

# "How to" instructions

- [How to upgrade FEJ to a new version](#)
- [How to configure session schedule that can't be defined via single cron-expression](#)
- [How to configure Echo between FIXEdge Java and Simple Client](#)

## How to upgrade FEJ to a new version

It is very simple to upgrade FEJ. First, download the new version of FEJ and unzip it (see [Quick Start](#)). Then, replace all files and directories in your FEJ home directory, except for the `.conf` directory. All custom user configurations are saved in `.conf`, therefore the upgrade will not affect personal settings.

## How to configure session schedule that can't be defined via single cron-expression

FIXEdge Java 1.8.0 and earlier doesn't support several schedules for the session. So there is a limitation preventing creating a [session schedule](#) requiring multiple cron expressions.

For example, the session should have the next maintenance windows:

- Saturday: 6PM – 7PM ET
- Sunday: 10AM – 11AM ET
- Sunday: 6PM – 7PM ET

To achieve this, the user can configure a weekly session with start at 7 PM Sun and stop at 6 PM Sat. And cover the rest time intervals via external tools and scripts ([FIXEdge JAVA administration shell CLI](#)) that can start the session remotely.



FIXEdge Java Administrative shell requests passing of a password to it.

Before using it as part of the script, the user should at least once connect to the administrative shell and save SSH keys and insert a password manually.

Crontab scripts examples for credentials:

- username - test
- password - test
- remote host - fixedgej.sever.com:2000
- session name - session1

1. It is needed to add two additional commands to the crontab that will **start** the session at 7 PM Sat and 11 AM Sun:

```
sshpass -p test ssh -t test@fixedgej.sever.com -p 2000 'session start session1'
```

2. It is needed to add two additional commands to the crontab that will **stop** the session at 11 AM Sun and 6 PM Sun

```
sshpass -p test ssh -t test@fixedgej.sever.com -p 2000 'session start session1'
```



`sshpass` tool is installed on the administrative host.

Optionally for the Windows remote host, you can use the `plink` tool (which is part of the PuTTY package):

```
plink.exe fixedgej.sever.com -P 2000 -l test -pw test -batch -t session start session1
```

## How to configure Echo between FIXEdge Java and Simple Client

Echo configuration between FIXEdge Java and Simple Client is performed in the following way:

1. Install `FIXEdge Java`.
2. To create the `TestEchoAcceptor` session, create a new file named `fixedgej-x.x\conf\session\ls_fix_TestEchoAcceptor.properties` Configure session parameters there:

### s\_fix\_TestEchoAcceptor.properties

```
sessionType=acceptor
senderCompID=FIXEdgeJ
targetCompID=FIXECHOCLIENT
fixVersion=FIX.4.4
startOnload = true
```

3. Configure Business Logic rules in the `fixedgej-x.x\conf\rules.groovy` file. This will route all messages from the `TestEchoAcceptor` session back to the `TestEchoAcceptor` session.

### rules.groovy

```
RoutingContext rc = routingContext as RoutingContext;
[
    messageRule("FIXEDGE -> FIXECHOCLIENT")
        .sourceCondition({ source -> source.id == "TestEchoAcceptor" })
        .action({
            ctx ->
                def destination = rc.getDestinationById("TestEchoAcceptor")
                destination.send(ctx.getMessage())
                ctx.exit()
        })
        .build(),
    getRejectionRule(routingContext)
]
```

The session-id value in the rule `source.id == "TestEchoAcceptor"` is the same as the ID in the `s_fix_TestEchoAcceptor.properties` session configuration file.

4. Run FIXEdge Java.  
For example, via the run script:
  - **Windows:** Go to the FIXEdge Java installation dir `fixedgej-x.x\bat\` and run the `runConsole.bat` script.
  - **Linux:** Go to the FIXEdge Java installation dir `fixedgej-x.x\bin` and run the `runConsole.sh` script.
5. Open [FIX Client Simulator \(FCS\)](#).  
Create a new session as **Initiator**:

```
Session type: Initiator
SenderCompID = FIXECHOCLIENT
TargetCompID = FIXEdgeJ
FIX Version = 4.4
Remote Port = 8911 (server.port parameter in FEJ fixedge.properties -> fixedgej-x.x\conf\fixedge.
properties )
Remote Host = localhost (or IP address/hostname of the FIXEdge Java server)
```

Create Session

Configuration  
 TestEchoAcceptorFEJ save delete

Session type  Initiator  Acceptor

SenderCompID  
 FIXECHOCLIENT

TargetCompID  
 FIXEdgeJ

Qualifier

FIX version  
 4.4

Remote host  
 127.0.0.1

Remote port  
 8911

Heartbeat interval  
 10

Encryption  
 NONE

Templates  
 none

InSeqNum  
 -1

OutSeqNum  
 -1

Use custom Logon message

Persistent storage type

Reset SeqNums

Use Extended properties  Show session messages

Extended Properties

SenderSubID

TargetSubID

SenderLocationID  
 TLocationId

TargetLocationID  
 SLocationId

UserName  
 user

Password  
 \*\*\*\*\*

SourceIPAddress

IntradayLogoutToleranceMode

ForceSeqNumReset

ForcedReconnect

EnableMessageRejecting

IgnoreSeqNumTooLowAtLogon

Use SSL

SSL

SSL Protocols

SSL v2  SSL v3

TLS v1.0  TLS v1.1  TLS v1.2

Ciphers

Certificate

Private Key

Validate Peer Certificate

CA Certificates file

OK Cancel

 The default port for session acceptors is 8911 and specified as the **server.port** parameter in the **fixedge.properties** file.

NOTE: **SenderCompID** on the FCS side should have the same value as **TargetCompID** on the FIXEdge side. **TargetCompID** on the FCS side should be the same as **SenderCompID** on FIXEdge side.

- Send the message from FIX Client Simulator (the Pipes character is used to represent the SOH character).

#### New Order - Echo test.msg

```
8=FIX.4.4|9=156|35=D|49=FIXECHOCLIENT|56=FIXEdgeJ|34=2|52=20210625-22:16:38.867
|11=Order#3|21=1|100=1|55=TESTSMBL|54=1|60=20131124-04:11:46.768|38=20000|40=2|44=34.7|10=026|
```

The file contains SOH symbols: [NewOrderEchoTestMessage.txt](#)

- The Echo message from FIXEdge is received.

```
8=FIX.4.4|9=156|35=D|34=2|49=FIXEdgeJ|56=FIXECHOCLIENT|52=20210625-22:16:38.878
|11=Order#3|21=1|100=1|55=TESTSMBL|54=1|60=20131124-04:11:46.768|38=20000|40=2|44=34.7|10=028|
```