TEMENOS
The Banking Software Company

Event Designer (EE)

Introduction

The Integration Framework architecture is a flexible framework that allows an event which is emitted from any T24 transaction. Event Designer (EE) is the IDE used to define the event and the flow (that is; message schema) which is emitted from T24. This document explains

- How to install Event Designer
- How to create Event Designer project
- How to publish the Event Designer project into T24

Assumptions

This document describes the concepts that require the basic knowledge of Eclipse 3.6 and above.

System Requirements

1. R13 GA or later
2. TAFC R13 GA
3. Eclipse 3.6 or later

1. R14 or later
2. TAFC
3. Eclipse Juno or later

1. 201405
2. TAFC 201405
3. Eclipse Juno or later

Installing Event Designer

Copy the Event Designer Plug-in into 'dropins' folder of Eclipse:

The name of the component is in the following format: 'com.temenos.integration.plugin-<version>.jar'

Environment Configuration

Start the TAFC Agent to establish a connectivity between Event Designer and TEMENOS T24. While starting the TAFC Agent, SSL certificates can also be used. Thus the connectivity between Event Designer and jremote is secured using certificates. Start the TAFC Agent from the command line/shell through the following executable:

tafc_agent -p [PORT NUMBER]

Example: tafc_agent -p 33333

(or)

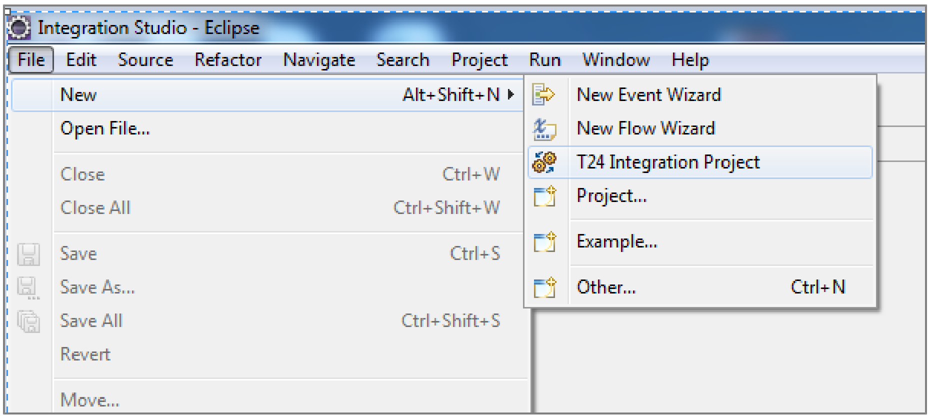
tafc_agent -p [PORT NUMBER] -c [CERTIFICATE PATH] -k [KEY PATH]

Example: tafc_agent -p 33333 -c c:\openssl\bin\keys\cert.cer -k c:\openssl\bin\keys\key.pem

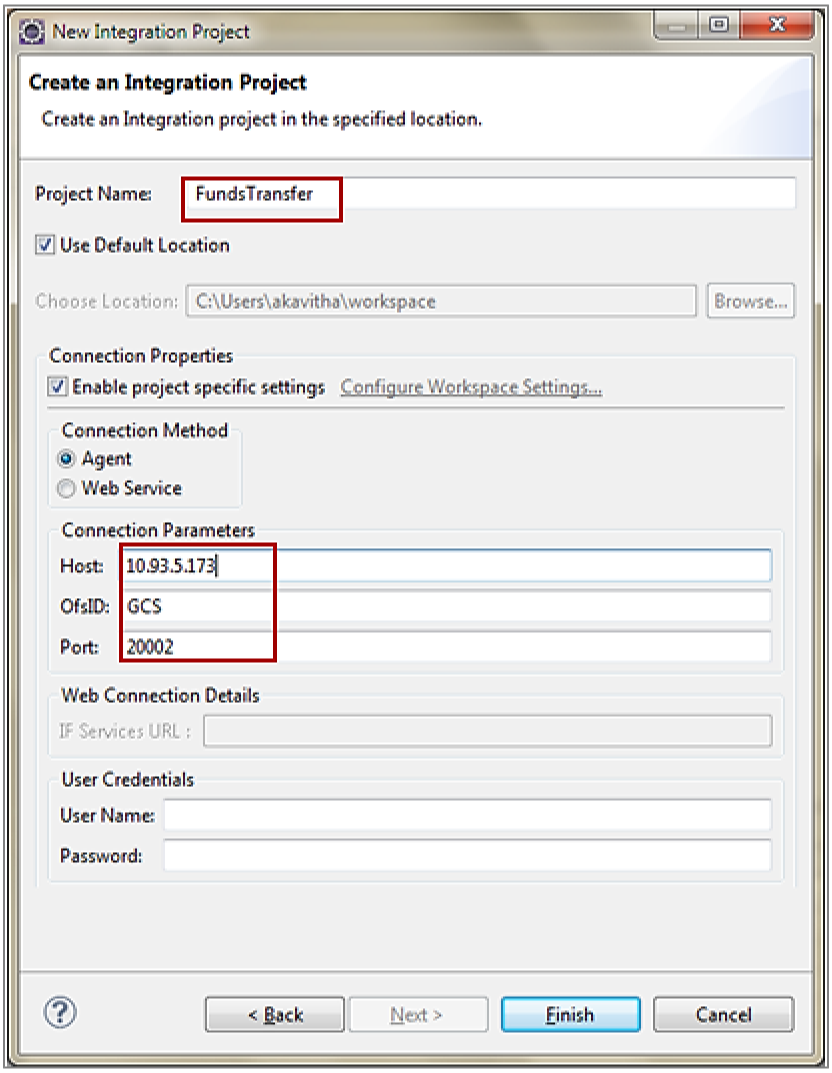
Using Event Designer

Create Event Designer project

- Event Designer project is created using the New Project wizard, once the Event Designer plugin is installed.



- Specify the name of the project, the connection parameters for connecting to T24 and the user credentials, in the new Integration Project window.


New Integration Project

Create an Integration Project

Create an Integration project in the specified location.

Project Name: FundsTransfer

☑ Use Default Location

Choose Location: C:\Users\akavitha\workspace

Connection Properties

☑ Enable project specific settings

Connection Method

● Agent

○ Web Service

Connection Parameters

Host: 10.93.5.173

OfsID: GCS

Port: 20002

Web Connection Details

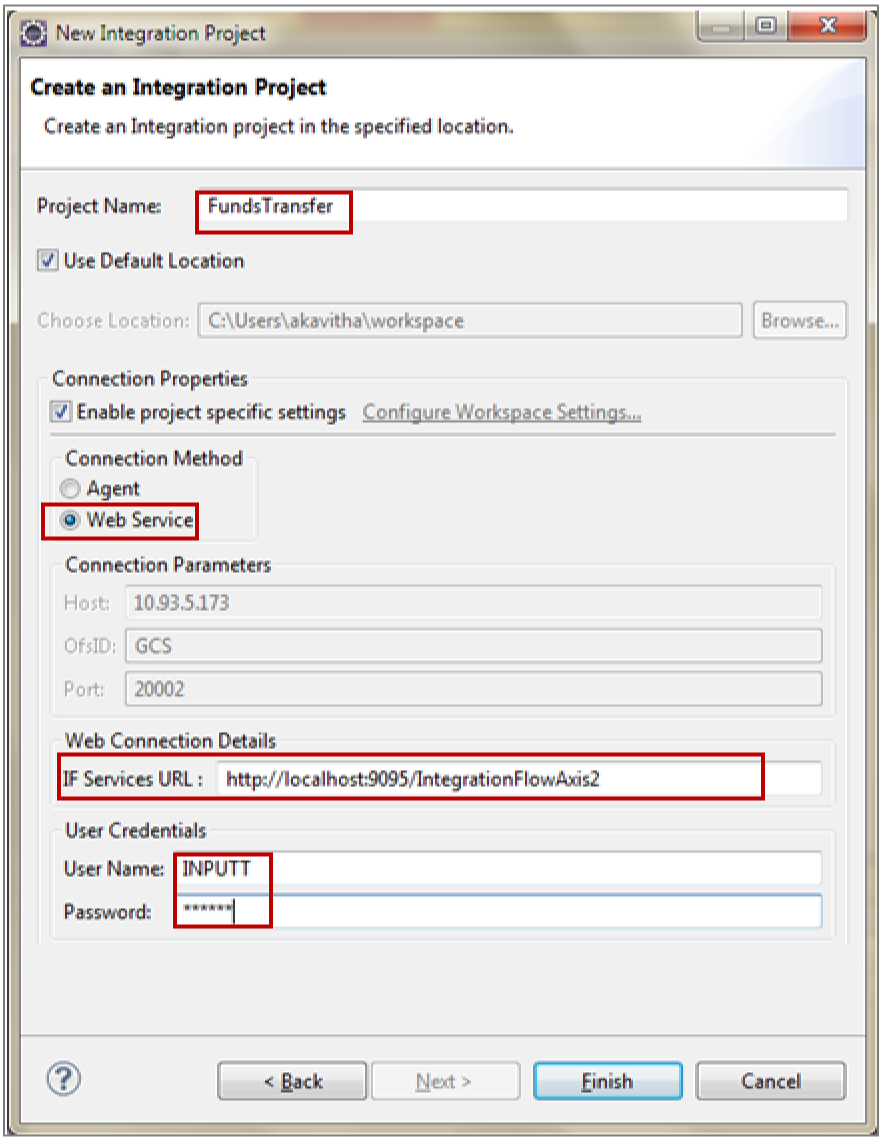
IF Services URL :

User Credentials

User Name:

Password:

The Web Service Connection Details option is used for connecting to T24 using the Web Service interface and the runtime is TAFC or TAFJ.



New Integration Project

Create an Integration Project

Create an Integration project in the specified location.

Project Name: FundsTransfer

☑ Use Default Location

Choose Location: C:\Users\akavitha\workspace

Connection Properties

☑ Enable project specific settings

Connection Method

○ Agent

● Web Service

Connection Parameters

Host: 10.93.5.173

OfsID: GCS

Port: 20002

Web Connection Details

IF Services URL : http://localhost:9095/IntegrationFlowAxis2

User Credentials

User Name: INPUTT

Password: ******

To use this option the following components must be packed as a web archive and then must be deployed in an application server like jBoss, weblogic or websphere.

- IntegrationLandscapeService
- IntegrationFlowService
- Catalog Service for R13 and above or service repository for lower releases

The IF Service URL must be provided along with the user credentials for connecting to T24. The format of IF Service is'http://<Host Name>/<jar Name>'.

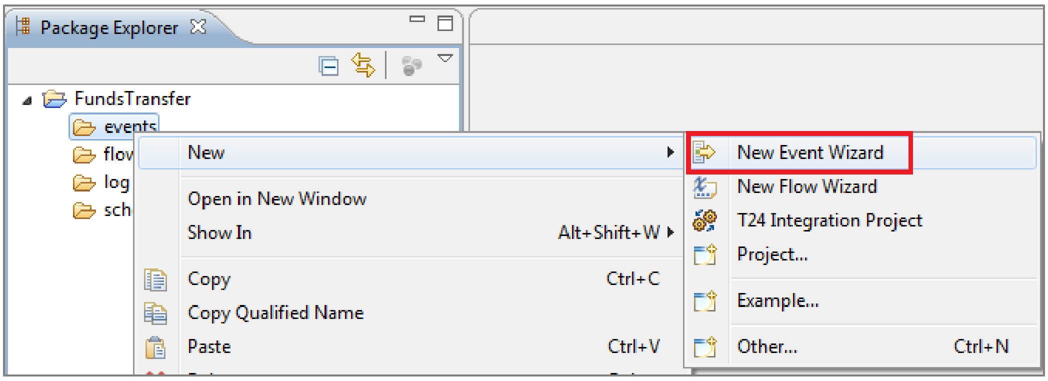
Note: Refer 'Deploying Component Service User Guide' for deploying a component in a J2EE application server.

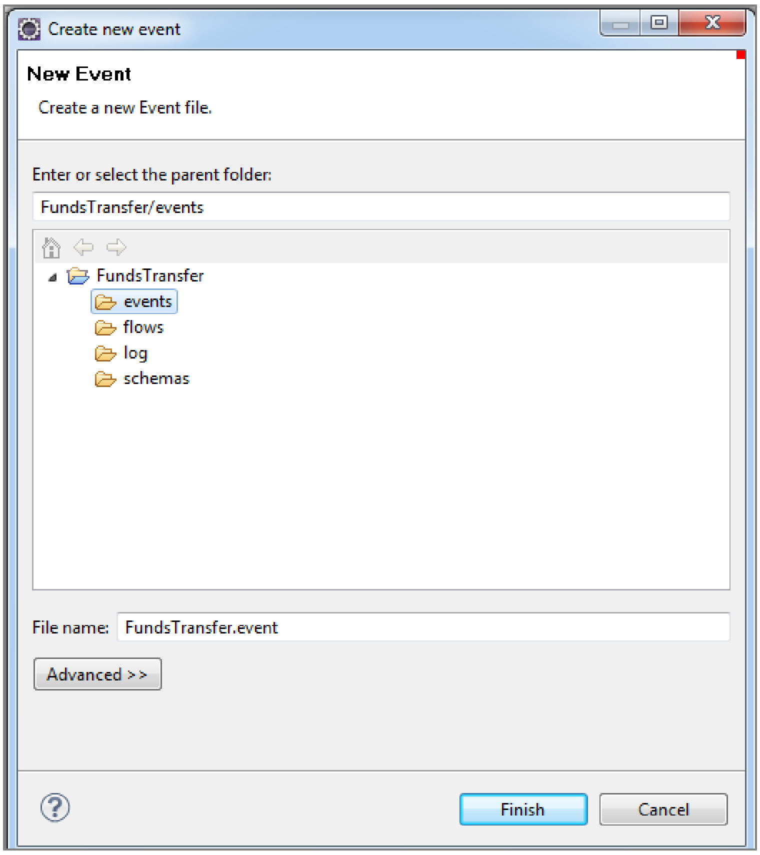
Note: The configuration file (integrationLandscapeServiceContext and integrationFlowServiceContext) under the .aar file has the configurations for both TAFC and TAFJ runtime. Based on the runtime used, the user comments/un-comments the required lines from this file.

- On creation of the project, create a new event using the new Event Wizard option. Create the new Event Wizard under the 'events' folder as shown in the below image.

Create a New Event

- To create a new event, right click the 'events' folder under the Event Designer project and specify the name of the event. Ensure that the event name has .event extension.




Create new event

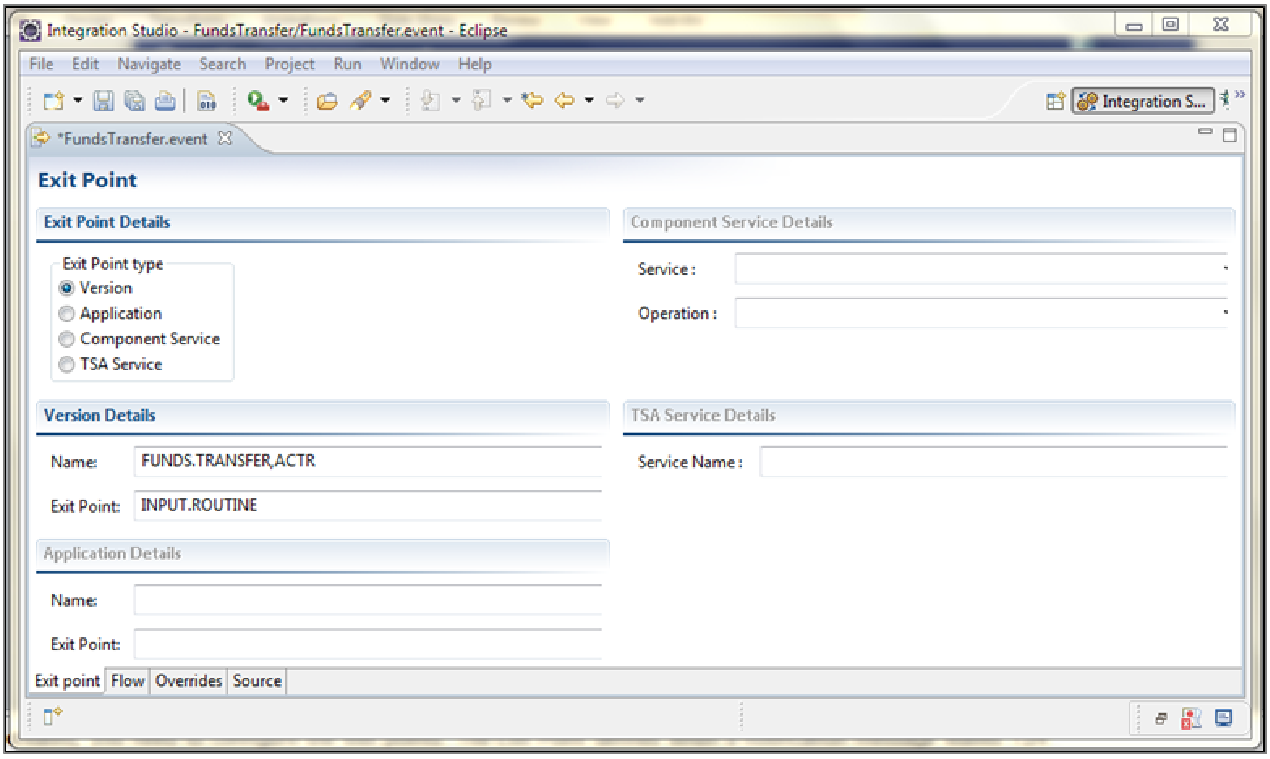
New Event

Create a new Event file.

Enter or select the parent folder: FundsTransfer/events

File name: FundsTransfer.event

- With a T24 Connection available, Integration Designer retrieves the T24 data like; Version, Application, Exit Points, Application Fields etc, and populates the relevant drop down boxes.
- Define the exit point in the new event. Exit point can be an input or authorisation routine of a Version, an Application, or Component Service operation. A new option TSA.SERVICE is also added to define events for CoB or Service. Currently, the exit point is to inform at which stage the event will be triggered.
- On selecting the exit point type, the respective section gets enabled, such that the exit point and the application/version name can be specified.



Integration Studio - FundsTransfer/FundsTransfer.event - Eclipse

*FundsTransfer.event

Exit Point

Exit Point Details

Exit Point type
- ● Version
- ○ Application
- ○ Component Service
- ○ TSA Service

Component Service Details

Service :

Operation :

Version Details

Name: FUNDS.TRANSFER,ACTR

Exit Point: INPUT.ROUTINE

TSA Service Details

Service Name :

Application Details

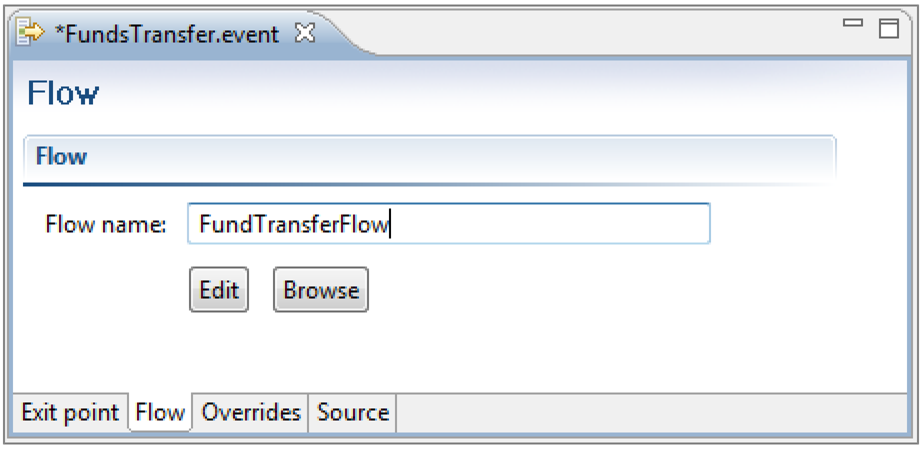
Name:

Exit Point:

Note: The component service option allows the user to define the event for component T24. Refer 'How to define an Event for Component Service' guide for more details. The TSA Service option allows the user to define the event for COB or for a service. Refer 'How to define an Event for TSA Service' guide for more details.

Defining the flow

- Click on the 'Flow' tab, to define the fields which holds the data that has to be sent out of T24. In the 'Flow name' field, specify the name for the flow and click 'Edit'. Click on 'Browse' to attach an existing flow to the event.



*FundsTransfer.event

Flow

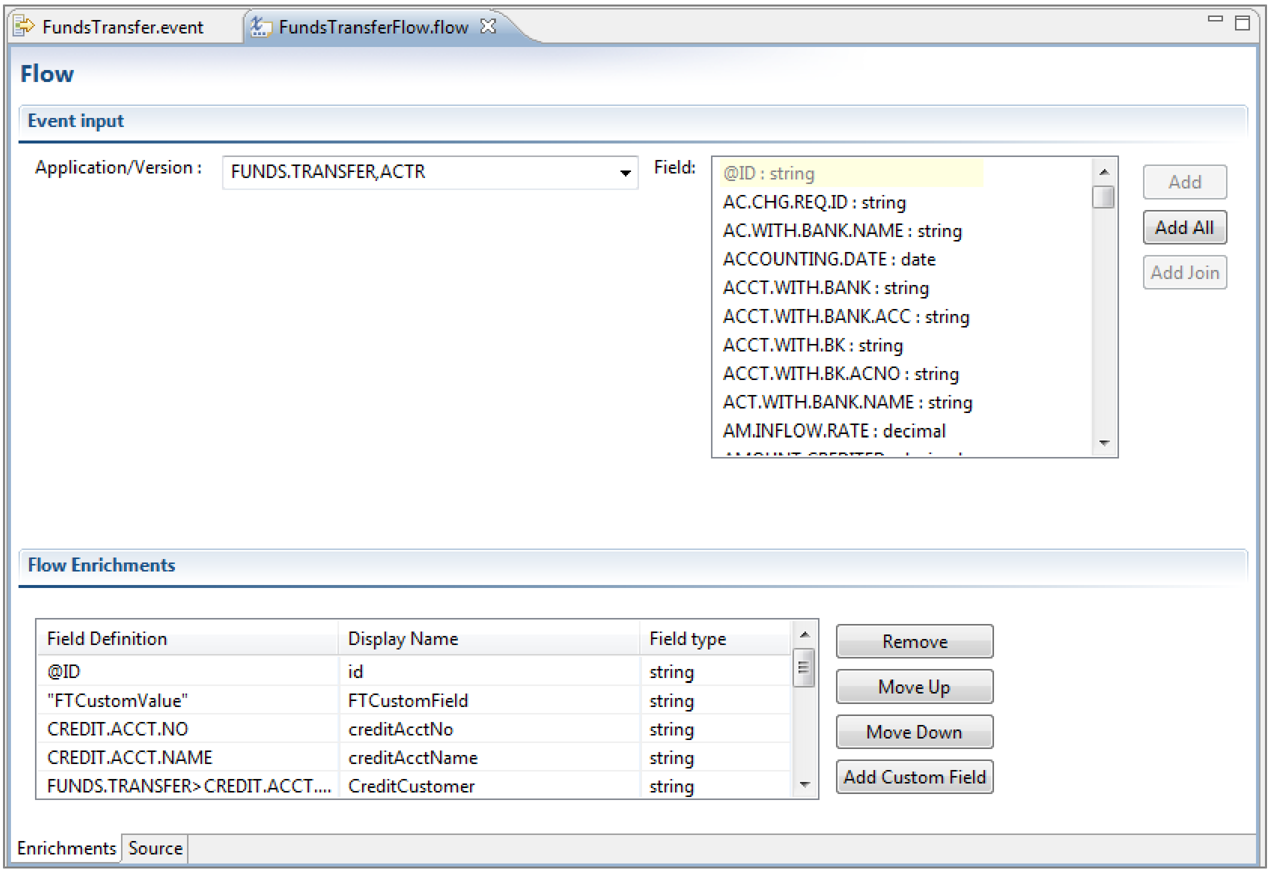
Flow

Flow name: FundTransferFlow

Note: Attach only valid flows to the integrator. For example, attaching a FundsTransfer flow to a Teller version will result in a publish error, as the fields in FundsTransfer are not be available in Teller.

- Based on the Application/Version selected in the exit point, a list of all the fields in STANDARD.SELECTION is displayed in 'Field' box. Required fields can be selected from the list. The selected fields appears in the Table section of the screen. The XML representation of the Table can be viewed in the 'Source' tab. This XML file is an expert view of the project file.

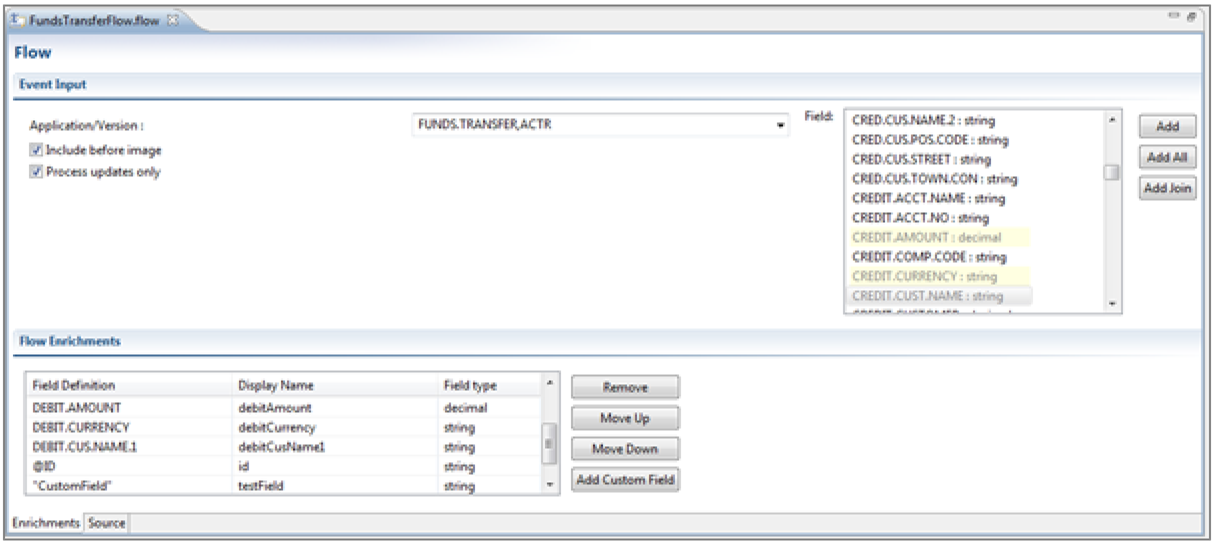
- A custom field can be used to define the event delivery options.
- Click 'Add', to add the selected field to the flow.
- Click on ‘Add All’, to add all the fields to the flow. Press the ‘Ctrl’ key and select the required fields, to select only the required fields. Press the ‘Shift’ key and click the last/first field, to select a continuous list of fields.
- Click on 'Remove', to remove any added field.
- Click 'Move Up' and 'Move Down', to modify the order of the added fields.



Application/Version : FUNDS.TRANSFER,ACTR

@ID : string
AC.CHG.REQ.ID : string
AC.WITH.BANK.NAME : string
ACCOUNTING.DATE : date
ACCT.WITH.BANK : string
ACCT.WITH.BANK.ACC : string
ACCT.WITH.BK : string
ACCT.WITH.BK.ACNO : string
ACT.WITH.BANK.NAME : string
AM.INFLOW.RATE : decimal

Field Definition	Display Name	Field type
@ID	id	string
"FTCustomValue"	FTCustomField	string
CREDIT.ACCT.NO	creditAcctNo	string
CREDIT.ACCT.NAME	creditAcctName	string
FUNDS.TRANSFER>CREDIT.ACCT....	CreditCustomer	string



Application/Version : FUNDS.TRANSFER,ACTR

☑ Include before image
☑ Process updates only

CRED.CUS.NAME.2 : string
CRED.CUS.POS.CODE : string
CRED.CUS.STREET : string
CRED.CUS.TOWN.CON : string
CREDIT.ACCT.NAME : string
CREDIT.ACCT.NO : string
CREDIT.AMOUNT : decimal
CREDIT.COMP.CODE : string
CREDIT.CURRENCY : string
CREDIT.CUST.NAME : string

Field Definition	Display Name	Field type
DEBIT.AMOUNT	debitAmount	decimal
DEBIT.CURRENCY	debitCurrency	string
DEBIT.CUS.NAME.1	debitCusName1	string
@ID	id	string
"CustomField"	testField	string

The option 'Include before image' adds the existing transaction data to the generated event.

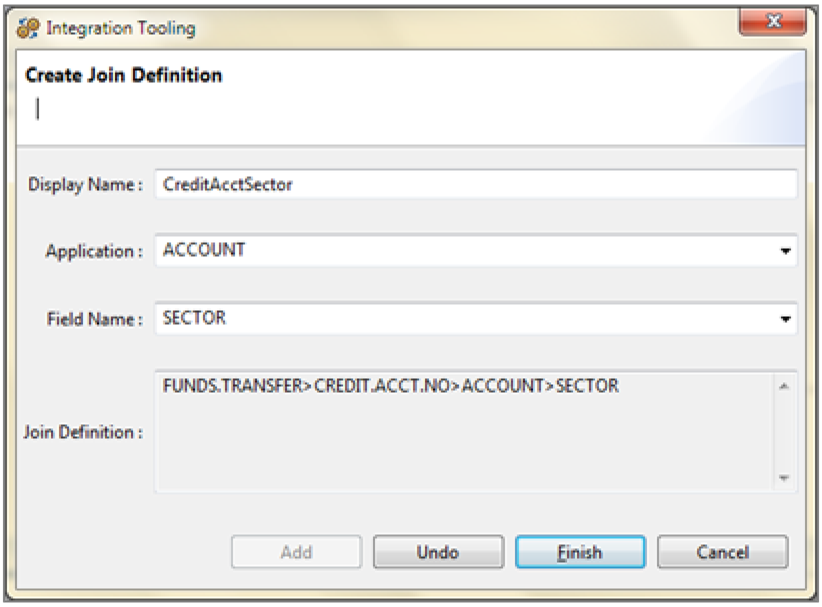
The option 'Process updates only' includes only the changed field value for the current transaction to the event and ignores all other fields while generating the event.

The Display Name column in the Flow Enrichments is an editable column. Once a T24 field is added to the schema, this column allows to specify a different name to the field only in the flow to make the names more meaningful. This does not change the field name in the underlying application

Join Field:

It is possible to add join field in the flow. A join field is a field that is used to retrieve the related information. To define a join field:

- Select the field through which the join has to be defined. (Here CREDIT.ACCT.NO)
- Select Join.
- A popup window as shown below appears.
- Select the application from 'Application' drop down box from which the join retrieves the data (Here ACCOUNT)
- Select the field from the 'Field Name' drop down box. This is the field from where the join field gets the value. (Here SECTOR)
- Click 'Add' to add to the join definition
- Repeat step 4-6 till the join field is completed in the 'Join Definition' field.
- Click 'Finish' to complete the join definition.



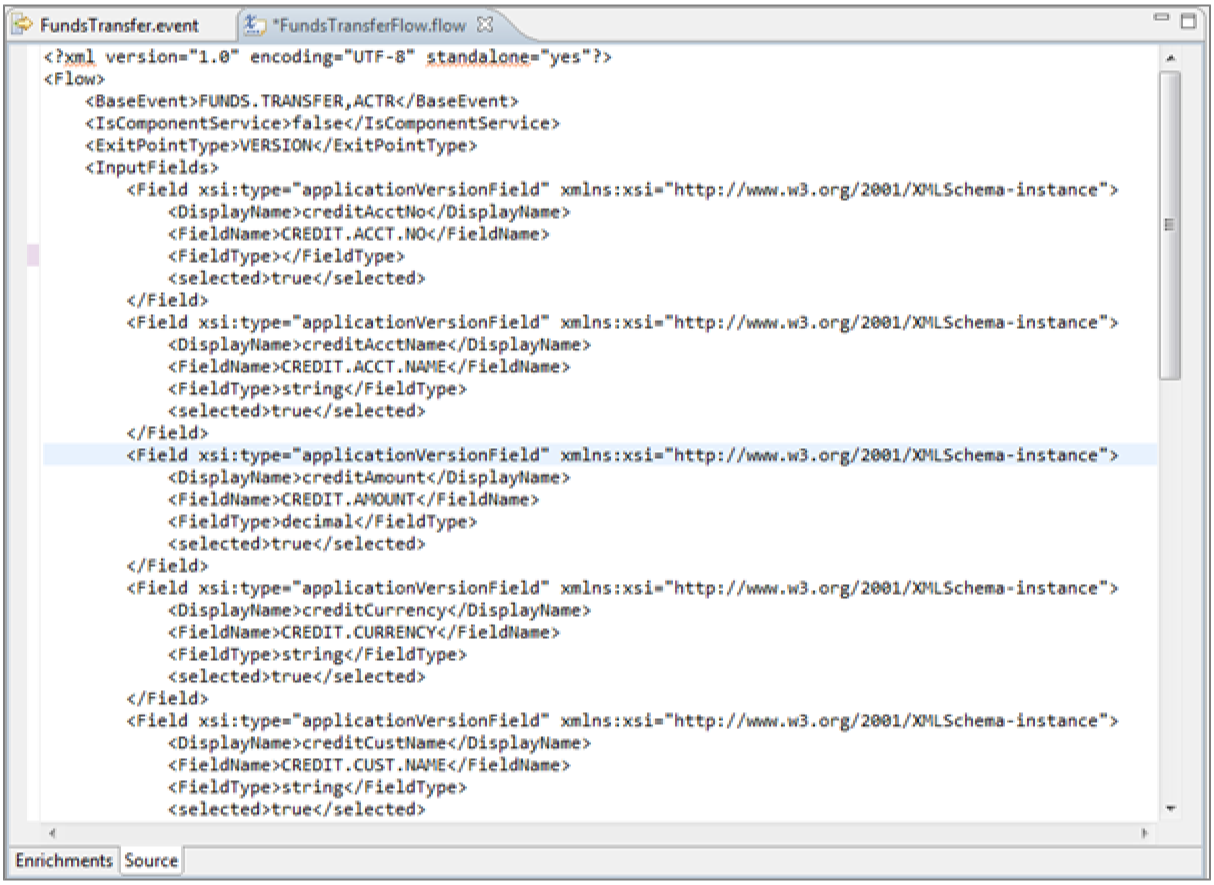
Display Name : CreditAcctSector

Application : ACCOUNT

Field Name : SECTOR

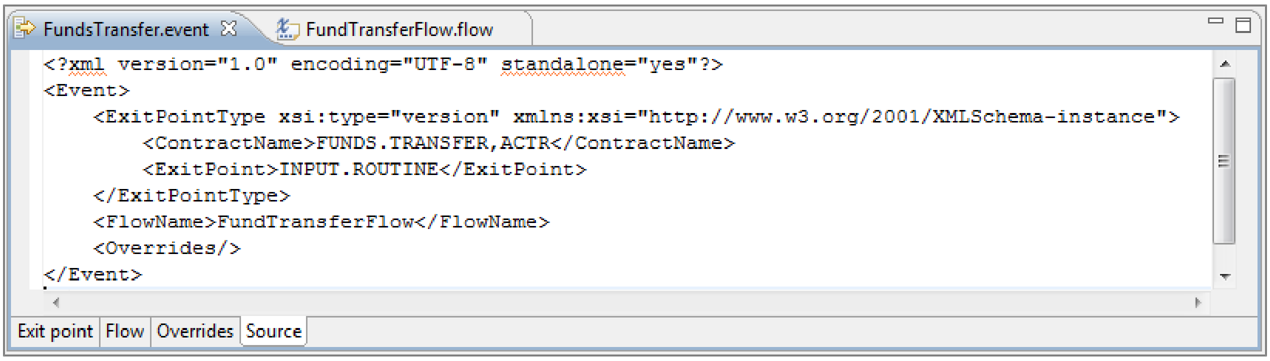
Join Definition : FUNDS.TRANSFER>CREDIT.ACCT.NO>ACCOUNT>SECTOR

- The Source tab provides the XML based representation of the Flow as shown below.



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Flow>
    <BaseEvent>FUNDS.TRANSFER,ACTR</BaseEvent>
    <IsComponentService>false</IsComponentService>
    <ExitPointType>VERSION</ExitPointType>
    <InputFields>
        <Field xsi:type="applicationVersionField" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <DisplayName>creditAcctNo</DisplayName>
            <FieldName>CREDIT.ACCT.NO</FieldName>
            <FieldType></FieldType>
            <selected>true</selected>
        </Field>
        <Field xsi:type="applicationVersionField" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <DisplayName>creditAcctName</DisplayName>
            <FieldName>CREDIT.ACCT.NAME</FieldName>
            <FieldType>string</FieldType>
            <selected>true</selected>
        </Field>
        <Field xsi:type="applicationVersionField" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <DisplayName>creditAmount</DisplayName>
            <FieldName>CREDIT.AMOUNT</FieldName>
            <FieldType>decimal</FieldType>
            <selected>true</selected>
        </Field>
        <Field xsi:type="applicationVersionField" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <DisplayName>creditCurrency</DisplayName>
            <FieldName>CREDIT.CURRENCY</FieldName>
            <FieldType>string</FieldType>
            <selected>true</selected>
        </Field>
        <Field xsi:type="applicationVersionField" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <DisplayName>creditCustName</DisplayName>
            <FieldName>CREDIT.CUST.NAME</FieldName>
            <FieldType>string</FieldType>
            <selected>true</selected>
```

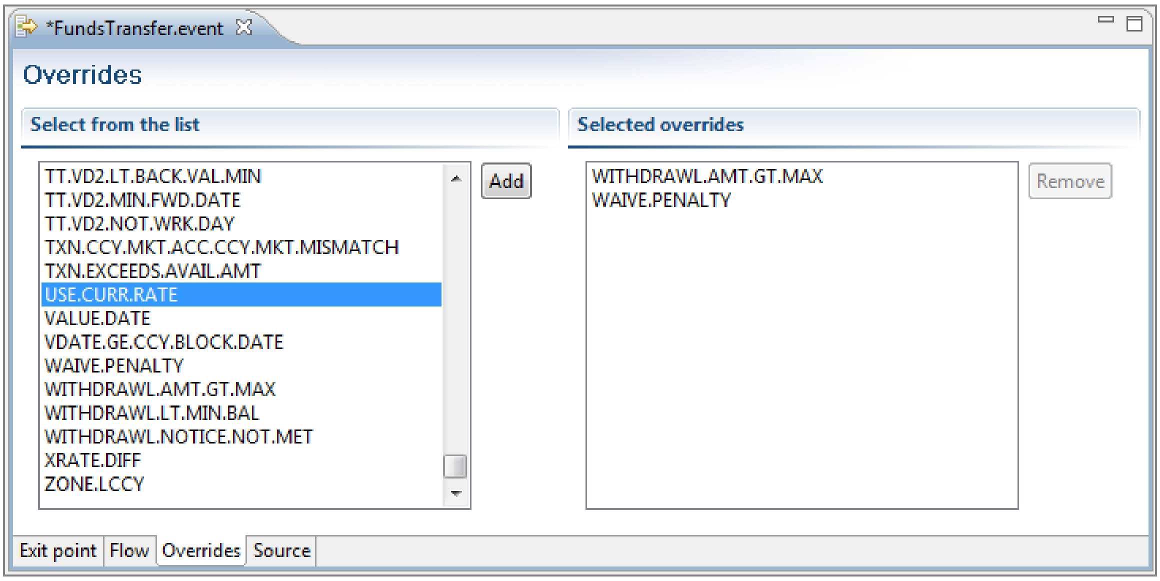
- This is the XML source of the event. This could be viewed from the Source Tab under the event



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Event>
    <ExitPointType xsi:type="version" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <ContractName>FUNDS.TRANSFER,ACTR</ContractName>
        <ExitPoint>INPUT.ROUTINE</ExitPoint>
    </ExitPointType>
    <FlowName>FundTransferFlow</FlowName>
    <Overrides/>
</Event>
```

- Overrides can be specified using the Overrides tab. This helps in implementing DISPO.PROCESSING, such that a particular user can authorise the record. Only Overrides with DISPO.PROCESSING set to FORCE are available. Overrides tab is enabled only for INPUT.ROUTINE exit point, as override is inappropriate during Authorized stage.

Note: Override option is not supported for Component Service option.

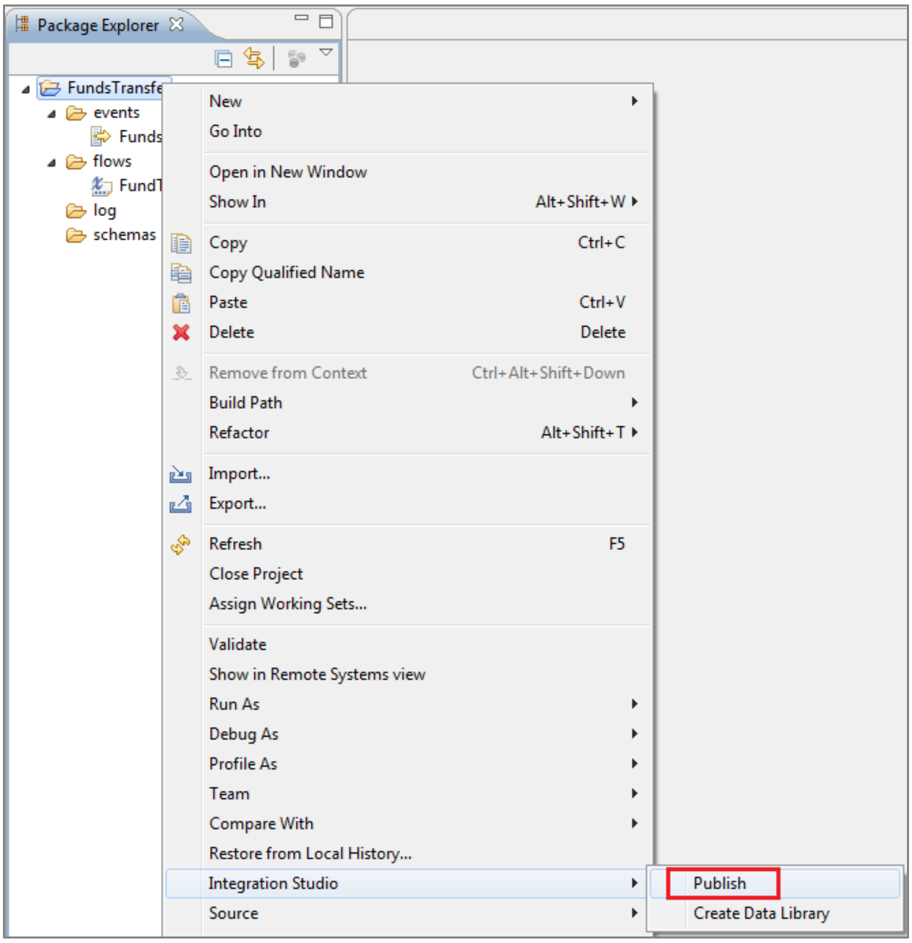


TT.VD2.LT.BACK.VAL.MIN
TT.VD2.MIN.FWD.DATE
TT.VD2.NOT.WRK.DAY
TXN.CCY.MKT.ACC.CCY.MKT.MISMATCH
TXN.EXCEEDS.AVAIL.AMT
USE.CURR.RATE
VALUE.DATE
VDATE.GE.CCY.BLOCK.DATE
WAIVE.PENALTY
WITHDRAWL.AMT.GT.MAX
WITHDRAWL.LT.MIN.BAL
WITHDRAWL.NOTICE.NOT.MET
XRATE.DIFF
ZONE.LCCY

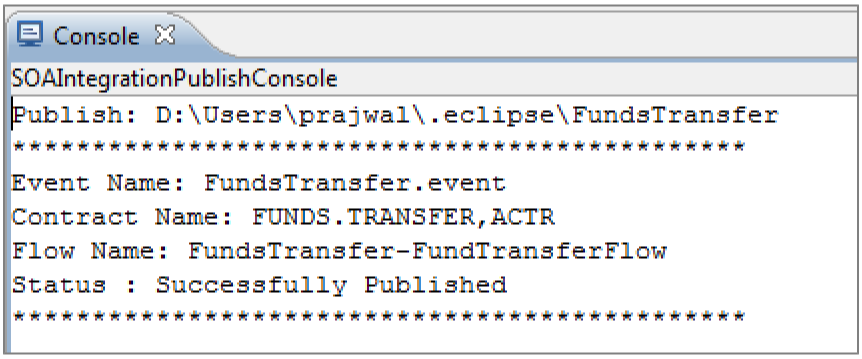
WITHDRAWL.AMT.GT.MAX
WAIVE.PENALTY

Publish the Event

- On defining the event and the flow for a event, the project can be published to T24 using the 'Publish' option in Event Designer. Publish creates the necessary records in the respective T24 applications.



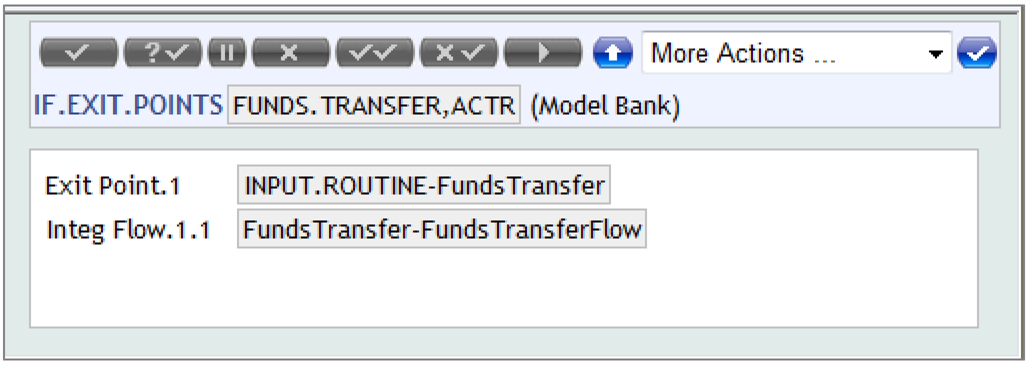
- Console displays the status of publish.



```
Publish: D:\Users\prajwal\.eclipse\FundsTransfer
**********************************************
Event Name: FundsTransfer.event
Contract Name: FUNDS.TRANSFER,ACTR
Flow Name: FundsTransfer-FundTransferFlow
Status : Successfully Published
**********************************************
```

T24 Records

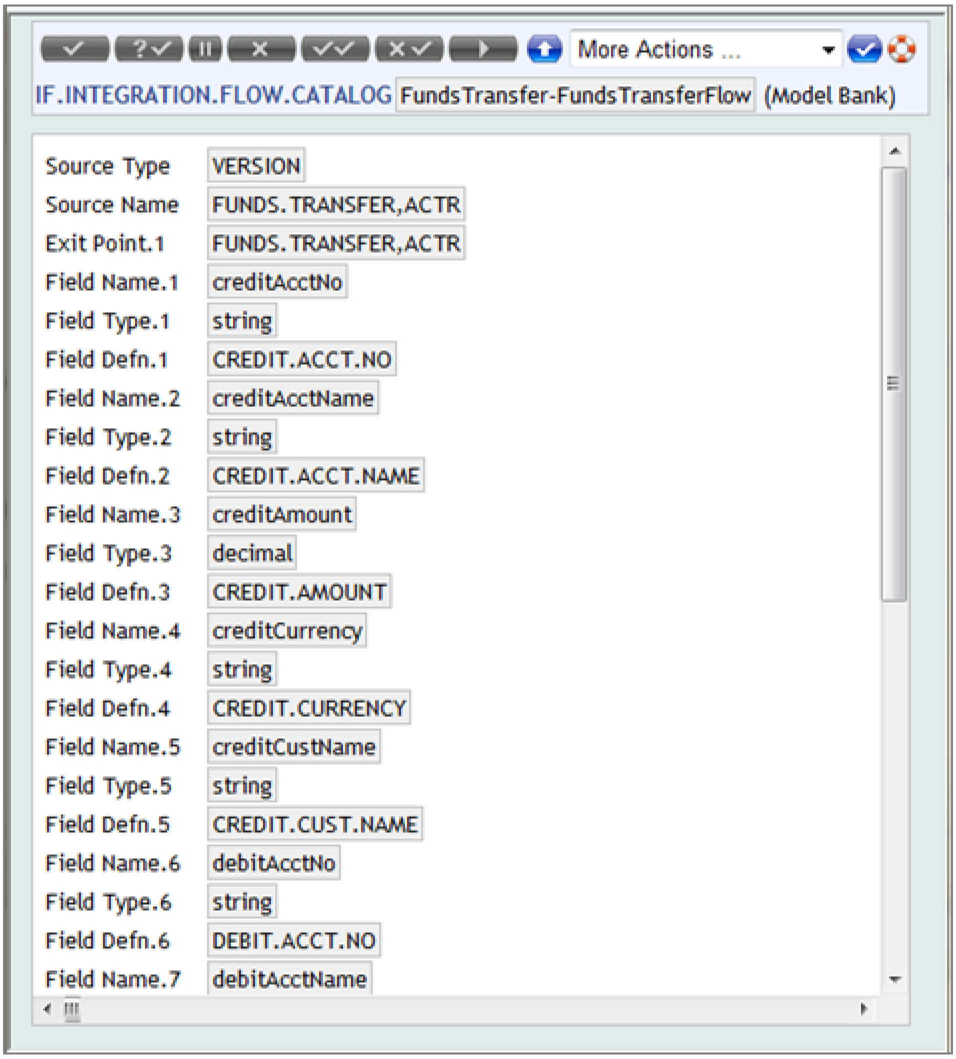
- The following applications are updated, when the Event Designer project is published successfully to T24.
  - IF.EXIT.POINTS: which holds records that binds the notification exit points with their integration flows.



IF.EXIT.POINTS FUNDS.TRANSFER,ACTR (Model Bank)

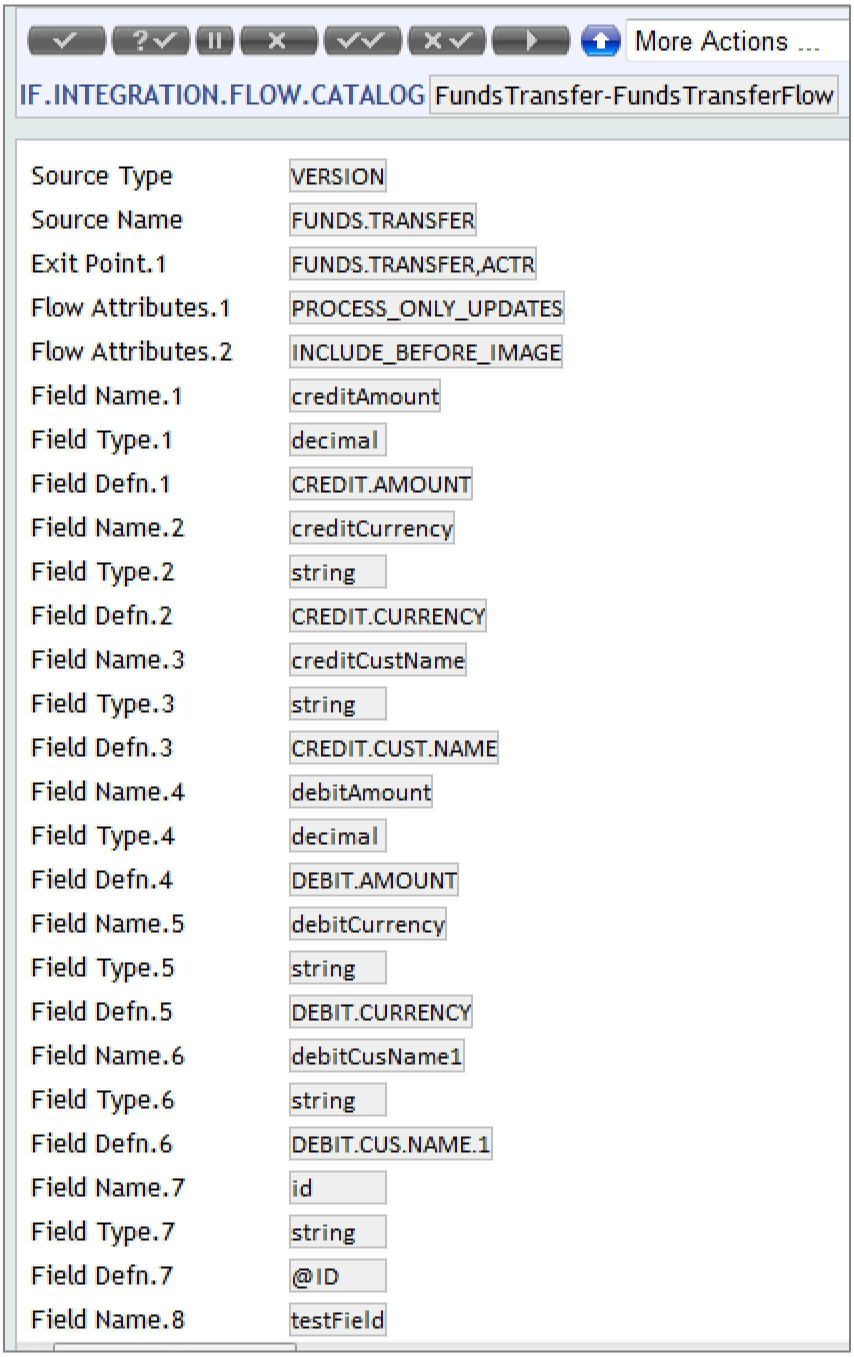
Exit Point.1	INPUT.ROUTINE-FundsTransfer
Integ Flow.1.1	FundsTransfer-FundsTransferFlow

  - IF.INTEGRATION.FLOW.CATALOG: This is a catalog for the Integration Flow records. Integration Flows are created from SOA Studio to be stored in this table at design time.
  - These flows are attached to Exit Points where the notifications are sent out from T24. The flow schema thus created will be stored in the FLOW.SCHEMA field of IF.INTEGRATION.FLOW.CATALOG table



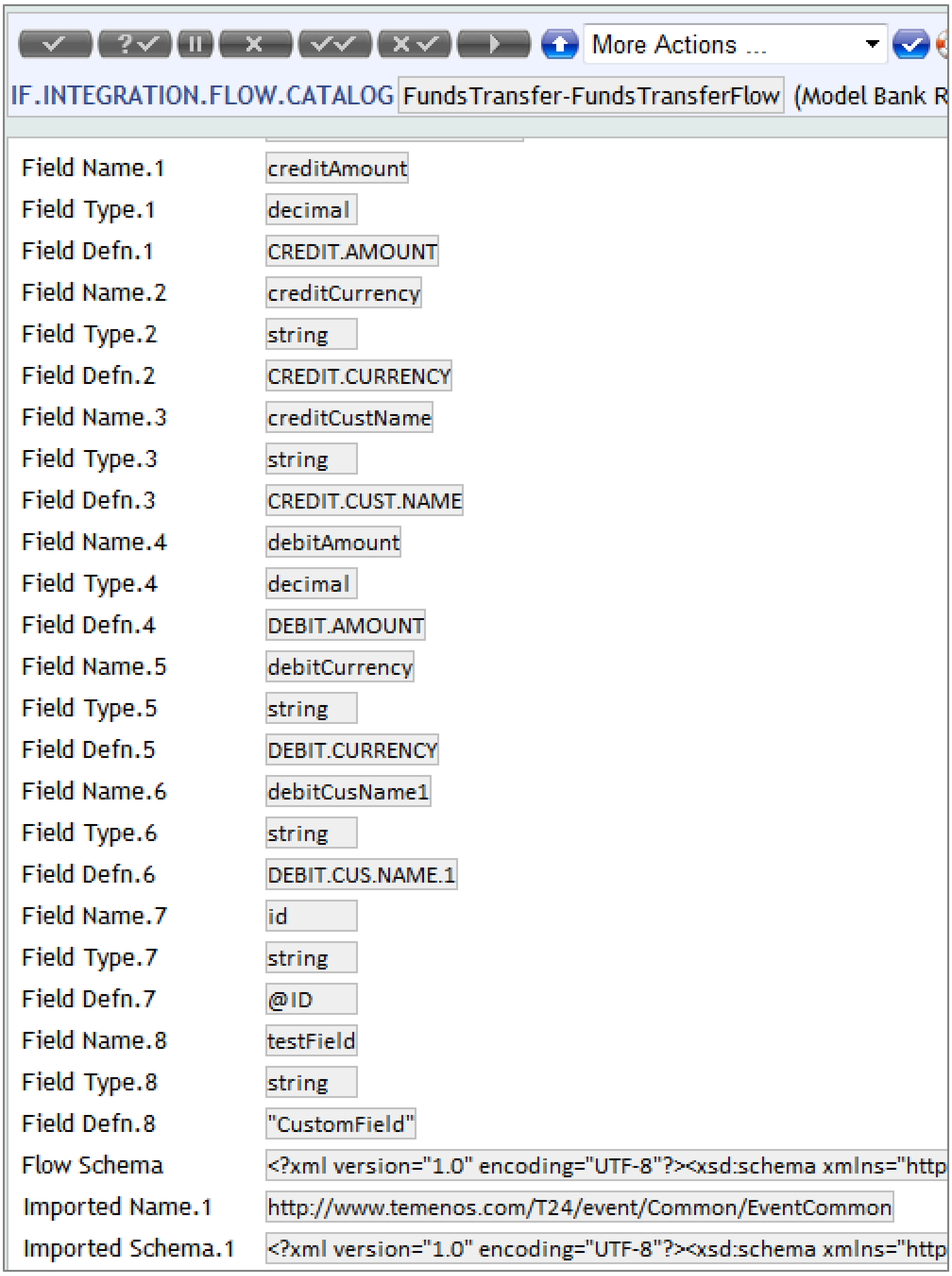
IF.INTEGRATION.FLOW.CATALOG FundsTransfer-FundsTransferFlow (Model Bank)

Source Type	VERSION
Source Name	FUNDS.TRANSFER,ACTR
Exit Point.1	FUNDS.TRANSFER,ACTR
Field Name.1	creditAcctNo
Field Type.1	string
Field Defn.1	CREDIT.ACCT.NO
Field Name.2	creditAcctName
Field Type.2	string
Field Defn.2	CREDIT.ACCT.NAME
Field Name.3	creditAmount
Field Type.3	decimal
Field Defn.3	CREDIT.AMOUNT
Field Name.4	creditCurrency
Field Type.4	string
Field Defn.4	CREDIT.CURRENCY
Field Name.5	creditCustName
Field Type.5	string
Field Defn.5	CREDIT.CUST.NAME
Field Name.6	debitAcctNo
Field Type.6	string
Field Defn.6	DEBIT.ACCT.NO
Field Name.7	debitAcctName



IF.INTEGRATION.FLOW.CATALOG FundsTransfer-FundsTransferFlow

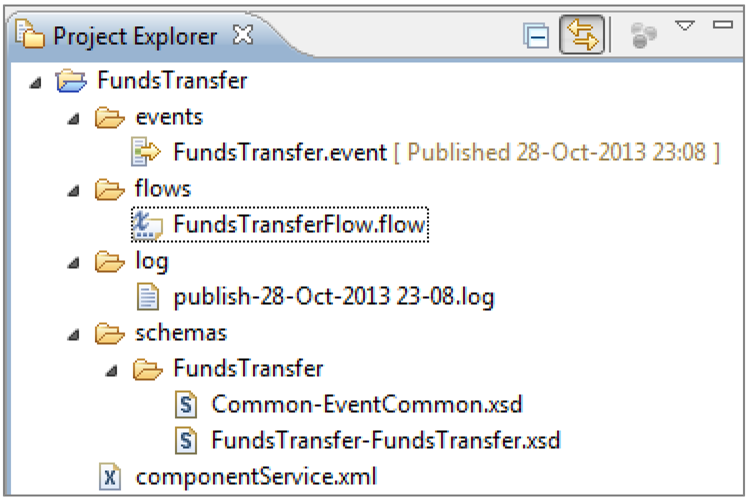
Source Type	VERSION
Source Name	FUNDS.TRANSFER
Exit Point.1	FUNDS.TRANSFER,ACTR
Flow Attributes.1	PROCESS_ONLY_UPDATES
Flow Attributes.2	INCLUDE_BEFORE_IMAGE
Field Name.1	creditAmount
Field Type.1	decimal
Field Defn.1	CREDIT.AMOUNT
Field Name.2	creditCurrency
Field Type.2	string
Field Defn.2	CREDIT.CURRENCY
Field Name.3	creditCustName
Field Type.3	string
Field Defn.3	CREDIT.CUST.NAME
Field Name.4	debitAmount
Field Type.4	decimal
Field Defn.4	DEBIT.AMOUNT
Field Name.5	debitCurrency
Field Type.5	string
Field Defn.5	DEBIT.CURRENCY
Field Name.6	debitCusName1
Field Type.6	string
Field Defn.6	DEBIT.CUS.NAME.1
Field Name.7	id
Field Type.7	string
Field Defn.7	@ID
Field Name.8	testField



IF.INTEGRATION.FLOW.CATALOG FundsTransfer-FundsTransferFlow (Model Bank R

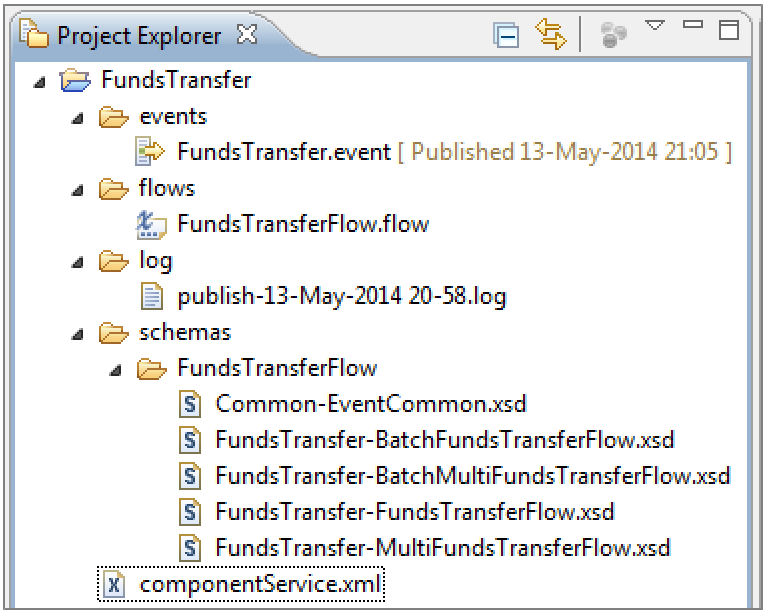
Field Name.1	creditAmount
Field Type.1	decimal
Field Defn.1	CREDIT.AMOUNT
Field Name.2	creditCurrency
Field Type.2	string
Field Defn.2	CREDIT.CURRENCY
Field Name.3	creditCustName
Field Type.3	string
Field Defn.3	CREDIT.CUST.NAME
Field Name.4	debitAmount
Field Type.4	decimal
Field Defn.4	DEBIT.AMOUNT
Field Name.5	debitCurrency
Field Type.5	string
Field Defn.5	DEBIT.CURRENCY
Field Name.6	debitCusName1
Field Type.6	string
Field Defn.6	DEBIT.CUS.NAME.1
Field Name.7	id
Field Type.7	string
Field Defn.7	@ID
Field Name.8	testField
Field Type.8	string
Field Defn.8	"CustomField"
Flow Schema	<?xml version="1.0" encoding="UTF-8"?><xsd:schema xmlns="http
Imported Name.1	http://www.temenos.com/T24/event/Common/EventCommon
Imported Schema.1	<?xml version="1.0" encoding="UTF-8"?><xsd:schema xmlns="http

- After the project is successfully published to T24, the flow schema can be viewed from the Integration project and schemas folder.



Project Explorer

```
FundsTransfer
    events
        FundsTransfer.event [ Published 28-Oct-2013 23:08 ]
    flows
        FundsTransferFlow.flow
    log
        publish-28-Oct-2013 23-08.log
    schemas
        FundsTransfer
            Common-EventCommon.xsd
            FundsTransfer-FundsTransfer.xsd
    componentService.xml
```

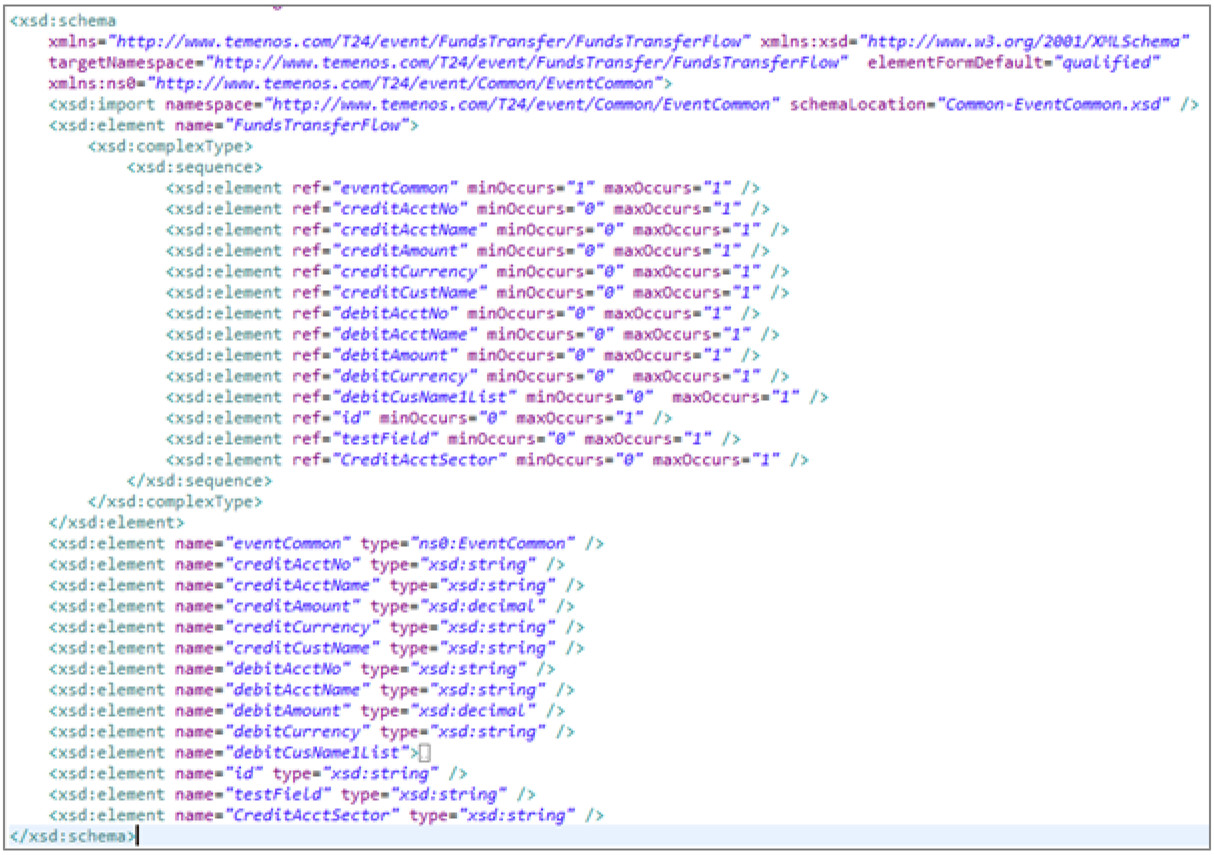


Project Explorer

```
FundsTransfer
    events
        FundsTransfer.event [ Published 13-May-2014 21:05 ]
    flows
        FundsTransferFlow.flow
    log
        publish-13-May-2014 20-58.log
    schemas
        FundsTransferFlow
            Common-EventCommon.xsd
            FundsTransfer-BatchFundsTransferFlow.xsd
            FundsTransfer-BatchMultiFundsTransferFlow.xsd
            FundsTransfer-FundsTransferFlow.xsd
            FundsTransfer-MultiFundsTransferFlow.xsd
    componentService.xml
```

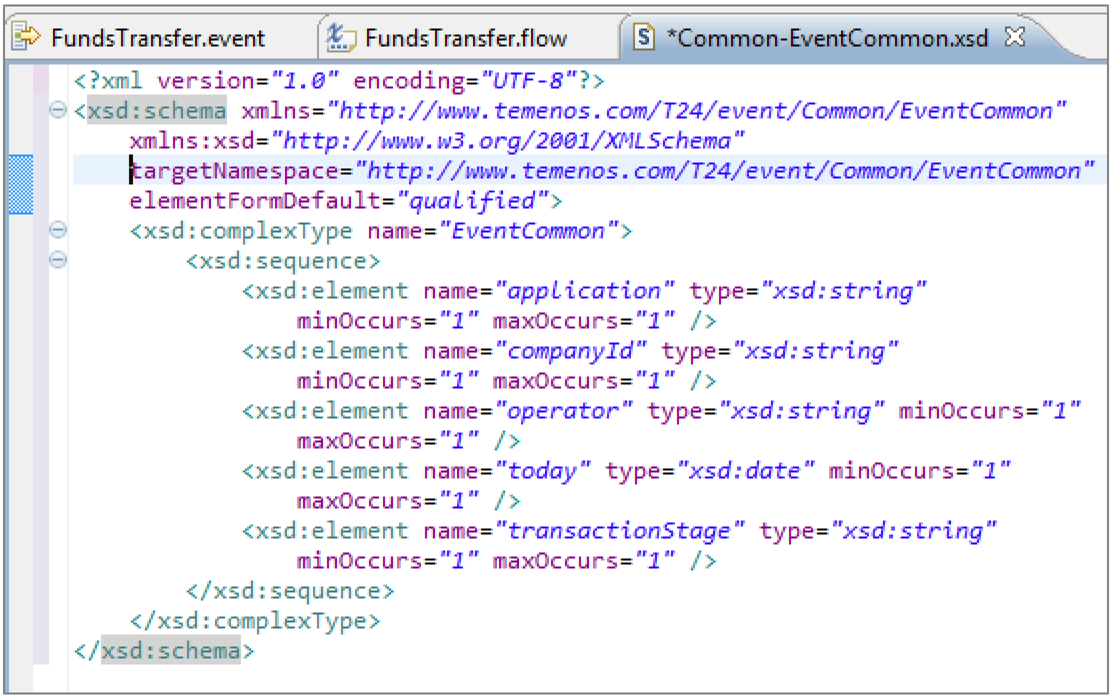
- The user has to refresh the project to get the schema once the schema is published. Now there will be more than one schema created known as schema documents, based on the type of the exit point. There will be two schema documents at least:
- The user has to refresh the project to get the schema once the schema is published. Now there will be more than one schema created known as schema documents, based on the type of the exit point and option include previous image. There will be two schema documents at least:

1. Master Schema



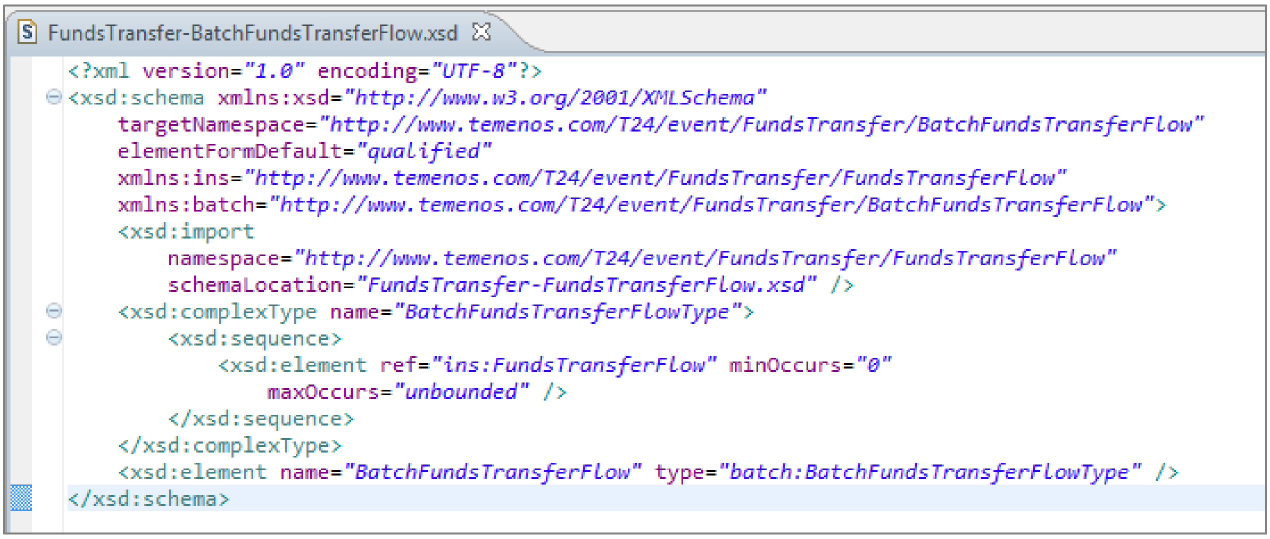
```
<xsd:schema
    xmlns="http://www.temenos.com/T24/event/FundsTransfer/FundsTransferFlow" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://www.temenos.com/T24/event/FundsTransfer/FundsTransferFlow"  elementFormDefault="qualified"
    xmlns:ns0="http://www.temenos.com/T24/event/Common/EventCommon">
    <xsd:import namespace="http://www.temenos.com/T24/event/Common/EventCommon" schemaLocation="Common-EventCommon.xsd" />
    <xsd:element name="FundsTransferFlow">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="eventCommon" minOccurs="1" maxOccurs="1" />
                <xsd:element ref="creditAcctNo" minOccurs="0" maxOccurs="1" />
                <xsd:element ref="creditAcctName" minOccurs="0" maxOccurs="1" />
                <xsd:element ref="creditAmount" minOccurs="0" maxOccurs="1" />
                <xsd:element ref="creditCurrency" minOccurs="0" maxOccurs="1" />
                <xsd:element ref="creditCustName" minOccurs="0" maxOccurs="1" />
                <xsd:element ref="debitAcctNo" minOccurs="0" maxOccurs="1" />
                <xsd:element ref="debitAcctName" minOccurs="0" maxOccurs="1" />
                <xsd:element ref="debitAmount" minOccurs="0" maxOccurs="1" />
                <xsd:element ref="debitCurrency" minOccurs="0"  maxOccurs="1" />
                <xsd:element ref="debitCusName1List" minOccurs="0"  maxOccurs="1" />
                <xsd:element ref="id" minOccurs="0" maxOccurs="1" />
                <xsd:element ref="testField" minOccurs="0" maxOccurs="1" />
                <xsd:element ref="CreditAcctSector" minOccurs="0" maxOccurs="1" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="eventCommon" type="ns0:EventCommon" />
    <xsd:element name="creditAcctNo" type="xsd:string" />
    <xsd:element name="creditAcctName" type="xsd:string" />
    <xsd:element name="creditAmount" type="xsd:decimal" />
    <xsd:element name="creditCurrency" type="xsd:string" />
    <xsd:element name="creditCustName" type="xsd:string" />
    <xsd:element name="debitAcctNo" type="xsd:string" />
    <xsd:element name="debitAcctName" type="xsd:string" />
    <xsd:element name="debitAmount" type="xsd:decimal" />
    <xsd:element name="debitCurrency" type="xsd:string" />
    <xsd:element name="debitCusName1List">
    <xsd:element name="id" type="xsd:string" />
    <xsd:element name="testField" type="xsd:string" />
    <xsd:element name="CreditAcctSector" type="xsd:string" />
</xsd:schema>
```

2. Schema to define common event fields



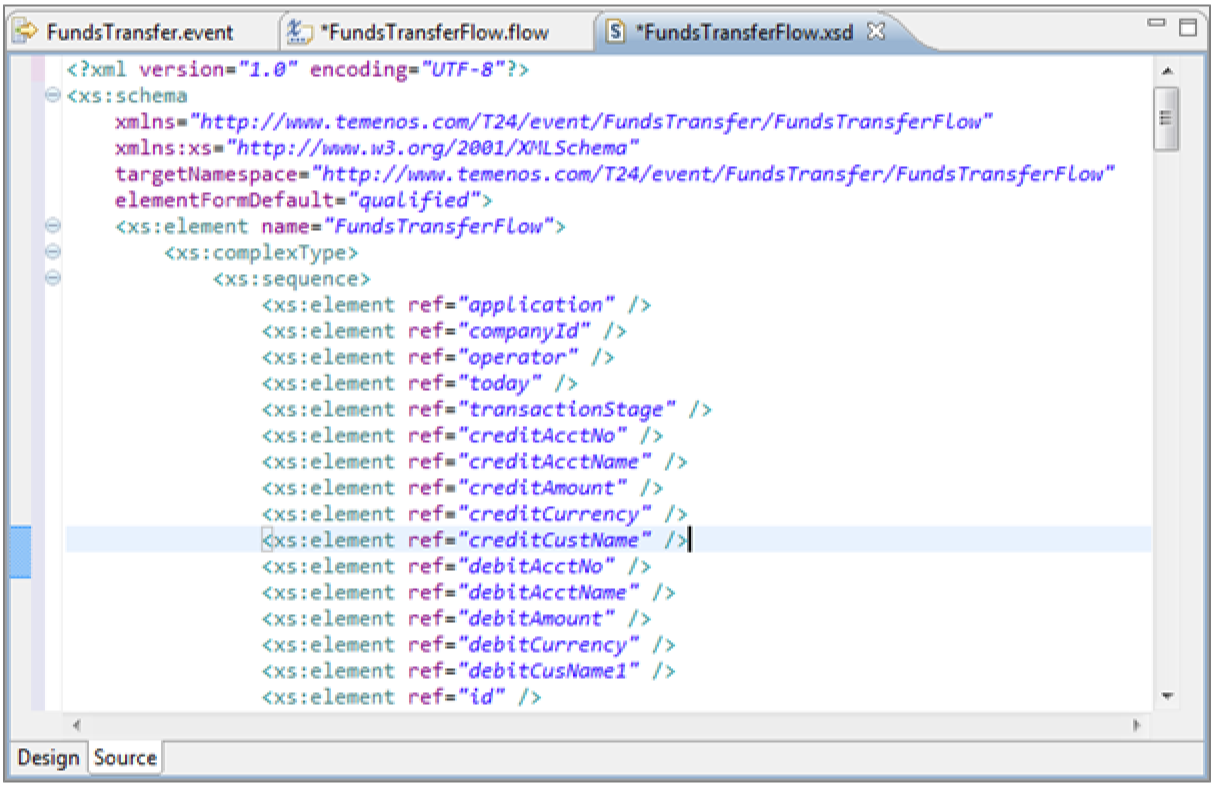
```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="http://www.temenos.com/T24/event/Common/EventCommon"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://www.temenos.com/T24/event/Common/EventCommon"
    elementFormDefault="qualified">
    <xsd:complexType name="EventCommon">
        <xsd:sequence>
            <xsd:element name="application" type="xsd:string"
                minOccurs="1" maxOccurs="1" />
            <xsd:element name="companyId" type="xsd:string"
                minOccurs="1" maxOccurs="1" />
            <xsd:element name="operator" type="xsd:string" minOccurs="1"
                maxOccurs="1" />
            <xsd:element name="today" type="xsd:date" minOccurs="1"
                maxOccurs="1" />
            <xsd:element name="transactionStage" type="xsd:string"
                minOccurs="1" maxOccurs="1" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

3. Batch Schema that provides structure for multiple events of the same type



```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://www.temenos.com/T24/event/FundsTransfer/BatchFundsTransferFlow"
    elementFormDefault="qualified"
    xmlns:ins="http://www.temenos.com/T24/event/FundsTransfer/FundsTransferFlow"
    xmlns:batch="http://www.temenos.com/T24/event/FundsTransfer/BatchFundsTransferFlow">
    <xsd:import
        namespace="http://www.temenos.com/T24/event/FundsTransfer/FundsTransferFlow"
        schemaLocation="FundsTransfer-FundsTransferFlow.xsd" />
    <xsd:complexType name="BatchFundsTransferFlowType">
        <xsd:sequence>
            <xsd:element ref="ins:FundsTransferFlow" minOccurs="0"
                maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="BatchFundsTransferFlow" type="batch:BatchFundsTransferFlowType" />
</xsd:schema>
```

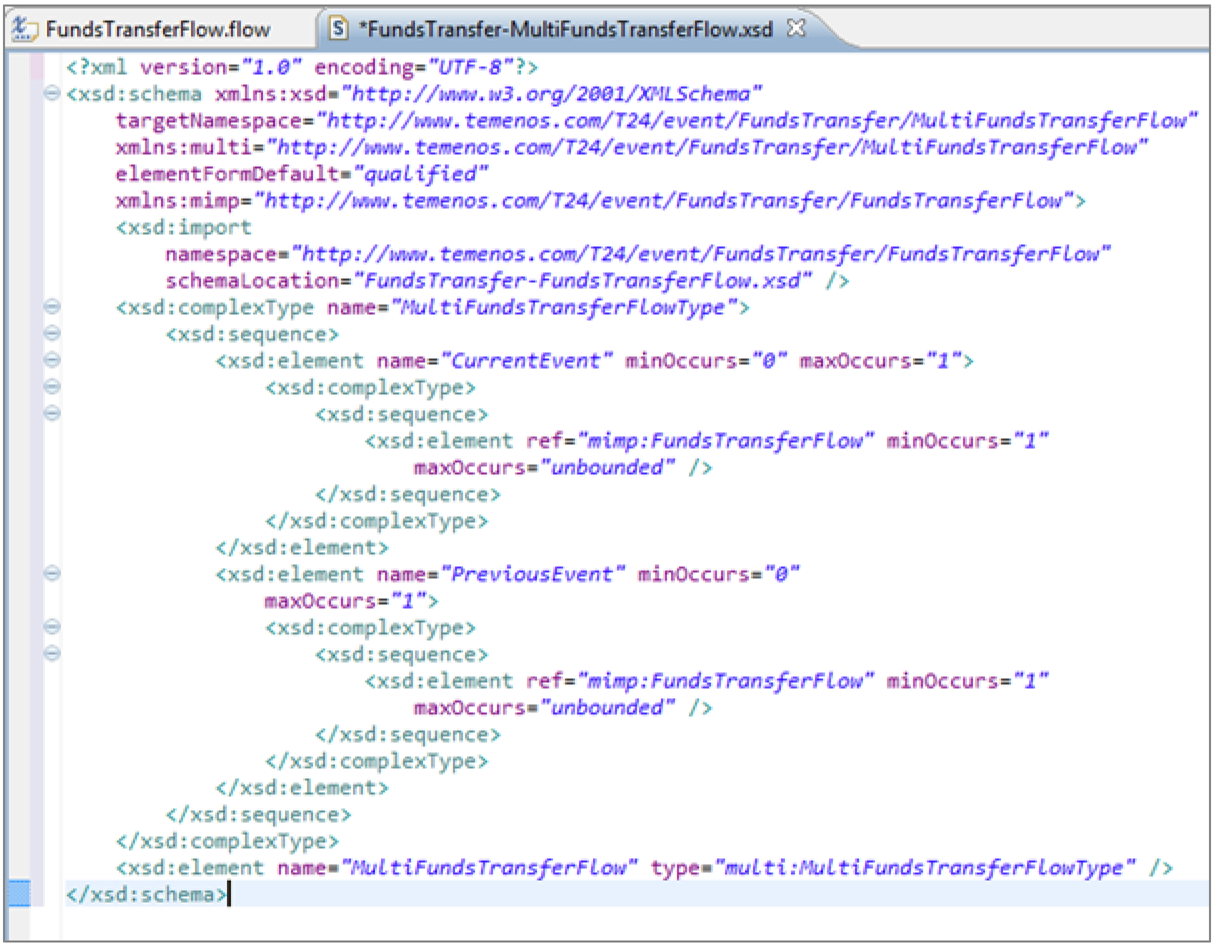
Refresh the project to get the schema, once the schema is published.



```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns="http://www.temenos.com/T24/event/FundsTransfer/FundsTransferFlow"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://www.temenos.com/T24/event/FundsTransfer/FundsTransferFlow"
    elementFormDefault="qualified">
    <xs:element name="FundsTransferFlow">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="application" />
                <xs:element ref="companyId" />
                <xs:element ref="operator" />
                <xs:element ref="today" />
                <xs:element ref="transactionStage" />
                <xs:element ref="creditAcctNo" />
                <xs:element ref="creditAcctName" />
                <xs:element ref="creditAmount" />
                <xs:element ref="creditCurrency" />
                <xs:element ref="creditCustName" />
                <xs:element ref="debitAcctNo" />
                <xs:element ref="debitAcctName" />
                <xs:element ref="debitAmount" />
                <xs:element ref="debitCurrency" />
                <xs:element ref="debitCusName1" />
                <xs:element ref="id" />
```

- Multi Schema:

This represents the schema for the event that had both the current version of data and previous version of data.



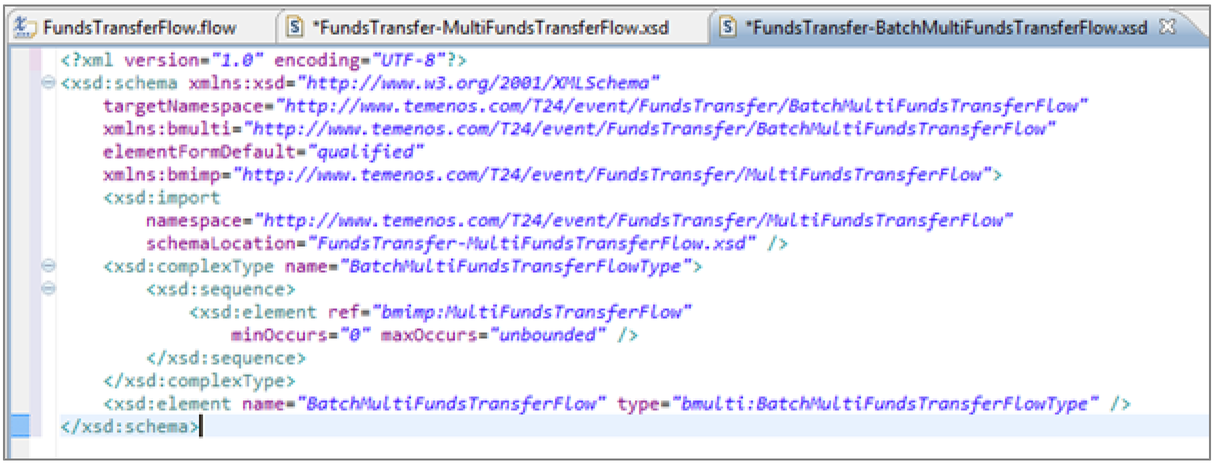
FundsTransferFlow.flow | *FundsTransfer-MultiFundsTransferFlow.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://www.temenos.com/T24/event/FundsTransfer/MultiFundsTransferFlow"
    xmlns:multi="http://www.temenos.com/T24/event/FundsTransfer/MultiFundsTransferFlow"
    elementFormDefault="qualified"
    xmlns:mimp="http://www.temenos.com/T24/event/FundsTransfer/FundsTransferFlow">
    <xsd:import
        namespace="http://www.temenos.com/T24/event/FundsTransfer/FundsTransferFlow"
        schemaLocation="FundsTransfer-FundsTransferFlow.xsd" />
    <xsd:complexType name="MultiFundsTransferFlowType">
        <xsd:sequence>
            <xsd:element name="CurrentEvent" minOccurs="0" maxOccurs="1">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="mimp:FundsTransferFlow" minOccurs="1"
                            maxOccurs="unbounded" />
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="PreviousEvent" minOccurs="0"
                maxOccurs="1">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="mimp:FundsTransferFlow" minOccurs="1"
                            maxOccurs="unbounded" />
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="MultiFundsTransferFlow" type="multi:MultiFundsTransferFlowType" />
</xsd:schema>
```

This schema is generated if and only the option Include Previous image is selected

- Batch Multi Schema

This is an extension of Batch schema and represents batch schema for the event that has both current version of the data and the previous version



FundsTransferFlow.flow | *FundsTransfer-MultiFundsTransferFlow.xsd | *FundsTransfer-BatchMultiFundsTransferFlow.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://www.temenos.com/T24/event/FundsTransfer/BatchMultiFundsTransferFlow"
    xmlns:bmulti="http://www.temenos.com/T24/event/FundsTransfer/BatchMultiFundsTransferFlow"
    elementFormDefault="qualified"
    xmlns:bmimp="http://www.temenos.com/T24/event/FundsTransfer/MultiFundsTransferFlow">
    <xsd:import
        namespace="http://www.temenos.com/T24/event/FundsTransfer/MultiFundsTransferFlow"
        schemaLocation="FundsTransfer-MultiFundsTransferFlow.xsd" />
    <xsd:complexType name="BatchMultiFundsTransferFlowType">
        <xsd:sequence>
            <xsd:element ref="bmimp:MultiFundsTransferFlow"
                minOccurs="0" maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="BatchMultiFundsTransferFlow" type="bmulti:BatchMultiFundsTransferFlowType" />
</xsd:schema>
```

This schema is generated if and only the option Include Previous image is selected

The flow schema generated has :

- The root element name as the flow name.
- The common fields added to the beginning of the flow.
- The T24 field association is now available with the Events generated by integration framework.

Glossary

Event

Generic term for the notification which T24 emits when a business event occurs.

Exit Point

Defines the hook, or API, in T24 to which an Event can be linked. The Exit Point therefore defines when a notification message leaves T24. Exit Points can be linked to T24 VERSIONs, T24 Applications, or to component service operations.

Integration

Flow Defines the layout of an XML message which will leave T24 when a business event occurs. Created in SOA Event Designer at design-time, and referenced at run-time in order that an enriched message can be constructed and dispatched.

Flow Catalog

A T24 table which contains details of all the Integration Flows which have been created by the system integrators. Accessed both at design-time and at run-time. The Flow Catalog provides a component-based interface to Eclipse, Visual Studio and T24 itself.

Reference

Refer the 'Integration Framework Event Delivery' guide, for continuation to this document.