How to save session state to History

Overview

For each change of session state, FIXEdge creates a new **internal** FIX message (35=C | 49=fake | 56=fake) and sends this message into the business layer to notify about the changes.

The tag 147 in this message is automatically filled by the pattern: [<Category>] <SenderCompID:TargetCompID> <Actual Session state>.

In order to track session state it is required to perform the following steps:

1. parse tag 147 using JavaScript;
2. save session state to history.

Message parsing

In order to get session state, it is recommended to use the script below. It allows to parse tag 147 in the 35=C message which contains session name and actual session state.

As the result of the script execution we get three tags: 56 - TargetCompID, 49 - SenderCompID, 147 - State.

prepareSessionState.js

```
status = getStringField(147);
var regexp = /.*\s+(\S+):(\S+)\s+(.*)/g;


match = regexp.exec(status);
if (match != null)
{
    target = match[1];
    sender = match[2];
    state = match[3];
    //print("[DEBUG] matched target = " + target + " sender = " + sender + " state = " + state);
        setStringField(56, target);
        setStringField(49, sender);
    setStringField(147, state);
}
```

ODBC history

Database configuration

In order to save session state to the database it is required to perform the following steps::

1. Create corresponding table to store session statuses;
2. Create stored procedure which allows to write data into the table.

Please find an example below. In this case PostgreSQL database is used.

SessionStatus.sql

```
create table session_actions(
    id                                      serial PRIMARY KEY,
    _SenderCompId                varchar(64) NOT NULL,
    _TargetCompId                varchar(64) NOT NULL,
    _OrigTime                        varchar(21) NOT NULL,
    _SessionState                varchar(20) NOT NULL
);


CREATE OR REPLACE FUNCTION insertSessionStatus
(
    _SenderCompId                varchar(64),
    _TargetCompId                varchar(64),
    _OrigTime                        varchar(21),
    _SessionState                varchar(20)
)
RETURNS void
AS $$
BEGIN
    insert into session_actions(SenderCompId, TargetCompId, OrigTime, SessionState) values ( _SenderCompId,
_TargetCompId, _OrigTime, _SessionState);
END;
$$
LANGUAGE plpgsql;
```

FIXEdge configuration

In order to save session state into the database, BL-Rules should be configured on the FIXEdge side:

1. Rule for history definition: history name, database parameters, fields definitions.
2. Rule for saving to history: get message 35 = C, get session status using JavaScript, save to history (database).

BL_Config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
<!DOCTYPE FIXEdge SYSTEM "BusinessLayer.dtd">
-->
<FIXEdge>
 <BusinessLayer>

  <!-- DB Table with Session Status -->
  <History
            Name="SessionStatus"
            StorageType="ODBC"
            TableName="FIXEdge.dbo.SessionStatus"
            ConnectionString="DSN=FIXEdgeDB;UID=user;PWD=password;"
            UseStoredProcForInsert="true"
            StoredProcName="insertSessionStatus"
            >
   <KeyField ColumnName="_SenderCompId">49</KeyField> <!-- SenderCompId -->
   <KeyField ColumnName="_TargetCompId">56</KeyField> <!-- TargetCompId -->
   <KeyField ColumnName="_OrigTime">42</KeyField> <!-- OrigTime -->
   <KeyField ColumnName="_SessionState">147</KeyField> <!-- Session's State -->
  </History>

  <!-- Save Events in 35=C messages notifing about session state to DB -->
  <Rule>
   <Source>
    <FixSession SenderCompID=".*" TargetCompID=".*" />
   </Source>
   <Condition>
    <MatchField Field="35" Value="C"/>
    <MatchField Field="147" Value=".*:.*"/> <!-- Assume sender and target in 147 field is divided by ':'. The
rest of the messages are redundant -->

    <Exclusion>
     <MatchMessage Value=".*147=.* AttemptToConnect.*"/>
    </Exclusion>
   </Condition>
   <Action>
    <Script Language="JavaScript" FileName="FIXEdge1/conf/prepareSessionState.js"/>
    <SaveToHistory Name="SessionStatus"/>
   </Action>
  </Rule>



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

 </BusinessLayer>
</FIXEdge>
```

File history

FIXEdge configuration

In order to save session state into the history file, BL-Rules should be configured on the FIXEdge side:

1. Rule for history definition: history name, history file parameters, fields definitions.
2. Rule for saving to history: get message 35 = C, get session status using JavaScript, save to history (file).

BL_Config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
<!DOCTYPE FIXEdge SYSTEM "BusinessLayer.dtd">
-->
<FIXEdge>
 <BusinessLayer>
  <!-- File history with Session Status -->
  <History
  Name="SessionStatus"
  StorageType="File"
  WorkingDirectory="FIXEdge1/conf/"
  StorageFileName="SessionStatus">
  <KeyFields>49, 56, 42, 147</KeyFields>
 </History>

  <!-- Save Events in 35=C messages notifing about session state to the file -->
  <Rule>
   <Source>
    <FixSession SenderCompID=".*" TargetCompID=".*" />
   </Source>
   <Condition>
    <MatchField Field="35" Value="C"/>
    <MatchField Field="147" Value=".*:.*"/> <!-- Assume sender and target in 147 field is divided by ':'. The
rest of the messages are redundant -->
    <Exclusion>
     <MatchMessage Value=".*147=.* AttemptToConnect.*"/>
    </Exclusion>
   </Condition>
   <Action>
   <Script Language="JavaScript" FileName="FIXEdge1/conf/prepareSessionState.js"/>
    <SaveToHistory Name="SessionStatus"/>
   </Action>
  </Rule>


  <DefaultRule>
   <Action>
    <DoNothing/>
   </Action>
  </DefaultRule>
 </BusinessLayer>
</FIXEdge>
```

As a result, FIXEdge creates history file with session statuses:

SessionStatus.history

```
<Record valid='Y' size='/' data='SENDER1 TARGET1 20160920-13:41:32 Established'/>

<Record valid='Y' size='/' data='SENDER2 TARGET2 20160920-13:41:32 Established'/>

<Record valid='Y' size='8' data='SENDER1 TARGET1 20160920-21:00:01 Terminated correctly'/>

<Record valid='Y' size='8' data='SENDER2 TARGET2 20160920-21:00:01 Terminated correctly'/>
```