Quickfix Setup Guide for ElectronX

Version: 1.0

Date: March 2026

DOCUMENT PURPOSE & SCOPE

Who This Is For:

- Developers integrating with ElectronX FIX API for the first time
- Teams experiencing QuickFix configuration issues
- Anyone troubleshooting session management, message replay, or logging problems

What This Covers:

- QuickFix library selection and versions
- Complete configuration file examples (copy-paste ready)
- Common configuration mistakes and how to avoid them
- SSL/TLS connectivity setup (STunnel vs. direct connection)
- Logging configuration and troubleshooting
- Session sequence number management
- Message replay functionality

What This Does Not Cover:

- FIX protocol fundamentals (see FIX Specification)
- ElectronX-specific message formats (see FIX Specification)
- Business logic implementation

THIRD-PARTY SOFTWARE DISCLAIMER

This QuickFix Setup Guide is provided for informational purposes only to assist ElectronX customers who choose to use QuickFix libraries for FIX protocol implementation.

Important Notices:

- **No Endorsement:** ElectronX does not endorse, recommend, or require the use of QuickFix or any specific FIX engine implementation. This guide does not constitute an endorsement of QuickFix over other available FIX libraries.
- **No Affiliation:** ElectronX is not affiliated with, does not own, and is not responsible for the QuickFix project or any of its implementations (QuickFIX, QuickFIX/N, QuickFIX/J, etc.). QuickFix is independently developed and maintained open-source software.
- **No Warranty:** This guide is provided "as-is" without warranty of any kind. ElectronX makes no representations regarding the accuracy, completeness, or suitability of this information for any particular purpose.
- **No Support Obligation:** ElectronX technical support only covers ElectronX FIX gateways and specifications. We do not provide support for QuickFix library installation, configuration, or troubleshooting beyond what is documented here.
- **Customer Responsibility:** Customers are solely responsible for selecting, implementing, testing, and maintaining their FIX client software. ElectronX is compatible with any standards-compliant FIX 5.0 SP2 implementation.

Why This Guide Exists: Many ElectronX customers have chosen to use QuickFix libraries. Based on common support inquiries and integration challenges, we have compiled this guide to help customers avoid frequently encountered configuration issues. This does not imply that QuickFix is the only or best option for your use case.

Table of Contents

Change History Log

Version	Date	Description
1.0	29 January 2026	Initial document.

SECTION 1: CHOOSING YOUR QUICKFIX IMPLEMENTATION

Supported QuickFix Libraries

ElectronX FIX gateways are compatible with standard QuickFix implementations, including but not limited to:

Language	Library
C++	QuickFIX
C# / .NET	QuickFIX/N
Java	QuickFIX/J
Python	quickfix
Go	quickfix/go

SECTION 2: CONFIGURATION FILE STRUCTURE

QuickFix uses an INI-style configuration file (typically named config.cfg or initiator.cfg for client applications).

Basic Structure

```
None
[DEFAULT]
# Global settings that apply to all sessions

[SESSION]
# Settings specific to each FIX session
# You can have multiple [SESSION] blocks for multiple connections
```

SECTION 3: ELECTRONX DATA DICTIONARIES

ElectronX provides custom FIX data dictionaries that include critical fixes and ElectronX-specific customizations. Using these dictionaries is strongly recommended

and will prevent common integration issues.

Why Use ElectronX Dictionaries:

- Include PossDupFlag support (fixes message replay issues)
- Include ElectronX-specific fields (ManualOrderIndicator, AggressorIndicator, etc.)
- Have correct enum values for ElectronX rejection reasons
- Are tailored to only include messages ElectronX actually uses
- Are tested and validated against ElectronX gateways

Where to Get Them:

ElectronX provides these dictionaries via SFTP (secure file transfer, recommended for automated retrieval) or the Shared Access Documents folder (alternative for manual download):

- FIX50SP2_OE.xml - Order Entry application dictionary
- FIXT11_OE.xml - Order Entry transport dictionary
- FIX50SP2_MD.xml - Market Data application dictionary
- FIXT11_MD.xml - Market Data transport dictionary

How to Use Them:

1. Download the appropriate dictionaries via SFTP or from the Shared Access Documents folder
2. Place them in your QuickFix working directory (or specify full paths in config)
3. Reference them in your configuration file:

```
None
TransportDataDictionary=FIXT11_OE.xml # For Order Entry or Drop Copy
AppDataDictionary=FIX50SP2_OE.xml     # For Order Entry or Drop Copy

# OR for Market Data:
DataDictionary=FIX50SP2_MD.xml         # For Market Data
TransportDataDictionary=FIXT11_MD.xml  # For Market Data
AppDataDictionary=FIX50SP2_MD.xml      # For Market Data
```

IMPORTANT: Use the Order Entry dictionaries (FIX50SP2_OE.xml, FIXT11_OE.xml) for Order Entry AND Drop Copy sessions. Use the Market Data dictionaries

(FIX50SP2_MD.xml, FIXT11_MD.xml) Only for Market Data sessions.

SECTION 4: COMPLETE CONFIGURATION EXAMPLES

Example 1: Order Entry Gateway (Production)

```
None
[DEFAULT]
# Connection settings - these specify how QuickFix behaves as an initiator (client)
ConnectionType=initiator
ReconnectInterval=30  # in seconds

# Directory names for application working directories
FileStorePath=store
FileLogPath=log

# Logging settings - Useful for debugging
ScreenLogShowIncoming=Y
ScreenLogShowOutgoing=Y
FileLogHeartbeats=Y

# ElectronX transport dictionary
TransportDataDictionary=FIXT11_OE.xml
# ElectronX Order Entry dictionary
AppDataDictionary=FIX50SP2_OE.xml

# Message validation
ValidateFieldsOutOfOrder=N
ValidateFieldsHaveValues=Y
ValidateUserDefinedFields=Y
```

```
# CRITICAL: Prevent automatic sequence resets
ResetOnLogon=N
ResetOnLogout=N
ResetOnDisconnect=N

# CRITICAL: Enable message persistence for replay
PersistMessages=Y

# Timeout settings
HeartBtInt=30
LogonTimeout=30
LogoutTimeout=10

# Do not reject inbound messages aggressively during development
RejectInvalidMessage=N

[SESSION]
# Session identification
BeginString=FIXT.1.1
DefaultApplVerID=9
SenderCompID=YOUR_SENDER_ID
TargetCompID=EXI

# CRITICAL: Include SenderSubID for Order Entry
SenderSubID=Johnsmith@yahoo.com [Trader's Email Address]

# Connection details
SocketConnectHost=YOUR_GATEWAY_HOST
SocketConnectPort=YOUR_GATEWAY_PORT
```

```
# Weekly sequence reset schedule (ElectronX resets all sequence
numbers on Sundays at 06:15 UTC)
Please note that this can be configured in multiple ways.
ResetOnTime=Y
ResetOnTimeDay=Sunday
ResetOnTimeTime=06:15:00
ResetOnTimeZone=UTC
```

Example 2: Market Data Gateway

```
None
[DEFAULT]
ConnectionType=initiator
ReconnectInterval=30
FileStorePath=store_md
FileLogPath=log_md

ScreenLogShowIncoming=Y
ScreenLogShowOutgoing=Y
FileLogHeartbeats=Y

StartTime=00:00:00
EndTime=00:00:00
UseDataDictionary=Y
# ElectronX Market Data dictionary
DataDictionary=FIX50SP2_MD.xml
# ElectronX transport dictionary
TransportDataDictionary=FIXT11_MD.xml
# ElectronX Market Data dictionary
AppDataDictionary=FIX50SP2_MD.xml
```

```
ValidateFieldsOutOfOrder=N
ValidateFieldsHaveValues=Y
ValidateUserDefinedFields=Y

# Important: Same as Order Entry
ResetOnLogon=N
ResetOnLogout=N
ResetOnDisconnect=N
PersistMessages=Y

HeartBtInt=30
LogonTimeout=30
LogoutTimeout=10
CheckLatency=N
RejectInvalidMessage=N

[SESSION]
BeginString=FIXT.1.1
DefaultApplVerID=9
SenderCompID=YOUR_SENDER_ID
TargetCompID=EXI

SocketConnectHost=YOUR_MD_GATEWAY_HOST
SocketConnectPort=YOUR_MD_GATEWAY_PORT

ResetOnTime=Y
ResetOnTimeDay=Sunday
ResetOnTimeTime=06:15:00
ResetOnTimeZone=UTC
```

Example 3: Drop Copy Gateway

```
None
[DEFAULT]
ConnectionType=initiator
ReconnectInterval=30
FileStorePath=store_dc
FileLogPath=log_dc

ScreenLogShowIncoming=Y
ScreenLogShowOutgoing=Y
FileLogHeartbeats=Y

StartTime=00:00:00
EndTime=00:00:00
UseDataDictionary=Y
# Use Order Entry dictionary for Drop Copy
DataDictionary=FIX50SP2_OE.xml
# ElectronX transport dictionary
TransportDataDictionary=FIXT11_OE.xml
# Use Order Entry dictionary for Drop Copy
AppDataDictionary=FIX50SP2_OE.xml

ValidateFieldsOutOfOrder=N
ValidateFieldsHaveValues=Y
ValidateUserDefinedFields=Y

# Important: Must persist messages for historical replay
ResetOnLogon=N
ResetOnLogout=N
ResetOnDisconnect=N
PersistMessages=Y
```

```
HeartBtInt=30
LogonTimeout=30
LogoutTimeout=10
CheckLatency=N
RejectInvalidMessage=N

[SESSION]
BeginString=FIXT.1.1
DefaultApplVerID=9
SenderCompID=YOUR_DC_SENDER_ID
TargetCompID=EXI

# NOTE: SenderSubID is NOT required for Drop Copy
# Drop Copy shows all firm activity regardless

SocketConnectHost=YOUR_DC_GATEWAY_HOST
SocketConnectPort=YOUR_DC_GATEWAY_PORT

ResetOnTime=Y
ResetOnTimeDay=Sunday
ResetOnTimeTime=06:15:00
ResetOnTimeZone=UTC
```

SECTION 5: CRITICAL CONFIGURATION SETTINGS EXPLAINED

Sequence Number Management (A Common Issue)

The Problem: Many QuickFix configurations default to resetting sequence numbers on every logon, breaking message replay and causing sessions to lose history.

The Solution:

```
None
ResetOnLogon=N    #Do NOT reset on normal logon
ResetOnLogout=N   #Do NOT reset on logout
ResetOnDisconnect=N  # Do NOT reset on abnormal disconnect
PersistMessages=Y  # ALWAYS save messages to disk for replay
```

Why This Matters / What Happens When Incorrect:

Incorrect Setting	What Happens If Incorrect
ResetOnLogon=Y	Every time you connect, seqnums reset to 1. You lose the ability to replay historical messages since Sunday.
ResetOnDisconnect=Y	Network hiccup means lost history. You will not be able to replay messages sent since Sunday while disconnected.
PersistMessages=N	Messages are not saved to disk. You cannot request a replay even with correct seqnums.

Correct Behavior:

- Session starts Sunday 06:15 UTC with seqnums = 1
- Throughout the week, seqnums increment (2, 3, 4…)
- If you disconnect on Wednesday, your next logon picks up where you left off
- You can request a replay of any message sent during the week
- Only on Sunday 06:15 UTC do seqnums reset to 1

Weekly Session Reset

ElectronX resets sessions every Sunday at 06:15 UTC. Your QuickFix client must do the same:

```
None
ResetOnTime=Y
ResetOnTimeDay=Sunday
ResetOnTimeTime=06:15:00
ResetOnTimeZone=UTC
```

What This Does:

- At 06:15 UTC Sunday, QuickFix will send Logon [A] with ResetSeqNumFlag=Y
- Both sides agree to reset seqnums to 1
- Fresh week begins

If You Don't Set This:

- Your seqnums will diverge from ElectronX's
- You will likely get "`MsgSeqNum too low`" rejections
- Manual intervention will be required to fix

PossDupFlag Configuration (Another Common Issue)

The Problem: When ElectronX replays messages (via ResendRequest), those messages have PossDupFlag=Y (Tag 43). Some QuickFix XML dictionaries don't include this tag at the application layer, causing replayed messages to be silently dropped.

Symptoms:

- You send ResendRequest [2]
- You see messages in FIX logs
- But your application's OnMessage() handler never fires
- QuickFix is filtering them out as duplicates

The Solution:

Option 1: Use ElectronX's Data Dictionaries (STRONGLY RECOMMENDED)

ElectronX provides custom data dictionaries that already include PossDupFlag support and all necessary customizations. These are available via SFTP or the Shared Access Documents folder (see Section 3).

- The best approach is to use ElectronX's dictionaries (FIX50SP2_OE.xml for Order Entry/Drop Copy)
- Prevents this issue entirely
- Includes other important customizations

Option 2: Modify Your FIX50SP2.xml Dictionary

Find the <message name='ExecutionReport' msgtype='8' msgcat='app'> section and ensure the following fields exist:

```xml
<message name='ExecutionReport' msgtype='8' msgcat='app'>
  <!-- ... other fields ... -->
  <field name='PossDupFlag' required='N'/>
```

```
  <field name='OrigSendingTime' required='N'/>
  <!-- ... other fields ... -->
</message>
```

Or globally in the <fields> section:

```XML
<fields>
  <!-- ... other fields ... -->
  <field number='43' name='PossDupFlag' type='BOOLEAN'/>
  <field number='122' name='OrigSendingTime' type='UTCTIMESTAMP'/>
  <!-- ... other fields ... -->
</fields>
```

Note: Using ElectronX's dictionaries is strongly preferred as they include many other important customizations beyond PossDupFlag.

Logging Configuration

Best Practices for Development:

```None
# Show all messages on screen (stdout)
ScreenLogShowIncoming=Y
ScreenLogShowOutgoing=Y

# Log heartbeats (helpful for debugging session issues)
FileLogHeartbeats=Y

# Separate log directories for each session type
FileLogPath=log           # Order Entry
FileLogPath=log_md        # Market Data
FileLogPath=log_dc        # Drop Copy

# Separate message stores for each session
```

```
FileStorePath=store       # Order Entry
FileStorePath=store_md    # Market Data
FileStorePath=store_dc    # Drop Copy
```

For Production: Consider setting ScreenLogShowIncoming=N and ScreenLogShowOutgoing=N to reduce console clutter, but keep file logging enabled.

Validation Settings

```
None
# Use ElectronX data dictionaries
UseDataDictionary=Y
DataDictionary=FIX50SP2_OE.xml              # Or FIX50SP2_MD.xml
for Market Data
TransportDataDictionary=FIXT11_OE.xml       # Or FIXT11_MD.xml for
Market Data
AppDataDictionary=FIX50SP2_OE.xml           # Or FIX50SP2_MD.xml
for Market Data



# Do not be too strict during initial development
ValidateFieldsOutOfOrder=N      # ElectronX sometimes sends fields
out of order
ValidateFieldsHaveValues=Y      # But do check that required fields
have values
ValidateUserDefinedFields=Y     # Validate custom fields
RejectInvalidMessage=N          # Do not reject messages during
development
```

For Production: You may want to set RejectInvalidMessage=Y to catch errors early.

SECTION 6: SSL/TLS CONNECTIVITY

ElectronX FIX gateways require encrypted connections. You have two options:

Option 1: Direct SSL (Recommended)

Modern QuickFix libraries support native SSL/TLS. This is the simplest approach.

Configuration Addition:

```
None
[SESSION]
# ... other settings ...
SocketUseSSL=Y
SocketKeyFile=client.key
SocketCertificateFile=client.crt
SocketCAFile=ca.crt
```

Files You Need (Provided by ElectronX):

- tls.key - Your private key
- cert.crt - Your certificate
- ca.pem - ElectronX's CA certificate

How to Get These: ElectronX provides these files in your "FIX Connectivity Details" package. Place them in the same directory as your config file or specify full paths.

Option 2: STunnel

If your QuickFix version does not support native SSL, use STunnel as a local proxy.

How STunnel Works:

```
None
Your QuickFix App → localhost:10001 (unencrypted)
                          ↓
                      STunnel
                          ↓
        ElectronX Gateway:PORT (encrypted SSL)
```

STunnel Configuration Example (stunnel.demo.conf):

```
None
# stunnel.conf
debug = 7
foreground = yes
```

```
[fix-order-gateway]
client = yes
accept = ###.#.#.#:#####
connect = ##.##.###.###:#####
cert = cert.crt
key = tls.key
CAfile = ca.pem
verify = 2
verifyChain = yes

[fix-market-data-gateway]
client = yes
accept = ###.#.#.#:#####
connect = ##.##.###.###:#####
cert = cert.crt
key = tls.key
CAfile = ca.pem
verify = 2
```

QuickFix Configuration (Point to Localhost):

```
None
[SESSION]
SocketConnectHost=YOUR_SOCKET_CONNECT_HOST
SocketConnectPort=YOUR_SOCKET_CONNECT_PORT
# No SSL settings needed - STunnel handles encryption
```

Starting STunnel:

```
Shell
# Windows
```

```
stunnel.exe stunnel.conf

# Linux/Mac
stunnel stunnel.conf
```

Common STunnel Issues:

Issue	Symptom	Fix
Certificate mismatch	VERIFY ERROR: depth=0, error=certificate verify failed	Ensure ca.pem matches ElectronX's CA
Wrong port	Connection times out	Check that accept port matches QuickFix's SocketConnectPort
Firewall	Connection refused	Open outbound port in firewall

SECTION 7: OTHER COMMON QUICKFIX ISSUES & FIXES

Issue 1: "MsgSeqNum too low" Rejections

Symptom:

```
Received: Logout [5] with Text="MsgSeqNum too low, expecting 45 but received 1"
```

Cause: Your QuickFix sent Logon [A] with MsgSeqNum=1 but ElectronX expected MsgSeqNum=45 (continuing from a previous session).

Fix:

```
# Ensure these are set:
ResetOnLogon=N
```

```
ResetOnDisconnect=N
PersistMessages=Y

# Delete your message store to force fresh start:
# Delete contents of FileStorePath directory (e.g., ./store/*)
# Then reconnect - both sides will reset
```

Prevention: Do not manually delete message store files during normal operations.

Issue 2: ResendRequest Not Working (Messages Not Replaying)

Symptom:

- You send ResendRequest [2]
- You see messages in FIX logs with PossDupFlag=Y
- Your OnMessage() handler does not fire

Cause: QuickFix is filtering out replayed messages because PossDupFlag is not properly supported in your XML dictionary.

Fix: See "PossDupFlag Configuration" section above. Modify your FIX50SP2.xml dictionary accordingly.

Additional C#/.NET Fix:

If you are using QuickFix/N and messages still are not arriving, ensure your FromApp method processes them:

CSharp

```
public void FromApp(Message message, SessionID sessionID)
{
    // QuickFix/N delivers ALL application messages here
    // Including those with PossDupFlag=Y

    if (message is QuickFix.FIX50SP2.ExecutionReport execReport)
    {
        // Check if this is a replayed message
        bool isPossibleDuplicate = false;
        if (message.getHeader().isSetField(Tags.PossDupFlag))
```

```
        {
            isPossibleDuplicate =
message.getHeader().getBoolean(Tags.PossDupFlag);
        }

        // Process the message (regardless of PossDupFlag)
        ProcessExecutionReport(execReport, isPossibleDuplicate);
    }
}
```

Issue 3: Cannot Connect - "Connection Refused"

Symptom:

```
None
Disconnecting: Connection refused
```

Possible Causes & Fixes:

1. **Wrong Host/Port**

```
None
# Double-check these match your FIX Connectivity Details:
SocketConnectHost=YOUR_GATEWAY_HOST
SocketConnectPort=YOUR_GATEWAY_PORT
```

2. **Firewall Blocking Outbound Connection**
    - Corporate firewalls often block non-standard ports
    - Work with your IT team to open the specific port
    - ElectronX uses high-numbered ports (typically 40000+)

3. **Not Using STunnel When Required**
    - If your QuickFix does not support SSL natively
    - Set up STunnel (see Section 5)

Issue 4: Continuous Disconnects Every 30 Seconds

Symptom:

```
None
Logon successful
[29 seconds later]
Disconnecting: Heartbeat timeout
```

Cause: HeartBtInt setting mismatch

Fix:

```
None
HeartBtInt=30
```

```
# Also ensure:
LogonTimeout=30
LogoutTimeout=10
```

Why This Helps:

- ElectronX expects heartbeats every 30 seconds
- If no message (order, heartbeat, etc.) is sent within 30s, send Heartbeat [0]
- QuickFix handles this automatically with correct HeartBtInt

Issue 5: "Invalid Tag Number" Errors

Symptom:

```
None
Reject: Tag not defined for this message type
```

Cause: Your data dictionary does not include ElectronX-specific fields, or you are sending fields that are not allowed.

Fix:

1. **Use ElectronX's Data Dictionaries (see Section 3)**
    - Download from SFTP or via the Shared Access Documents folder
    - FIX50SP2_OE.xml for Order Entry/Drop Copy
    - FIX50SP2_MD.xml for Market Data
    - Replace your default FIX50SP2.xml with the appropriate ElectronX version

2. **Check FIX Specification**
    - Verify the tag is required/optional for that message type
    - Example: SenderSubID (Tag 50) is REQUIRED on Order Entry but NOT on Market Data

SECTION 8: TESTING YOUR CONFIGURATION

Step 1: Validate Config File Syntax

Before running, check for common mistakes:

```shell
# Look for typos
grep -i "reset" initiator.cfg

# Should show:
# ResetOnLogon=N
# ResetOnDisconnect=N
# ResetOnLogout=N
# ResetOnTime=Y
```

Common Typos:

- ResetOnLogin (wrong) vs. ResetOnLogon (correct)
- PersistMessage (wrong) vs. PersistMessages (correct)
- SocketConnectPort=YOUR_SOCKET_CONNECT_PORT with extra spaces

Step 2: Test Connection (Beta Environment)

Always test in ElectronX's Beta environment first:

```
# Beta/Demo credentials
SenderCompID=YOUR_BETA_SENDER_ID
SocketConnectHost=beta.gateway.electronx.com
SocketConnectPort=YOUR_BETA_CONNECT_PORT
```

What Success Looks Like:

```
Logon successful
Sent: Logon [A]
Received: Logon [A]
[Session established]
```

Step 3: Test Message Replay

1. **Send a Test Order:**

```
None
NewOrderSingle [D] → Placed successfully
ExecutionReport [8] received (MsgSeqNum=5)
```

2. **Disconnect (Close Your Application)**

3. **Reconnect**

```
None
Logon [A] sent with ResetSeqNumFlag=N
```

4. **Request Replay:**

```
None
Send ResendRequest [2]:
  BeginSeqNo=5
  EndSeqNo=5
```

5. **Verify Receipt:**

```
None
ExecutionReport [8] received with:
  MsgSeqNum=5
  PossDupFlag=Y
```

SECTION 9: PRODUCTION READINESS CHECKLIST

Before going live:

- [ ] Configuration uses ResetOnLogon=N
- [ ] Configuration uses ResetOnDisconnect=N
- [ ] Configuration uses PersistMessages=Y
- [ ] Using ElectronX data dictionaries (not standard QuickFix dictionaries)
    - FIX50SP2_OE.xml for Order Entry and Drop Copy
    - FIX50SP2_MD.xml for Market Data
    - FIXT11_OE.xml and FIXT11_MD.xml for transport layers
- [ ] Weekly reset scheduled for Sunday 06:15 UTC
- [ ] SSL/TLS connection working (direct or STunnel)

- [ ] Tested ResendRequest message replay successfully
- [ ] Separate log directories for each session type
- [ ] Firewall ports opened for production gateway
- [ ] Production credentials (SenderCompID, host, port) confirmed
- [ ] Data dictionaries include PossDupFlag support
- [ ] Log rotation strategy implemented
- [ ] Monitoring/alerting for disconnects in place

APPENDIX A: TROUBLESHOOTING REFERENCE

Symptom	Possible Cause	Section to Check
Cannot connect	Firewall, wrong host/port, SSL issue	Section 7, Issue 3
MsgSeqNum too low	Reset settings wrong	Section 7, Issue 1
ResendRequest does not work	PossDupFlag dictionary issue	Section 5
Heartbeat timeouts	HeartBtInt mismatch	Section 7, Issue 4
Invalid tag errors	Wrong data dictionary	Section 7, Issue 5
Messages in logs but not in app	PossDupFlag filtering	Section 7, Issue 2
Weekly seqnum problems	No ResetOnTime config	Section 5

APPENDIX B: GETTING HELP

ElectronX Support:

- Email: support@electronx.com
- Include: Your config file (redact credentials), logs, specific error messages, your QuickFix version, and which data dictionaries you're using (ElectronX's or standard QuickFix)

Document Version: 1.0
Last Updated: January 2026
Feedback: support@electronx.com