# How to automatically send a message to a session right after getting it established



## Overview

This article describes the way how to automatically send a particular message to a session right after it was established by means of FIXEdge Business Rules and Scripts.

In the examples below it is shown how to generate and send News (MsgType = B) or Market Data Request (MsgType = V) messages after the successful logon handshake.

Common workflow assumed is the following:

1. A session connects to FIXEdge or FIXEdge acts as an initiator.
2. A counterparty accepts the connection and sends confirming Logon (35=A) message
3. Right after the session is established FIXEdge sends a message (35=B or 35=V) to the session.

In general, FIXEdge rises internal notification event on Business Level (as 35=C message) after a session changed its state.

In order to accomplish the task,

1. an event of successful session establishment is handled with Business rules,
2. then the required message is generated with scripts,
3. lastly, the message is sent to the session.

***Following the instructions below the best approach to generate any desired message may be chosen.***

## How to send News message to the established session

News (MsgType = B) message is generated with parseMessage(<msg>) function and sent to every session with an '**Established**' state.

### Handle session state change event on BL and run a script

Below is the sample of BL_Config which can be used for the session state change handling and calling **FIXEdge1/conf/processEstablishedSession.js** sc ript:

**BL_Config.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<FIXEdge>
        <BusinessLayer>

                <Rule Description="Send News to every connected session">
                        <Source>
                                <FixSession SenderCompID="fake" TargetCompID="fake" />
                        </Source>
                        <Condition>
                                <!-- Catching of internal FIX message with session status -->
                                <EqualField Field="35" Value="C"/>
                                <MatchField Field="147" Value=".*(AttemptToConnect|Established|Terminated
correctly|Non-gracefully terminated)"/>
                        </Condition>
                        <Action>
                                <!-- Execute script for this event -->
                                <Script Language="JavaScript" FileName="FIXEdge1/conf/processEstablishedSession.
js"/>
                        </Action>
                </Rule>

                <DefaultRule>
                        <Action>
                                <DoNothing/>
                        </Action>
                </DefaultRule>

        </BusinessLayer>
</FIXEdge>
```

## Generate a message from raw text and send it

**FIXEdge1/conf/processEstablishedSession.js**

```
// Send News (35=B) Message to established session.

message="8=FIX.4.49=12335=B49=FIXCLIENT56=FIXEDGE34=252=20170518-08:52:04.492148=Welcome to TEST Server.
10=213"; // The message should be with SOH symbols, find it attached below

status = getStringField(147); // Get information about session event.
//147=[xxxx] (Sender):(Target) (state)
var regexp = /.*\s+([\w-]+):([\w-]+)\s+(.*)/g;

match = regexp.exec(status);
if (match != null)
{
    target = match[1];
    sender = match[2];
    state = match[3];
    print("processEstablishedSession.js session '"+sender+"'-'"+target+"' state='"+state+"'.") //for debuging
    if (state == 'Established')
    {
        // Create News message
        parseMessage(message);

        // Example of message fields update
        setNumField(33, 2); // Update two line of text in group 33.
        group = getGroup(33);
        setStringField(group, 0, 58, "Welcome " + sender + "!");
        time = getCurrentDate(DATETIMEUtc);
        setStringField(group, 1, 58, "Connection time: " + dateToString(time));

        // Send message back to Established session
        send(target, sender); // Swap sender and target before sending back
    }
}
```

A message for parseMessage function should contain SOH as separator

The script with SOH symbols: processEstablishedSession.js

As a result, the following message will be sent to the session:

```
8=FIX.4.4|9=152|35=B|49=FIXCLIENT|56=FIXEDGE|34=2|52=20170518-08:52:04.492|148=Welcome to TEST Server.
|33=2|58=Welcome FIXCLIENT!|58=Connection time: 20171018-18:48:16|10=027|
```

# How to send Market Data Request to the established session

Market Data Request (MsgType = V)  message is generated with transform(<targetprotocol>,<targetmessagetype>) function and sent only to a specific session with an '**Established**' state.

## Handle session state change event on BL and run a script

Below is the sample of BL_Config which can be used only for the session (**123-456**) state change handling and calling **FIXEdge1/conf /MarketDataSubscription.js** script:

**BL_Config.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<FIXEdge>
    <BusinessLayer>

        <Rule Description="Rule to send Market Data Request right after the session 123:456 is established">
            <Source>
                <FixSession SenderCompID="fake" TargetCompID="fake" />
            </Source>
            <Condition>
                <!-- Catching of internal FIX message about established session with SenderCompID = 123,
TargetCompID = 456 -->
                <EqualField Field="35" Value="C" />
                <MatchMessage Value=".*147=\[NOTE\] 123:456 Established.*"/>
            </Condition>
            <Action>
                <!-- Call script for this event -->
                <Script Language="JavaScript" FileName="FIXEdge1/conf/MarketDataSubscription.js"/>
            </Action>
        </Rule>

        <DefaultRule>
            <Action>
                <DoNothing/>
            </Action>
        </DefaultRule>

    </BusinessLayer>
</FIXEdge>
```

## Prepare a message and send it

**FIXEdge1/conf/MarketDataSubscription.js**

```
// ********* Generation of Market Data Request **************

msgType = getStringField(35);

switch (msgType)
{
    case "C":
    {
        sesStat =  getStringField (147);
        n= sesStat.indexOf (":");
        sessionSender = sesStat.substring(7, n);
        m = sesStat.indexOf (" ", 7);
        sessionTarget = sesStat.substring(n+1, m);

        nLinesOfText     = getNumField(33);
        groupLinesOfText = getGroup(33);
        if ((sessionSender+":"+sessionTarget == getStringField(groupLinesOfText, 2, 58)) && ("Established" ==
getStringField(groupLinesOfText, 3, 58)))
        {
            // Session is established >> Market Data Request creation

            transform("FIX.4.4", "V");

            rand = Math.random();
            reqID="ReqID" + sessionSender.substring(3)+ "_" + rand;   // Assign random RequestID

            setStringField(262, reqID);
            setStringField(263, '1'); // SubscriptionRequestType =  Snapshot + Updates (Subscribe)
            setStringField(264, '1'); // MarketDepth = Top of book
            setNumField(267,2);
            groupNoPartyIDs = getGroup(267);
            setNumField(groupNoPartyIDs,0,269,0); // MDEntryType = Bid
            setNumField(groupNoPartyIDs,1,269,1); // MDEntryType = Offer
            setNumField(146,1);
            groupNoRelatedSym = getGroup(146);
            setStringField(groupNoRelatedSym, 0, 55, 'EUR/USD'); // Symbol to subscribe to is EUR/USD

            send(sessionSender, sessionTarget); // Send message to the session generated the event

        }
    }
    break;
}
```

As a result, the following message will be sent to the session:

```
8=FIX.4.4|9=157|35=V|49=123|56=456|34=2|142=SLocationId|143=TLocationId|52=20171019-12:25:31.650|262=ReqID_0.
6840461604994303|263=1|264=1|267=2|269=0|269=1|146=1|55=EUR/USD|10=010|
```