit out-of-sequence messages and a new resend request should be sent.

# Session parameters

## LogonTimeFrame

This parameter sets the time period after which a session is non-gracefully terminated if a response is not received to a first "Logon" message (message type A).

The corresponding Logout message is sent to the counterparty. This value is in seconds. The recommended value is 30 seconds for dedicated connections or private networks. Trading connections via the internet will require calibration. If it is set to "0", then the time period is unlimited. The value must be integer and not negative.

## LogoutTimeFrame

This parameter sets the time period after which a session is automatically terminated if a response is not received to a "Logout message" (message type 5).

This value is in seconds. The recommended value is 10 seconds for dedicated connections or private networks. Trading connections via the internet will require calibration. The value must be an integer and greater than 0.

## IntradayLogoutTolerance

An option not to reset sequence numbers after Logout.

Valid values:

- true - enables *IntradayLogoutTolerance* mode;
- false - disables *IntradayLogoutTolerance* mode.

If set to true, sequence numbers are not reset after Logout. Logout sender should initiate session recovery by sending Logon message with SeqNum = <last outgoing=" seqnum="> + 1; expecting reply Logon with SeqNum = <last incoming=" seqnum="> + 1. If a gap is detected, standard message recovery or gap filling process takes place.

If set to false, sequence numbers are reset after Logout.

## ReasonableTransmissionTime

This parameter specifies the delta (increment) to the Heartbeat interval between a TestRequest message being sent by FIX Engine and a Response Heartbeat being received.

The session attains a "telecommunication failed state" if no Response Heartbeat message is received after the normal Heartbeat interval plus delta. For example, if no message (application or session level) is received during the Heartbeat interval then Engine sends a TestRequest message. If the required Response Heartbeat message is not received during Heartbeat interval plus Delta then the session moves to the state "Telecommunication link failed". This parameter is specified in (Heartbeat Interval/100). The recommended value is twenty percent.

### ForceSeqNumReset

This parameter allows to resolve sequence gap problem automatically.

When mode is ON session uses 141(ResetSeqNumFlag) tag in sending/confirming Logon message to reset SeqNum at the initiator or at the acceptor. Valid values:

- "0" or "false" - Disable ForceSeqNumReset mode;
- "1" or "true" - Enable SeqNum reset at first time of session initiation;
- "2" - Enable SeqNum reset for every session initiation.

### DuplicateResendRequestLimit

Specifies how much same ResendRequests received before *Application::onResendRequestLoop* is called.

This option is disabled if value is less than 2.

### AdditionalParsersList

This parameter defines the list of additional (custom) protocols which require registering of the corresponding parsers.

When FIX Antenna is started, all dictionaries from [DictionariesFilesList](#) are loaded. If dictionaries are standard, predefined standard parsers (like FIX40; FIX41; FIX42; FIX43; FIX44; FIXT11) are created basing on them. If any dictionary from [DictionariesFilesList](#) is custom, parser is not automatically created and therefore such parsers should be listed in *AdditionalParsersList* parameter.

Format of the value: *[UNIQUE\_PARSER\_NAME@[FIXT\_PROTOCOL1:]APP\_PROTOCOL1[,APP\_PROTOCOL2...]*, where:

- UNIQUE\_PARSER\_NAME - **optional**, name of the parser;
- FIXT\_PROTOCOL1 - **optional**, session protocol. Required if APP\_PROTOCOL doesn't contain **session messages**;
- APP\_PROTOCOL1 - **required**, application protocol;
- APP\_PROTOCOL2, ... - **optional**, list of other application protocols.

Below are examples of possible parsers with correct format:

- MyFixParser@FIXT11:FIX50SP2, FIX50SP1;
- FIXT11:FIX50SP2, FIX50SP1;
- FIXT11:FIX50SP2;
- FIX44.

These are examples of one entry. Entries can be combined with semicolon to define more than one parser:

```
AdditionalParsersList = FIX44; FIXT11:FIX50SP2; FIXT11:FIX50SP2, FIX50SP1; MyFixParser@FIXT11:FIX50SP2, FIX50SP1
```



Parsers are not used for Market Data Adapters and for FAST Sessions.

### ThirdPartyRoutingIsEnabled

FIX Engine has inbuilt FIX message routing capability and fully supports the "Deliver To On Behalf Of" mechanism as specified by the FIX protocol.

If this parameter is set to "true" then Engine will redirect FIX messages automatically to other FIX sessions it maintains if OnBehalfOfCompID field in the message is defined. If this parameter is set to "false" Engine directs all messages received to the client application.

### DelayedProcessing.DeliveryTriesInterval

This parameter specifies the time interval between attempts to deliver an application level message to a registered client application in the event the application does not confirm receipt and operation upon the message.

The value is specified in milliseconds. The value must be integer and greater than 0. This parameter is required only if the DelayedProcessing.MaxDeliveryTries parameter is specified.

### DelayedProcessing.MaxDeliveryTries

**Optional.** This parameter specifies the number of attempts that will be made to deliver an application level message to the registered client application.

If this value is exceeded then the session will be closed with the logout reason "Application not available". The recommended value is 10. The value must be an integer and not negative.

### **Reconnect.Interval**

This parameter specifies the time interval between reconnection attempts in order to restore a communications link.

This value is specified in milliseconds (seconds\*10-3). The recommended value is 1000 for dedicated connections and private networks. Internet connections require calibration. The value must be an integer and greater than 0.

### **Reconnect.MaxTries**

This parameter specifies the number of attempts to restore the session.

The session is considered as restored if the telecommunication link was restored and the exchange of Logon messages was successful. If it is set to "-1", then the number of attempts is unlimited. This value is integer.

### **MessageTimeToLive**

**Optional.** This parameter sets the time period after which a message rejecting is started while the session doesn't exist.

Value is specified in milliseconds (seconds\*10-3), must be integer and > 0.

### **CheckVersionOfOutgoingMessages**

This parameter is an option whereby the version of FIX protocol used for the outgoing message is validated against that of the established session.

If set to "true" then the application must use the same version of the protocol as the established session otherwise an error occurs. If set to false then the application level message will be sent to the counterparty. The recommended value is "true".



Changing this value will impact the performance of FIX Engine. In the development should be true. In production should be false.

### **ResendMessagesBlockSize**

Resend Request resend messages by blocks.

This parameter defines how many messages proceed in block. The value "0" means that all messages will be resent in one block. The value must be an integer and not less than 0.

The default value is "1000".

### **ResetSeqNumAfter24hours**

**Optional.** An option to send a Logon message with the ResetSeqNumFlag set after each 24 hour period of session's activity to establish a new set of sequence numbers (starting with 1).

The default value is false.



This option does not affect sessions which use version 4.0 of the FIX protocol.

### **DictionariesFilesList**

This parameter contains a list of names of XML files with extensions of the FIX protocols delimited by semicolon.

### **EncryptionConfigFile**

Encryption config file name.

### **TotalOutgoingStorageMemoryLimit**

Specifies the total amount of the memory (in MB) that active session may use.

0 - means infinite. When the limit is reached, the "hardest" sessions will be closed non-gracefully.

### **Debug.LogSessionExtraParameters**


If this property value is true, all session parameters will be printed to the engine.log file.

### **OutOfSequenceMessagesStrategy**


**Optional.** This parameter sets strategy for processing out-of-sequence messages for specific session. Parameter is available since **FIX Antenna 2.23.0**

Valid values:

- **RequestAlways** - send resend request with BeginSeqNo (7) = <out-of-sequence message number>, EndSeqNo (16) =0 on any out-of-sequence message. ResendRequestBlockSize is not supported (will be used strategy RequestOnce).
- **RequestOnce** - send resend request on gap (considering ResendRequestBlockSize parameter). Duplicates of resend requests for the same interval are not sent.
- **IgnoreGap** - send resend request on gap and process out-of-sequence messages.
- **Queue** - use temporary queue for storing out-of-sequence messages and process them after the gap is closed.

 Is not supported for FAST sessions

Default value is engine parameter OutOfSequenceMessagesStrategy.

 Old parameters DeliverAppMessagesOutOfOrder, SuppressDoubleResendRequest must be removed for specified strategy for sessions. Old parameters have higher priority.

### OutOfSequenceMessagesStrategy.QueueSize

**Optional.** The size of temporary queue for processing out-of-sequence messages (measured in a count of messages). The parameter is used if parameter OutOfSequenceMessagesStrategy=Queue. Parameter is available since **FIX Antenna 2.23.0**

Default value is engine parameter OutOfSequenceMessagesStrategy.QueueSize.

A queue should be cleared after exceeding the limit out-of-sequence messages and a new resend request should be sent.

### SocketOpPriority

The priority of the socket Send/Receive operations.

Valid values:

- **EVEN** (default) - share worker thread among all session in the Engine;
- **AGGRESSIVE\_SEND** - use dedicated per session thread to send outgoing messages;
- **AGGRESSIVE\_RECEIVE** - use dedicated per session thread to receive incoming messages;
- **AGGRESSIVE\_SEND\_AND\_RECEIVE** - use dedicated per session threads to send and receive messages;
- **DIRECT\_SEND** - use the current thread for sending, if this would block, performs as "EVEN".

### AggressiveReceiveDelay

**Optional.**

Defines a polling interval in microseconds for reading data from the socket. Works only with aggressive sessions i.e. Engine::AGGRESSIVE\_RECEIVE\_SOCKET\_OP\_PRIORITY or Engine::AGGRESSIVE\_SEND\_AND\_RECEIVE\_SOCKET\_OP\_PRIORITY modes. Setting lower value reduces the latency but causes high CPU utilization.

The default value is 500 microseconds.

 There will not be a delay after an attempt to receive a message if the socket buffer still is not empty.

### SSL

*For Initiator FIX Session:*

**Mandatory** if SSL is used. Enable SSL support for the session.  
Default value is false.

*For Acceptor FIX Session:*


**Optional.** If true, accept SSL connection only. If false, accept both SSL and non-SSL connections for the session.  
Default value is false.

 Introduced in FIX Antenna 2.26

### ParserVersion




Name of unique parser for FIX protocol

 Introduced in FIX Antenna 2.25.1


### Role

Role of the session.

 Introduced in FIX Antenna 2.25.1

### Host

The host for initiator session

 Introduced in FIX Antenna 2.25.1

### Port

TCP port for connection to the remote site.

 Introduced in FIX Antenna 2.25.1

### HBI

The heartbeat Interval in seconds for initiator session

 Introduced in FIX Antenna 2.25.1

### CustomLogonMessageFileName

path to file with a Custom logon message for the initiator.


The default value is empty.

## Storages

### LogIncomingMessages

This property provides an option to log incoming FIX messages (those received) from a counterparty FIX Engine.

They will be stored in the directory specified by the LogDirectory parameter in a file with extension "in".

 Changing this value will impact the performance of FIX Engine. In the development should be true. In production should be false.

### OutgoingMessagesStorageSize

This parameter defines the upper limit to the number of outgoing messages that are resent in the event of a Resend Request.

The recommended value is 20000 if no data on mean activity is known.

 Changing this value will impact the performance of FIX Engine. Large storage increases application working set.

### ExtraSafeMode

If this parameter is true than file streams are flushed after each I/O operation.



Changing this value will impact the performance of FIX Engine. In the development and in production should be true.

### TimestampsInLogs

**Optional.** An option to write timestamps in the log files.

The default value is true.



Changing this value will impact the performance of FIX Engine. In the development should be true. In production should be false.

### EnableIncrementalLogFileCreation

Setting property "true" will reserve 10Mb of disc space for logging.

If log reaches 10Mbs, another 10Mbs will be reserved and so on. When property is "true" FixAntenna performance is greatly increased.

### Persistents.LogIncrementSize

Specifies the amount of memory in bytes reserved for memory mapped file message storage for \*.out/\*.in files.

It defines how FIX Antenna reserves space on the HDD for PersistentMM (in any cases) and Persistent (if EnableIncrementalLogFileCreation is ON). It is recommended to set large enough value to minimize the number of resizes during the storage life time. Each resize can cause a latency spike.



Changing this value will impact the performance of FIX Engine.

### Persistents.IdxIncrementSize

Specifies the amount of memory in bytes reserved for memory mapped file message storage for \*.idx file It defines how FIX Antenna reserves space on the HDD for PersistentMM (in any cases) and Persistent (if EnableIncrementalLogFileCreation is ON).

It is recommended to set large enough value to minimize the number of resizes during the storage life time. Each resize can cause a latency spike.



Changing this value will impact the performance of FIX Engine. In the development should be true. In production should be false.

### SplitPersistentMsgStorage.BackupDir

This property is the path to the directory, where sliced message storage will search for log files in case they are not found in the primary location.

It is the user's responsibility to move files from the primary location to backup. The default value is empty - backup path disabled.

### SplitPersistentMsgStorage.MaxSliceSize

Specifies the maximum size of one slice for sliced message storage.



Storage slices are created with step, specified with *Persistents.LogIncrementSize* parameter value. If MaxSliceSize value is no multiple of *Persistents.LogIncrementSize*, result file size will be less, than specified. If value is less than LogIncrementSize, LogIncrementSize value will be used.

Recommended value: any multiple of Persistents.LogIncrementSize Default value is 500000000 - 500Mb.

### MessageStorage.IndexRebuildEnabled

**Optional.** If this parameter is true, FIX Engine will try to restore the index file from the log file in case the index file is missed or corrupted. The default value is false.



Please be careful when you enable this property: restoring process uses some heuristic algorithms, and some data (not messages) could be missed.

### MessageStorage.RequireRejectFlagSupport

**Optional.** If this option is enabled, FIX Engine will not rebuild index files for message storages, that does not support embedded Reject Flag Marker. The default value is true.

## MessageStorage.TimestampUnit

This property specifies precision of timestamps in session logs (.in / .out storage files). Possible values are: "millisecond", "microsecond", "nanosecond".

**Optional.** The property is global for all sessions. The default value is "millisecond".

## Unregistered Acceptors parameters

### CreateUnregisteredAcceptorSession

This parameter provides an option whereby FIX Engine will accept a FIX session for which it has no registered application (an acceptor).

If set to "true" Engine accepts incoming connection and creates corresponding Session object. If there is no Application associated with the session, all application level messages are rejected with Application Level Reject(3) message. If an application is registered, the behavior is as standard. If set to false then Logon messages are ignored and incoming connection is dropped. Property is obsolete, *UnregisteredAcceptor.CreateSession* should be used instead.

### UnregisteredAcceptor.CreateSession

This parameter provides an option whereby FIX Engine will accept a FIX session for which it has no registered application (an acceptor).

If set to "true" Engine accepts incoming connection and creates corresponding Session object. If there is no Application associated with the session, all application level messages are rejected with Application Level Reject(3) message. If an application is registered, the behavior is as standard. If set to false then Logon messages are ignored and incoming connection is dropped.

### UnregisteredAcceptor.IgnoreSeqNumTooLowAtLogon

This parameter allow to resolve seqNum too low problem at logon.

When it true - the session continues with received seqNum.

### UnregisteredAcceptor.maxMessagesAmountInBunch

FA able to join packages that wait for sending into the socket, these parameters control how many messages could be joined.

0 means infinite. The value should be less than 1000000.

### UnregisteredAcceptor.RejectMessageWhileNoConnection

When true, unregistered acceptors will reject messages in case they couldn't be sent during the interval specified in the *MessageTimeToLive* parameter.

You can find more information about how to use this parameter here: [How to use "RejectMessageWhileNoConnection" parameter](#)

### UnregisteredAcceptor.tcpBufferDisabled

When true, the TCP buffer (Nagle algorithm) will be disabled for unregistered acceptors.

Otherwise, TCP may join and enqueue small packages until timeout ends.

### UnregisteredAcceptor.SessionStorageType

Default storage type of created unregistered sessions.

By default persistentMM storage type used. Use "transient" value to use transient storage for the sessions.

## Message Validation parameters

### MessageMustBeValidated

Mandatory. Defines whether each [application message](#) is checked for the existence of required tags.

*Valid values:* "true" | "false"

If set to 'true' then all application-level messages are checked. If set to 'false' then the responsibility for message validity rests with the [counterparty](#).

**Note:** Session level messages are checked in all cases. The recommended setting is 'false'. Changing this value will impact the FIX Engine performance. Use 'true' for debugging and 'false' to increase performance.

*Default value:* MessageMustBeValidated = **false**

### VerifyTagsValues

Optional. Defines whether each [application message](#) is verified for tag values

*Valid values:* "true" | "false"

If set to 'true' then all messages will be verified. If set to 'false' then the responsibility for message verification rests with the counterparty.

**Note:** Ignored if [MessageMustBeValidated](#) = false

*Default value:* VerifyTagsValues = **true**

### ProhibitUnknownTags

Optional. Defines whether each [application message](#) is checked for the unknown tags.

*Valid values:* "true" | "false".

If set to 'false' then all application messages with unknown tags will be processed. If set to 'true' then the engine will reject messages with unknown tags.

**Note:** Ignored if [MessageMustBeValidated](#) = false

*Default value:* ProhibitUnknownTags = **false**

### VerifyRepeatingGroupBounds

Optional. Defines whether each [application message](#) is verified for repeating group size.

*Valid values:* "true" | "false".

If set to 'true' then the engine will reject messages with incorrect repeating group size. Otherwise, the message will be passed to the application layer. In this case, the responsibility for message validity rests with the counterparty.

**Note:** Ignored if [MessageMustBeValidated](#) = false

*Default value:* VerifyRepeatingGroupBounds = **true**



VerifyRepeatingGroupBounds = "false" has no effect since FIX Antenna 2.26

### IgnoreUnknownFields

*Valid values:* "true" | "false".

Mandatory. If set to 'true' then the unknown (undefined or user's) fields would not affect the message processing. In this case, FIX Engine removes unknown fields so that performance can be increased. If set to 'false', then the unknown tags will remain in the message consuming memory and reducing performance.

**Note:** Ignored if [MessageMustBeValidated](#) = false

*Default value:* IgnoreUnknownFields = **false**

### Validation.CheckRequiredGroupFields

Optional. Controls the validation of required fields in the repeating group.

*Valid values:* "true" | "false".

If set to 'true' then repeating groups will be checked for the presence of required fields. If set to 'false' then the responsibility for repeating group validity rests with the counterparty. The recommended setting is 'true'.

**Note:** Ignored if [MessageMustBeValidated](#) = false

*Default value:* Validation.CheckRequiredGroupFields = **true**

### AllowEmptyFieldValue

Mandatory. Defines whether the raw message may contain tags without values.

*Valid values:* "true" | "false".

If set to 'true', the raw message may contain tags without values that will be ignored. Otherwise, an exception will be fired.

*Default value:* AllowEmptyFieldValue = **false**

# Logging parameters

## Log.File.Format

Customformatoflogfilerecord. Can be defined separately for each log file e.g. Log.Version.File.Format = ...

Format string can contain any symbols and keywords:

- %% - put '%' symbol;
- %level - put severity level name: 'ERROR', 'WARN' etc.;
- %tablevel - put extra spaces for vertical alignment of records after %level;
- %date{FIX} - date in 'YYYYMMDD-HH:MM:SS.sss' format;
- %date{ISO8601} - date in 'YYYY-MM-DD HH:MM:SS.sss' format;
- %date - date in 'DD MMM YYYY HH:MM:SS.sss' format;
- %timezone - time zone abbreviation or name (setup TZ env variable to take abbreviation in windows) or 'UTC';
- %logger - log category, component;
- %thread - thread id;
- %thread\_name - prints a name of the thread
- %message - logged message.

Default value:

```
Log.File.Format = [%level] %tablevel%date{FIX} [%thread] [%logger] - %message
```

## Log.File.CreateUniqueFileName

If true new unique file name will be created. Log.File.Name value is a pattern.

If false Log.File.Name will be used as a file name.

If *Log.Device* is not specified explicitly (in other words, *Log.Device* = "File" as default), *Log.File.CreateUniqueFileName* = "false" as default.

## Log.Cycling

Enables/disables log file cycling.

Valid values:

- true - to enable cycling;
- false - to disable cycling.

If set to true, then repeating records in logs will be controlled and replaced with the single "cycling record" according to the specified cycling parameters. See *cycling* parameters described below for details.

## Log.Cycling.Ignore

Number of repeating records to be placed to log before cycling is started.

Example: Log.Cycling.Ignore = 3

## Log.Cycling.BlockSize

Number of repeating records to be accumulated (hidden) before write the "cycle record" to the log.

Example: Log.Cycling.BlockSize = 10

## Log.Cycling.Multiplier

Multiplier for the BlockSize. If BlockSize number of messages is accumulated and the same message still appears then next BlockSize is calculated as the previous one multiplied by Multiplier.

Example: Log.Cycling.Multiplier = 10

## Log.Device

Default target devices.

Valid values:

- File - messages will be written to the file;
- Console - messages will be written to the console;
- EventLog (Windows specific) - messages will be written to the Windows Event Log;
- WinDebug (Windows specific) - messages will be written to the OutputDebugString;
- Syslog (Linux/Unix specific) - messages will be written to the system log.

- Log4Cplus - messages will be written to [Log4Cplus](#) (since FIXEdge version 6.7)

If *Log.Device* is not specified explicitly, *Log.Device* = "File" as default.

The properties of the device are in the "Log.<name>" section. For instance Log.File contains all properties of the File device.

Multiple devices can be chosen - the names of devices should be separated by a space in this case. E.g.: by setting **Log.Device = File Log4Cplus** logging will be performed with both creating standard log files and forwarding to Log4Cplus.



Log4Cplus can be used to send log messages to external services such as Splunk. FIXEdge does not guarantee each log message delivery when:

1. Log message receiver is not available (not listening on the given port).
2. Log message receiver throughput is less than log flow produced by FIXEdge.

In these cases, FIXEdge will drop the messages it was unable to send.

## Log.EventLog.EventSource

A string that specifies the name of the source. The source name must be a subkey of a log file entry under the EventLog key in the registry.

For example, "WinApp" is a valid source name if the registry has the following key:

"HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services\Event Log\Application\WinApp".

## Log.File.AutoFlush

If set to 'true', then buffer will be flushed after each logging call.

If set to false then flush is not called.

Setting to true decreases program performance; setting to false increases a risk of records loss in case of program failure.

## Log.File.Backup.Time

**Optional.** Local time when logs should be backed up. When the specified backup time comes, the scheduler moves log files to the specified [backup directory](#).

Should be in format 'hh:mm' or 'hh:mm:ss'.

## Log.File.Locked

If set to true then lock file <Log.File.RootDir/Log.File.Name>.lock will be created. This will prevent other instance of the application use the same file.

Valid values:

- true - lock log file;
- false - do not lock log file.

## Log.File.Recreate

If true then file will be recreated on each start.

If false then new records will be appended to the existent file.

## Log.File.RootDir

The root directory for log files. It will be concatenated with each file name specified by 'Log.File.Name'. If *Log.Device* is not specified explicitly (in other words, *Log.Device* = "File" as default), *Log.File.RootDir* = [LogDirectory](#) as default.

You can find more information about how to configure paths to log files here: [How to configure paths to FIXEdge log files](#).

## Log.File.Rotate

Enables/disables log file rotation.

Valid values:

- true - to enable rotation;
- false - to disable rotation.

If set to true, after the specified file's size is reached, it is renamed and a new log file is created. See rotation parameters described below for details.

## Log.File.Rotate.SizeLimit

Sizeoffileinbytes.Whenlogfilereaches this size it is renamed (follow standard rotation approach) and the new file is created.

Example: Log.File.Rotate.SizeLimit = 104857600

### **Log.File.Rotate.FileLimit**

Number of files to be involved in rotation. When the number is exceeded the older file is removed.

Example: Log.File.Rotate.FileLimit = 10

### **Log.File.Rotate.Dir**

Rotation directory. Directory to store rotated files.

Example: Log.File.Rotate.Dir = <%SERVICE\_NAME%>/log

### **Log.File.TimeZone**

Time zone of the log message timestamp.

Valid values:

- Local - display time as local;
- UTC - UTC.

### **Log.File.Name**

File name. If more than one category uses files with the same name the same file will be used simultaneously.

### **Log.DebugsOn**

Enable Debug messages.

Valid values:

- true - to enable Debug messages;
- false - to disable Debug messages.

### **Log.ErrorsOn**

Enable Error messages.

Valid values:

- true - to enable Error messages;
- false - to disable Error messages.

### **Log.FatalsOn**

Enable Fatal Error messages.

Valid values:

- true - to enable Fatal Error messages;
- false - to disable Fatal Error messages.

### **Log.NotelsOn**

Enable Note messages.

Valid values:

- true - to enable Note messages;
- false - to disable Note messages.



Starting from FIX Antenna 2.10.16.3, Note messages are indicated as **INFO** records in logs.

### **Log.WarnsOn**

Enable Warning messages.

Valid values:

- true - to enable Warning messages;
- false - to disable Warning messages.

## Log.TracelsOn

Enable Trace messages.

Valid values:

- true - to enable Trace messages;
- false - to disable Trace messages

## Log.OpenLDAP.TracelsOn

Enables writing of debug log records from OpenLDAP library to the logs. The parameter has been added since FIXEdge 6.2.

Valid values:

- true - to enable OpenLDAP library debug-level logging;
- false - to disable OpenLDAP library debug-level logging

# Affinity parameters

## WorkerCpuAffinity

Defines an affinity mask for all worker pool threads.

Affinity is specified as bit mask where the serial number of the bit is the serial number of the core if read from the right to the left. Mask should be presented **in the decimal system**.

**i** E.g. to specify second, third and fourth cores, the affinity in bytes will look like "1110". It is "14" in the *decimal* system. So, for proper use the property should be set in the following way:

```
WorkerCpuAffinity = 14
```

Default value is 0. It means that the engine will not change thread affinity mask.

## HelperCpuAffinity

Defines an affinity mask for dispatcher and other auxiliary threads.

Affinity is specified as bit mask where serial number of the bit is the serial number of the core if read from the right to the left. Mask should be presented **in the decimal system**.

**i** E.g. to specify third and fourth cores, the affinity in bytes will look like "1100". It is "12" in the *decimal* system. So, for proper use the property should be set in the following way:

```
HelperCpuAffinity = 12
```

Default value is 0. It means that the engine will not change thread affinity mask.

## CpuAffinity

Specifies affinity mask for dedicated threads of session. It makes sense only for aggressive modes.

Affinity is specified as bit mask where serial number of the bit is the serial number of the core if read from the right to the left. Mask should be presented **in the decimal system**.

**i** E.g. to specify fourth, sixth and eighth cores, the affinity in bytes will look like "10101000". It is "168" in the *decimal* system. So, for proper use the property should be set in the following way:

```
CpuAffinity = 168
```

Default value is 0. It means that the engine will not change thread affinity mask.



## RecvCpuAffinity

Specifies affinity mask for dedicated receiving thread of session. It makes sense only for aggressive receive mode.

Affinity is specified as bit mask where serial number of the bit is the serial number of the core if read from the right to the left. Mask should be presented ***in the decimal system***.



E.g. to specify third, sixth, seventh and eighth cores, the affinity in bytes will look like "11100100". It is "228" in the *decimal* system. So, for proper use the property should be set in the following way:

```
RecvCpuAffinity = 228
```

Default value is 0. It means that the engine will not change thread affinity mask.

## SendCpuAffinity

Specifies affinity mask for dedicated sending thread of session. It makes sense only for aggressive send mode.

Affinity is specified as bit mask where serial number of the bit is the serial number of the core if read from the right to the left. Mask should be presented ***in the decimal system***.



E.g. to specify second, fifth and seventh cores, the affinity in bytes will look like "01010010". It is "82" in the *decimal* system. So, for proper use the property should be set in the following way:

```
SendCpuAffinity = 82
```

Default value is 0. It means that the engine will not change thread affinity mask.

# Backup parameters

## BackupDirectory

Relative path to the backup folder.

This folder will be used for message storage files of the backup connections.

## EnableAutoSwitchToBackupConnection

Defines if there should be an auto switch to the backup connection in case the primary connection is used and it is interrupted.

Valid values:

- true - to enable EnableAutoSwitchToBackupConnection mode;
- false - to disable EnableAutoSwitchToBackupConnection mode.

## EnableCyclicSwitchBackupConnection

Defines if there should be an auto switch to the primary connection in case backup connection is used and it is interrupted.

Valid values:

- true - to enable EnableCyclicSwitchBackupConnection mode;
- false - to disable EnableCyclicSwitchBackupConnection mode.

# Monitoring parameters

Monitoring parameters can be defined as defaults (Monitoring.AdminSessionDef.<Name\_Of\_Parameter>) or for a particular administrative session (Monitoring.AdminSession.<Name\_Of\_Administrative\_Session>.<Name\_Of\_Parameter>). Below the defining of defaults is presented.

## Monitoring.AdminSessionDef.DisableTCPBuffer

When true TCP buffer (Nagle algorithm) will be disabled for the session.

### **Monitoring.AdminSessionDef.EncryptMethod**

The expected value of the encryption method.

### **Monitoring.AdminSessionDef.ForceSeqNumResetMode**

Force SeqNum reset mode.

An option to use 141 tag in Logon message to reset sequence number.

### **Monitoring.AdminSessionDef.IgnoreSeqNumTooLowAtLogon**

When true, session ignore 'SeqNum too low' at incoming Logon message and continue with received SeqNum.

### **Monitoring.AdminSessionDef.IntradayLogoutToleranceMode**

Intraday logout tolerance mode.

An option to reset or not to reset sequence numbers after Logout.

### **Monitoring.AdminSessionDef.MaxMessagesAmountInBunch**

Enqueued outgoing messages could merged and sent as a single buffer.

This parameter controls how many messages could be merged into the bunch. The 0 means an infinite amount.

### **Monitoring.AdminSessionDef.Password**

The expected value of the Password (Tag 554) field in the incoming Logon message.

If the real value is not equal to the expected one then the session is disconnected without sending a message and the error condition is generated in the log output.

### **Monitoring.AdminSessionDef.SenderLocationID**

SenderLocationID (tag 142) - assigned value used to identify specific message originator's location (i.e. geographic location and/or desk, trader).

### **Monitoring.AdminSessionDef.SocketOpPriority**

Priority of the socket SendReceive operations.

Valid values:

- EVEN (default) - share worker thread among all session in the Engine;
- AGGRESSIVE\_SEND - use dedicated thread to send outgoing messages;
- AGGRESSIVE\_RECEIVE - use dedicated thread to send outgoing messages;
- AGGRESSIVE\_SEND\_AND\_RECEIVE - use dedicated per session threads to send and receive messages;
- DIRECT\_SEND - use the current thread for sending, if this would block, performs as "EVEN".

### **Monitoring.AdminSessionDef.SourceIPAddress**

The expected value of the source IP address.

If the real value is not equal to the expected one then the session is disconnected without sending a message and the error condition is generated in the log output.

### **Monitoring.AdminSessionDef.TargetLocationID**

TargetLocationID\_ (tag 143) - assigned value used to identify specific message destination's location (i.e. geographic location and/or desk, trader).

### **Monitoring.AdminSessionDef.TargetSubID**

TargetSubID (tag 57) - assigned value used to identify specific individual or unit intended to receive message.

### **Monitoring.AdminSessionDef.Username**

The expected value of the Username (Tag 553) field in the incoming Logon message.

If the real value is not equal to the expected one then the session is disconnected without sending a message and the error condition is generated in the log output.

### **Monitoring.ListenPort**

The engine TCP listen port that is enabled for administrative sessions. If the port is configured it will be added to set of engine listen ports and administrative sessions will be accepted on this port only. If not exists or empty common engine listen ports are used to accept administrative sessions.



To enable SSL on Monitoring.ListenPort, the port number must be also added to ListenSSLPort parameter

### Measuring.Enable

Enables/disables Engine measuring.

Valid values:

- true - to enable Engine measuring;
- false - to disable Engine measuring.

If this parameter omitted, it is "false" by default.

### Monitoring.Enable

Enables/disables Engine monitoring.

## SSL parameters

### ListenSSLPort

**Optional.** Engine's listen SSL port.

Supports multiple values delimited by comma.

Must not contain the value of **ListenPort**



This parameter is similar to ListenPort. It is the list of ports, where SSL connection will be accepted.

### SSLCACertificate

**Optional.** Engine's SSL CA certificate file path.

### SSLCertificate

**Mandatory** if ListenSSLPort is not empty. Engine's SSL certificate file path.

### SSLPrivateKey

**Mandatory** if ListenSSLPort is not empty. The path to the file containing certificate private key. Certificate and private key can be the same file.



FIX Antenna 2.26 supports certificates in PEM, PFX and DER formats along with private keys in PEM and DER formats. Earlier versions support \*.pem certificates only.

### SSLCheckPrivateKey

**Optional.** Engine's SSL option whatever to check if the private key matches the certificate.

Default value is false.



Deprecated since FIX Antenna 2.26. Correspondence between the key and certificate will be checked automatically. FIX Antenna does not establish a session if the key doesn't correspond to the certificate.

### SSLProtocols

**Mandatory.** Engine's SSL protocols to use.

It is a comma-separated list of protocols to use. Valid values are **SSLv2**, **SSLv3**, **TLSv1**, **TLSv1\_1**, **TLSv1\_2**.



Since some security issues with SSL v3 and RC4 encryption algorithm have been discovered recently and TLS 1.0, TLS 1.1 also contain known vulnerabilities, we strongly recommend using TLS 1.2 protocol.

### SSLRequireClientCertificate

**Optional.** Whatever client's certificate is mandatory.

Decline clients without a valid certificate if true. The default value is false.



Deprecated since FIX Antenna 2.26. Use **SSLValidatePeerCertificate** parameter instead.

### SSLCertificatePassword

**Optional.** A certificate's password.



Introduced in FIX Antenna 2.26

### SSLPrivateKeyPassword

**Optional.** A private key's password.



Introduced in FIX Antenna 2.26

### SSLValidatePeerCertificate

**Optional.** If set to "true" then client connections without a certificate will be rejected.

The default value is false.



Introduced in FIX Antenna 2.26. Use **SSLValidatePeerCertificate** parameter instead of deprecated **SSLRequireClientCertificate**.

### SSLCiphersList

The parameter is optional. The default OpenSSL built-in configuration is used in case of the parameter absence.

The list is passed as is to OpenSSL during engine initialization in case of acceptor sessions (acceptor sessions are sharing the same configuration) and during session initialization in case of initiator sessions. See a list of the supported ciphers and list format is at official OpenSSL site: <https://www.openssl.org/docs/man1.0.2/apps/ciphers.html>



Introduced in FIX Antenna 2.25.1

## TCP Protection parameters



The set of parameters below is applicable since FIX Antenna C++/.Net 2.24.0

### ProtectionTCP.Enabled

The property enables TCP protection.

Valid values: true/false. Default value is false.

### ProtectionTCP.WaitLogon

The property to specify connection timeout (in milliseconds) of waiting for logon.

Disable when equal to 0. Default value is 0

The property will be enabled only if *ProtectionTCP.Enabled = true*

### ProtectionTCP.SizeWaitHostMax

The property to specify a maximum of connections waiting for logon from one host. When the number of connections from the same host exceeds the specified value of the property, new connections will be rejected.

Disable when equal to 0. Default value is 0

The property will be enabled only if *ProtectionTCP.Enabled = true*

### **ProtectionTCP.SizeBufferMax**

Specifies the maximum size (in bytes) of the buffer to be able to avoid the situation when users end high-loaded garbage.

When the limit is exceeded, the connection will be closed and the corresponding reason will be written to the log.

The minimum size of the property which can be specified is 262144. If a lower value is specified in the property, it will be replaced with 262144 on the start of the FIXEdge.

If 0 is specified, there is no limit for the size of the buffer.

The default value is 0.

The property will be enabled only if *ProtectionTCP.Enabled = true*

## Other parameters

AnchorFile

### **Myricom.Dbl.TcpListenAddresses**

**Optional.** DBL TCP listen address and port.

Format: <XXX.XXX.XXX.XXX>:<port>.

### **Myricom.Dbl.UseSocketAdaptor**

**Optional.** This property tells how will be initialized Myricom DBL TCP API.

If true it will be used SocketAdaptor to emulate DBL API via generic sockets. Default value is false.

### **EnableDnsEntrySpoofingDetection**

When it's set to true there is being enabled backward resolve based DNS entry spoofing detection mechanism in DNS related functions.



This option is disabled by default.



If the parameter is enabled, a user should use Fully Qualified Domain Name (FQDN) or IP addresses as configuration/function parameters (e. g. 'workstation1.epam.com' instead of 'workstation1'). Otherwise, errors about DNS spoofing will be reported.

### **KeepMillisecondsInUTCTimeFields**

Defines how to serialize UTCTimestamp or UTCTimeOnly field values:

0 - default mode - if the value of milliseconds is not equal to 0 then it will be serialized.

1 - always serialize milliseconds

2 - always truncate milliseconds

By default, this parameter is 0.



It is a global option for FixEngine. It works only when a value is passed as Engine::UTCTimestamp or Engine::UTCTimeOnly.

