

FIX Engine latency tuning guide

- Latency tuning
 - `FixEngine::createSession`
 - Receiving
 - `engine.properties`
 - `SessionExtraParameters`
 - Sending
 - `engine.properties`
 - `SessionExtraParameters`
 - `Session::put`

Latency tuning

FixEngine::createSession

- Session persistent mode (ordered by latency: the best is on the top):
 - `null_storageType` – should be chosen if session will **not** resend last sent messages by request and recovery on start will be performed by user
 - `transient_storageType` – should be chosen if session will resend last sent messages by request and recovery on start will be performed by user
 - `persistentMM_storageType` – should be chosen if session will resend last sent messages by request and recovery on start will be performed by FIX antenna
 - `persistent_storageType` – should be chosen if `persistentMM_storageType` is not supported by OS

Receiving

engine.properties

- `LogIncomingMessages = false`
- `ThirdPartyRoutingIsEnabled = false`
- `EnableIncrementalLogFileCreation = true` (for persistent mode)
- `TimestampsInLogs = false`
- `DictionariesFilesList` – to get best latency user should create FIX protocol XML file with minimal message/field set. Only used fields should be listed in the message. Path to the XML file should be passed to `DictionariesFilesList` parameter in the `engine.properties`. `SessionExtraParameters::validation_.ignoreUnknownFields_` should be true.

SessionExtraParameters

- `SessionExtraParameters::validation_.isEnabled = false`
- `SessionExtraParameters::validation_.ignoreUnknownFields_ = true`
- `validateChecksum_ = false`; More about validation parameters is here: [How to configure FIX messages validation](#)
- `RecvCpuAffinity_` – you can find the description of this parameter here: [FIX Engine parameters#RecvCpuAffinity](#)
- `socketPriority_ |= Engine::AGGRESSIVE_RECEIVE_SOCKET_OP_PRIORITY`
- `aggressiveReceiveDelay_ = 0`; this will cause 100% core usage. For Linux see also: [How to use SO_BUSY_POLL socket option](#)

Sending

engine.properties

- `CheckVersionOfOutgoingMessages = false`
- `TimestampsInLogs = false`
- `EnableIncrementalLogFileCreation = true` (for persistent mode)
- `DictionariesFilesList` – to get best latency user should create FIX protocol XML file with minimal message/field set. Only used fields should be listed in the message. Path to the XML file should be passed to `DictionariesFilesList` parameter in the `engine.properties`. `SessionExtraParameters::validation_.ignoreUnknownFields_` should be true.

SessionExtraParameters

- `SessionExtraParameters::validation_.isEnabled = false`
- `SessionExtraParameters::validation_.ignoreUnknownFields_ = true`
- `generateChecksum_ = false`; if remote side does not validate `Checksum(10)` field, user can disable calculation of this field
- `SendCpuAffinity_` – you can find the description of this parameter here: [FIX Engine parameters#SendCpuAffinity](#)
- `disableTCPBuffer_ = true`
- `socketPriority_ |= Engine::AGGRESSIVE_SEND_SOCKET_OP_PRIORITY`

Session::put

- `PutOptions.overrideSendingTime_ = false`; With this option user should get current system time only once and copy obtained value to `SendingTime` and `TransactTime` (if defined) field
- `PreparedMessage` – where it is possible user should use this class instead of `FIXMessage`
- If `PreparedMessage` is not suitable, you should reuse sent `FIXMessage` objects