Solution 1: Configure Event Fired rules

1. **Create the Event Fired rules for Fraud Check.**

   a) Navigate to **Event Fired** rule set.

   b) Create the following rules based on best practices structure.

   - Right-click on **Event Fired** rule set and select **New Rule**.
   - Enter **ABEF2000 – Fraud Check** for the rule name.
   - Move the new rule under **ABEF1000 – Vendor Recommendat**ion
     - Rule condition should check for the destination.

```
1    USES:
2
3    CONDITION (messageContext  :  entity.MessageContext):
7    return messageContext.DestID == 30
8    ACTION (messageContext  :  entity.MessageContext, actions : gw.rules.Action):
13   // execute child rules
14   END
```

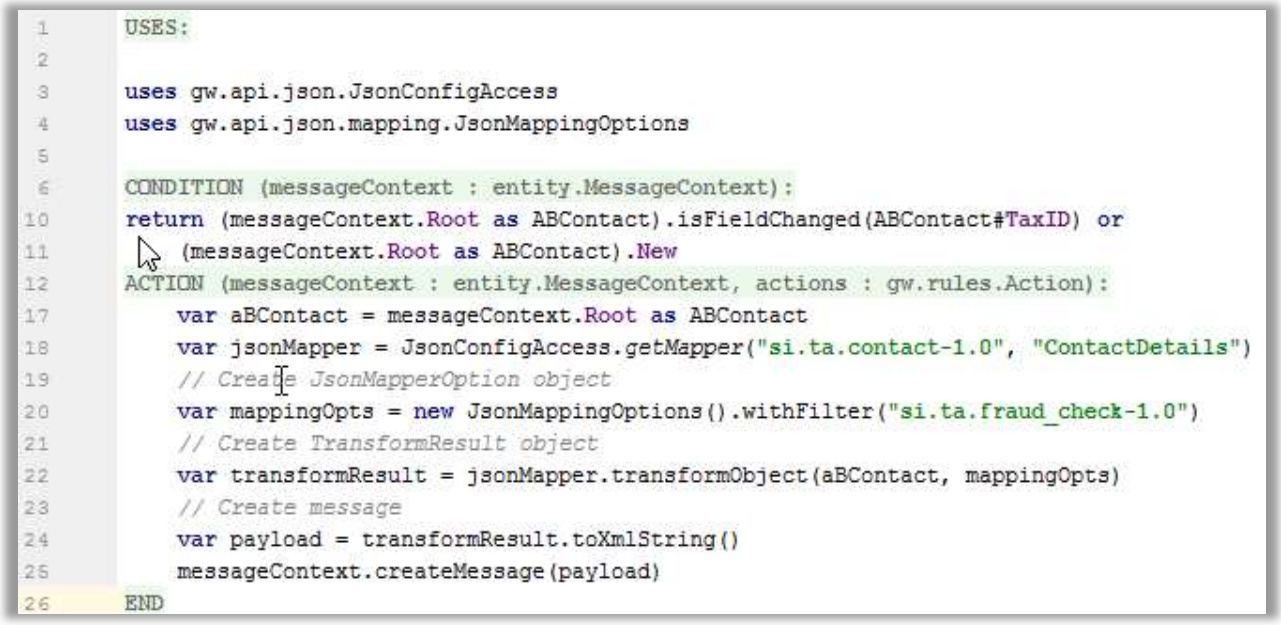
   - Create **ABEF2100 – ABContact** child rule.
     - Rule condition should check for entity type.

```
1   USES:
2
3   CONDITION (messageContext  :  entity.MessageContext):
7   return messageContext.Root typeis ABContact
8   ACTION (messageContext  :  entity.MessageContext, actions : gw.rules.Action):
13  // execute child rules
14  END
```

- Create **ABEF2110 – Added or Changed** child rule.
  - Rule condition should check for event type.

```
1   USES:
2
3   CONDITION (messageContext    :    entity.MessageContext):
7   return messageContext.EventName == "ABContactAdded" or
8          messageContext.EventName == "ABContactChanged"
9   ACTION (messageContext    :    entity.MessageContext, actions : gw.rules.Action):
14  // execute child rules
15  END
```

- Create **ABEF2111 – Field Changed** child rule.
  - Rule condition should check if tax ID field has changed or if new contact.
  - Rule condition should create a message whose payload is in XML format.



```
USES:

uses gw.api.json.JsonConfigAccess
uses gw.api.json.mapping.JsonMappingOptions

CONDITION (messageContext : entity.MessageContext):
return (messageContext.Root as ABContact).isFieldChanged(ABContact#TaxID) or
    (messageContext.Root as ABContact).New
ACTION (messageContext : entity.MessageContext, actions : gw.rules.Action):
    var aBContact = messageContext.Root as ABContact
    var jsonMapper = JsonConfigAccess.getMapper("si.ta.contact-1.0", "ContactDetails")
    // Create JsonMapperOption object
    var mappingOpts = new JsonMappingOptions().withFilter("si.ta.fraud_check-1.0")
    // Create TransformResult object
    var transformResult = jsonMapper.transformObject(aBContact, mappingOpts)
    // Create message
    var payload = transformResult.toXmlString()
    messageContext.createMessage(payload)
END
```

```
var aBContact = messageContext.Root as ABContact
    var jsonMapper = JsonConfigAccess.getMapper("si.ta.contact-1.0", "ContactDetails")
    // Create JsonMapperOption object
    var mappingOpts = new JsonMappingOptions().withFilter("si.ta.fraud_check-1.0")
    // Create TransformResult object
    var transformResult = jsonMapper.transformObject(aBContact, mappingOpts)
    // Create message
    var payload = transformResult.toXmlString()
    messageContext.createMessage(payload)
```

2. **Deploy code changes.**

    a) From the **Studio** menu, **Restart** the server.

3. **Perform verification steps.**