

# Configuring other FIXEdge properties

- [Overview](#)
- [TCP Protection parameters](#)
  - [ProtectionTCP.Enabled](#)
  - [ProtectionTCP.WaitLogon](#)
  - [ProtectionTCP.SizeWaitHostMax](#)
  - [ProtectionTCP.SizeBufferMax](#)
- [Transport Adaptors common parameters](#)
  - [FIXEdge.AllowNGTStatusNotifications4TAClient](#)
  - [TransportLayer.UnknownClientsAllowed](#)
- [Control Centre parameters](#)
  - [ControlCentre.XMLSockMonitor.Port](#)
  - [ControlCentre.Lock.File.Name](#)
- [Settings of FIXICC Functionality Availability](#)
  - [FIXICC.Schedules](#)
- [Monitoring Memory Usage](#)
  - [MemoryWatch.Limit](#)
  - [MemoryWatch.Interval](#)
- [Logging for lifecycle FIXEdge events](#)
  - [Log.Events.LogCategory](#)
  - [Log.Events.LogLevel](#)
  - [Log.Events.Event.<NameOfEvent>](#)
  - [Log.Events.Event.<NameOfEvent>.LogCategory](#)
  - [Log.Events.Event.<NameOfEvent>.LogLevel](#)

## Overview

The settings in this section describe only FIXEdge parameters. These parameters don't belong to [FIX sessions parameters](#) or [FIX Engine parameters](#).

These settings of the FIXEdge are defined into '*FIXEdge.properties*' configuration file.

## TCP Protection parameters

This section describes properties in [FIXEdge.properties](#) file which are responsible for TCP protection.

### ProtectionTCP.Enabled

The property enables TCP protection.

Valid values: true/false. **Default value:** ProtectionTCP.Enabled = false.

Example: ProtectionTCP.Enabled = true

### ProtectionTCP.WaitLogon

The property to specify connection timeout (in milliseconds) of waiting for logon. When the time comes out, the connection is closed with corresponding reason:

```
[INFO] 20160706-07:40:53.032 [1274500] [Engine] - Logon message wasn't received in a given time interval(1000 ms) from 127.0.0.1:60376
[INFO] 20160706-07:40:53.032 [1274500] [Engine] - Incoming TCP connection was closed (from 127.0.0.1:60376).
```

Disable when equal to 0. **Default value:** ProtectionTCP.WaitLogon = 10000

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

### ProtectionTCP.SizeWaitHostMax

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

```
[INFO] 20160706-07:30:21.566 [1271868] [Engine] - Incoming TCP connection was rejected (from 127.0.0.1:60300). Exceeded limit of connections (5) from a single host.
```

Disable when equal to 0. **Default value:** `ProtectionTCP.SizeWaitHostMax = 5`

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 user sends high-loaded garbage.

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

```
[INFO] 20160706-07:21:19.809 [1271256] [Engine] - Incoming TCP connection was detected (from 127.0.0.1:60258).  
[INFO] 20160706-07:21:19.965 [1271896] [Engine] - Exceeded buffer received limited(1000000 ) from 127.0.0.1:60258  
[INFO] 20160706-07:21:19.966 [1271896] [Engine] - Incoming TCP connection was closed (from 127.0.0.1:60258).
```

When connection is re-established after closure, Client will send Logon message with `MsgSeqNum = <last outgoing sequence number> + 1`. As a result, FIXEdge will send ResendRequest and will receive the same message which caused disconnection. Thus, connection will be closed again.

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.

Default value: **`ProtectionTCP.SizeBufferMax = 0`**.

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

For more information please read the article [How to configure TCP protection in FIXEdge](#).

## Transport Adaptors common parameters

The section contains fields which are common for all Transport Adaptors.

### FIXEdge.AllowNGTStatusNotifications4TAClient

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

When the option is true, FIX Edge uses the latest Client's status in Status Notification message. When option is false, FIX Edge sends Client's status 'Terminated Correctly' in Status Notification message;

Default value is **`false`**.

### TransportLayer.UnknownClientsAllowed

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

When the option is true, Transport Layer allows to call onLogout() callback for the TA Clients without onLogon() call. When the option is false, Transport Layer ignores onLogout() callbacks for the client that didn't call onLogon() before.

Default value is **false**.

## Control Centre parameters

### ControlCentre.XMLSockMonitor.Port

XML Sock monitoring port to accept requests from the special tool FEAdminConsole.

Example: ControlCentre.XMLSockMonitor.Port = 8902

### ControlCentre.Lock.File.Name

File that prevents from starting multiple FIXEdge instances at the same directory.

Example: ControlCentre.Lock.File.Name = FIXEdge1/log/\_fixedge.lock



Starting several FIXEdge instances at the same directory will cause an error:

ERROR: Cannot create lock file './FIXEdge1/log/\_fixedge.lock'. Possibly another instance of the application is running using the same directory.

## Settings of FIXICC Functionality Availability

### FIXICC.Schedules

*Optional*. Defines if the [new schedule management functionality](#) should be available in FIXICC or not.

- If `FIXICC.Schedules = true`, new schedule management functionality is available in FIXICC
- If `FIXICC.Schedules = false` or `FIXICC.Schedules` is not specified, new schedule management functionality is not available in FIXICC.

*Default value*: false.

## Monitoring Memory Usage

### MemoryWatch.Limit

*Optional*. Enable this parameter to force engine to monitor RAM capacity occupied by FIXEdge.

Once value is reached and exceeded FIXEdge generates [OnNotificationEvent](#) to notify about exceeding the available memory.

Value set in Mb.

### MemoryWatch.Interval

*Optional*. Frequency of checking occupied memory.

Value set in seconds.

#### Example

```
FIXEdge.MemoryWatch.Limit = 100
FIXEdge.MemoryWatch.Interval = 60
```

Described functionality is implemented in FIXEdge Windows version only.

## Logging for lifecycle FIXEdge events

Since FIXEdge version 6.7.0 the following lifecycle events are supported:

Name	Description
------	-------------

AppStarting	Application is being to be started and log system is initialized
AppStarted	Properties are loaded and sessions are runned
AppReady	All components of application was started
AppFailed	Application is crashed
AppSigTermDetected	Signal SIGINT or SITERM is detected
AppShutdown	Application is being to shut down (stop sessions / destroy objects)
AppComplete	All work is done, stopped as planned

The events are logged and FIXEdge can notify other system about events

Parameters for setting up logging of these events are described in this section.



Events can be forwarded to ArcSight

See [How to configure forwarding FIXEdge lifecycle events to ArcSight](#) for more information

### **Log.Events.LogCategory**

*Mandatory.* Default category for all events

### **Log.Events.LogLevel**

*Optional.* Default log level for all events

*Valid values:* TRACE, DEBUG, INFO, ERROR, FATAL

Default value: INFO

### **Log.Events.Event.<NameOfEvent>**

*Optional.* Defines the logging format (pattern).



if the parameter is not defined, the event logs will not be generated



See [Format of event entries for transfer to ArcSight](#) for more information

### **Log.Events.Event.<NameOfEvent>.LogCategory**

*Optional.* Log category for a single event

Default value: value of Log.Events.LogCategory parameter

### **Log.Events.Event.<NameOfEvent>.LogLevel**

*Optional.* Log level for a single event

*Valid values:* TRACE, DEBUG, INFO, ERROR, FATAL

Default value: value of Log.Events.LogCategory parameter