TIBCO ActiveMatrix BusinessWorks™ Plug-in for SWIFT

Examples

Version 6.10.0 | August 2024

Contents

Examples Overview

The examples shipped with TIBCO ActiveMatrix BusinessWorks™ Plug-in for SWIFT demonstrate the basic usage of the plug-in.

After installing the plug-in, you can find five sample projects in the *TIBCO_HOME*/bw/palettes/swift/*version_number*/samples directory. Each sample project is compressed to a .zip file. Before running a sample project, you have to import it. For more information, see Importing Sample Projects.

Note: For information about the prerequisites of running the MX sample projects, see the "Creating a Load SWIFT MX Schema Shared Resource" section in *TIBCO ActiveMatrix BusinessWorks™ Plug-in for SWIFT User Guide*.

The following sample projects are provided:

- BW-activity-MT102-MT103-routing-and-parsing

  This project shows how to use the Route SWIFT MT and the Parse SWIFT MT activities to route and parse MT messages.

  For more information, see Using Route SWIFT MT and Parse SWIFT MT Activities .

- BW-activity-MT535-rendering

  This project shows how to use the Render SWIFT MT activity to render MT messages.

  For more information, see Using Render SWIFT MT Activity.

- BW-activity-BICPlusIBAN-validation-and-generation

  This project shows how to use the Generate SWIFT BICPlusIBAN and the Validate SWIFT BICPlusIBAN activities to generate and validate International Bank Account Number (IBAN).

  For more information, see Using Generate SWIFT BICPlusIBAN and Validate SWIFT BICPlusIBAN Activities.

- MX_Parse_Render

  This project shows how to use the Parse SWIFT MX and the Render SWIFT MX activities to parse and render MX messages.

For more information, see Using Parse SWIFT MX and Render SWIFT MX Activities .

- MT_MX_Mapping

  This project shows how to use the MT and the MX activities to convert SWIFT MT messages to SWIFT MX messages and to convert SWIFT MX messages to SWIFT MT messages.

  For more information, see Mapping SWIFT MT and SWIFT MX Messages.

- CBPRPlus_Parse_Render

  This project shows how to use the Parse SWIFT MX and the Render SWIFT MX activities to parse and render CBPR+ messages.

  For more information, see Using Parse SWIFT MX and Render SWIFT MX Activities for CBPR+ Messages

Importing Sample Projects

Before running a sample project, you have to import it to TIBCO Business Studio™.

Procedure

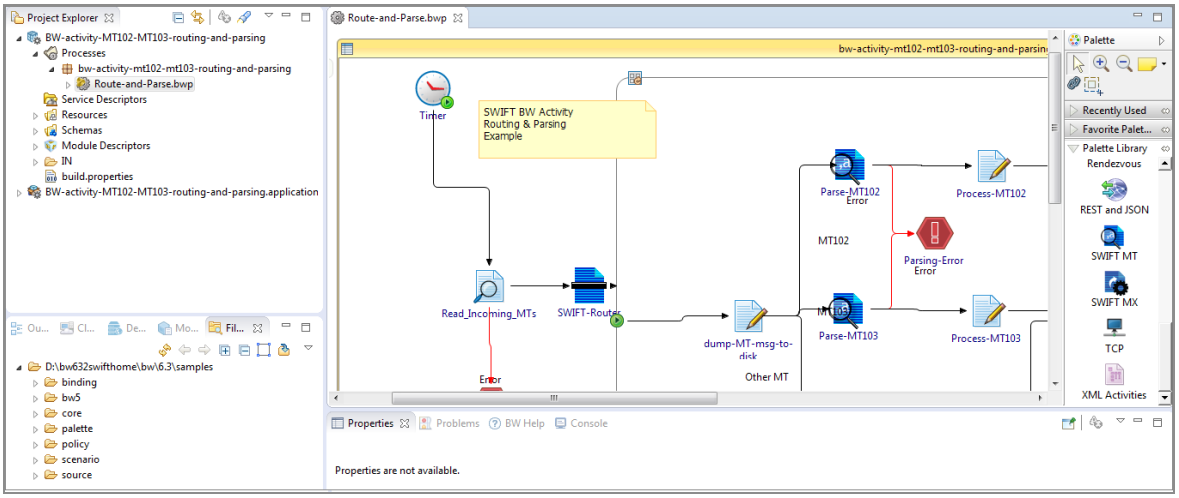
1. Start TIBCO Business Studio.
2. From the menu bar, click **File > Import**.
3. In the Import dialog, expand the **General** folder, and then click **Existing Studio Projects into Workspace**. Click **Next.**
4. In the Import Projects dialog, click **Select archive file**, and then click **Browse** next to it to locate a sample project. For example, select the `BW-activity-MT102-MT103-routing-and-parsing.zip` file, and then click **Open**.

   > **Note:** The sample projects are located in the *`TIBCO_HOME`*`/bw/palettes/swift/`*`version_number`*`/samples` directory.

5. Click **Finish**.

Result

The BW-activity-MT102-MT103-routing-and-parsing project is imported to TIBCO Business Studio.



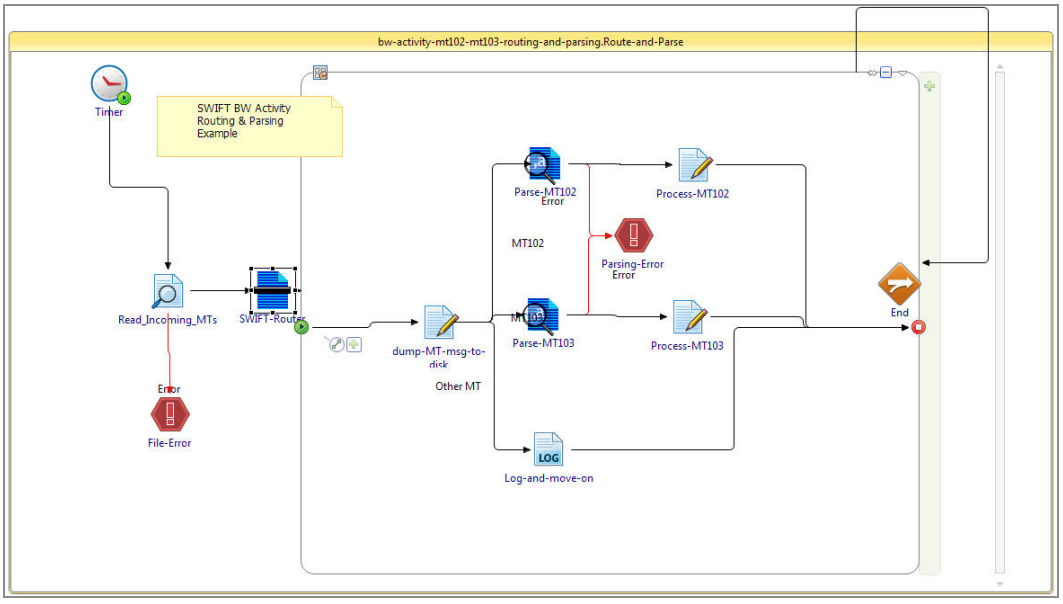
Using Route SWIFT MT and Parse SWIFT MT Activities

The BW-activity-MT102-MT103-routing-and-parsing project contains the Route-and-Parse process, which shows how to use the Route SWIFT MT and the Parse SWIFT MT activities to route and parse MT messages.

Route-and-Parse Process

This process uses a file that contains a multi-message MT string as the input, splits it into individual messages, and then routes them to appropriate activities. The MT102 and MT103 messages are parsed while other messages are simply logged.

The following figure describes the Route-and-Parse process:



The Route-and-Parse process performs the following operations:

1. The Timer activity triggers the process when you run it.
2. The Read_Incoming_MTs activity reads the `$Root/IN/10x-messages.fin` file, and then passes the multi-message MT string to the SWIFT-Router activity.

3. The SWIFT-Router activity splits the multi-message MT string into individual messages, and then passes each message to the dump-MT-msg-to-disk activity in the MT-Processing group.
4. For each message in the file, the following activities take place in the MT-Processing group:
    a. The dump-MT-msg-to-disk activity writes the message to the `/OUT/received-mt#-timestamp.fin` file in the `$Root` directory.
    b. The dump-MT-msg-to-disk activity sends the message to the Parse activities according to the message type. An MT 102 message is sent to the Parse-MT 102 activity, an MT 103 message is sent to the Parse-MT103 activity, and any other message is sent to the Log-and-move-on activity.
    c. The Parse-MT102 and the Parse-MT103 activities parse the incoming MT 102 and MT 103 messages to a structured format.
    d. The Process-MT102 and the Process-MT103 activities write the structured message to the `$Root` directory as `/OUT/processed-mt#-timestamp.txt`. This structured message can be easily manipulated, merged, mapped, and so on, by using any other TIBCO ActiveMatrix BusinessWorks activities.
    e. The Log-and-move-on activity logs any message that is not of the MT102 or MT103 type.

Setting Up the Project

Before running the project, you have to set it up.

Before you begin

You have to import the BW-activity-MT102-MT103-routing-and-parsing project before setting it up. For more information, see Importing Sample Projects.

Procedure

1. Expand the imported project in the Project Explorer view.
2. Expand the **Module Descriptors** resource, and then double-click **Module Properties**.
3. In the Module Properties panel, expand the **SWIFTExamples** folder, and then set the value of the `Root` parameter to the path where the project is located in your

workspace.

For example, `C:/workspace_SWIFT_example/BW-activity-MT102-MT103-routing-and-parsing`.

4. From the menu bar, click **File > Save** to save the project.

Running the Project

After setting up the project, you can run it.

Procedure

1. In the Project Explorer view, expand the **Module Descriptors** resource, and then double-click **Components**.
2. In the Components editor, click  and select the process that you want to run.
3. On the toolbar, click the  icon to save your changes.
4. From the menu bar, click **Run > Debug Configurations** to run the selected process.
5. In the Debug Configurations dialog, expand **BusinessWorks Application**, and then click **BWApplication**.
6. In the right panel, click the **Applications** tab, and then select the checkbox next to **BW-activity-MT102-MT103-routing-and-parsing.application**.
7. Click **Debug** to run the process.
8. Click the  icon to stop the process.

Result

When the Route-and-Parse process runs successfully, the `$Root/OUT` directory is automatically generated, and the following files are contained in this folder:

- `processed-mt102-20200206170256026.txt`
- `processed-mt102-20205506165530789.txt`
- `processed-mt102-20205506165531516.txt`
- `processed-mt103-20205506165531172.txt`

- processed-mt103-20205506165531717.txt
- received-mt102-20200206170255760.fin
- received-mt102-20205506165528703.fin
- received-mt102-20205506165531415.fin
- received-mt102STP-20200206170256921.fin
- received-mt102STP-20205506165530882.fin
- received-mt103-20200206170257494.fin
- received-mt103-20205506165530986.fin
- received-mt103-20205506165531621.fin
- received-mt104-20205506165531259.fin

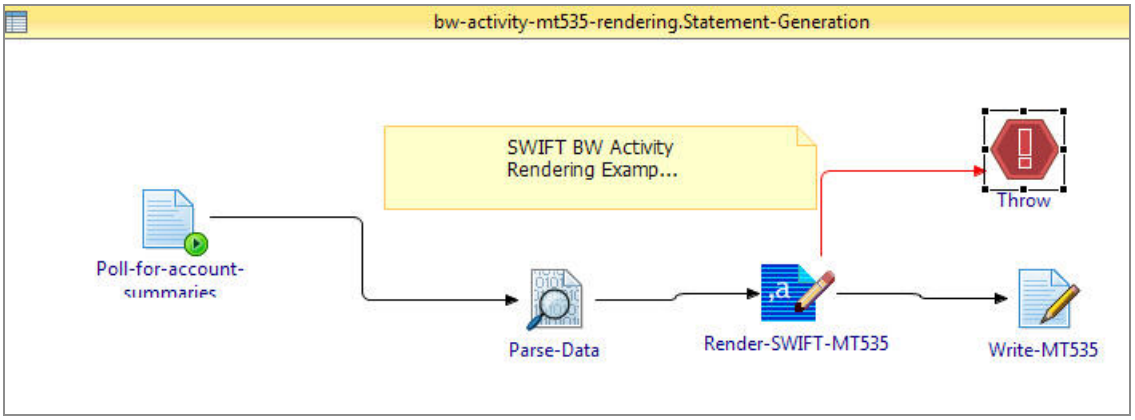
Using Render SWIFT MT Activity

The BW-activity-MT535-rendering project contains the Statement-Generation process, which shows how to use the Render SWIFT MT activity to render MT messages.

Statement-Generation Process

This process uses account summary files as the input, parses the data from the files, passes the data to the Render activity, and then produces MT535 "statements of holdings" MT messages.

The following figure describes the Statement-Generation process:



The Statement-Generation process performs the following operations:

1. The Poll-for-account-summaries activity searches for account summary files in the `$Root/holdings` directory.

   An account summary file is a text file, and each line in the file contains four values: Account Number, ISIN, Quantity, and Quantity Type. For example, an account summary file contains:

   ```
   014010|ISIN US88632Q1031|10004,0|UNIT
   0014010|ISIN US3838831051|02000,0|UNIT
   0014010|ISIN NL0000009470|82000,0|UNIT
   0014010|ISIN ES0130670112|00100,0|UNIT
   ```

```
0014010|ISIN FR0000120073|04587,0|UNIT
0014010|ISIN US7170811035|00153,0|UNIT
0032020|ISIN BE0003796134|00511,0|UNIT
0032030|ISIN BE0003796134|45000,0|UNIT
0045600|ISIN UX383XX31052|04344,0|UNIT
```

2. The Parse-Data activity parses the data from the account summary files into a structured format, and passes it to the Render-SWIFT-MT535 activity.
3. The Render-SWIFT-MT535 activity produces an MT535 "statement of holdings" MT message.
4. The Write-MT535 activity writes the MT message to the `mt535-acct-`*`accountnumber`*`.txt` file in the `$Root` directory.

Setting Up the Project

Before running the project, you have to set it up.

Before you begin

You have to import the BW-activity-MT535-rendering project before setting it up. For more information, see Importing Sample Projects.

Procedure

1. Expand the imported project in the Project Explorer view.
2. Expand the **Module Descriptors** resource, and then double-click **Module Properties**.
3. In the Module Properties panel, expand the **SWIFTExamples** folder, and then complete the following tasks:
   a. Set the value of the `Root` parameter to the path where the project is located in your workspace.

      For example, `C:/workspace_SWIFT_example/BW-activity-MT535-rendering`.

   b. Set the value of the `FileName` parameter to the path where the account summary file is located in your workspace.

      For example, `C:/workspace_SWIFT_example/BW-activity-MT535-rendering/holdings/acct-*`.

4. From the menu bar, click **File > Save** to save the project.

Running the Project

After setting up the project, you can run the process.

Procedure

1. In the Project Explorer view, expand the **Module Descriptors** resource, and then double-click **Components**.
2. In the Components editor, click  and select the process that you want to run.
3. On the toolbar, click the  icon to save your changes.
4. From the menu bar, click **Run > Debug Configurations** to run the selected process.
5. In the Debug Configurations dialog, expand **BusinessWorks Application** , and then click **BWApplication**.
6. In the right panel, click the **Applications** tab, and then select the checkbox next to **BW-activity-MT535-rendering.application**.
7. Click **Debug** to run the process.
8. Click the  icon to stop the process.

Result

After the Statement- Generation process runs successfully, two new text files are generated in the `$Root` directory, they are `mt535-acct-0032030.txt` and `mt535-acct-0053440.txt`.

Take the `mt535-acct-0032030.txt` file for example. This file contains the following information:

```
{1:F01TIBCNL20AXXX0999012345}{2:O5351355030929LRLRXXXX4A0300003
997421507311355N}{3:{108:MT535}}{4:
   :16R:GENL
   :28E:00001/ONLY
   :20C::SEME//01295
   :23G:NEWM
   :98A::STAT//20150731
   :22F::SFRE//ADHO
```

```
    :22F::CODE//COMP
    :22F::STTY//CUST
    :22F::STBA//CONT
    :97A::SAFE//0014011
    :17B::ACTI//Y
    :17B::CONS//N
    :16S:GENL
    :16R:SUBSAFE
    :16R:FIN
    :35B:ISIN US88632Q1031
    :90A::MRKT//DISC/1,3456789012345
    :93B::AGGR//UNIT/N10004,0
    :19A::HOLD//NUSD1,23
    :16S:FIN
    :16S:SUBSAFE
    -}
```

Using Generate SWIFT BICPlusIBAN and Validate SWIFT BICPlusIBAN Activities

The BW-activity-BICPlusIBAN-validation-and-generation project shows how to use the Generate SWIFT BICPlusIBAN and Validate SWIFT BICPlusIBAN activities to generate and validate International Bank Account Number (IBAN).

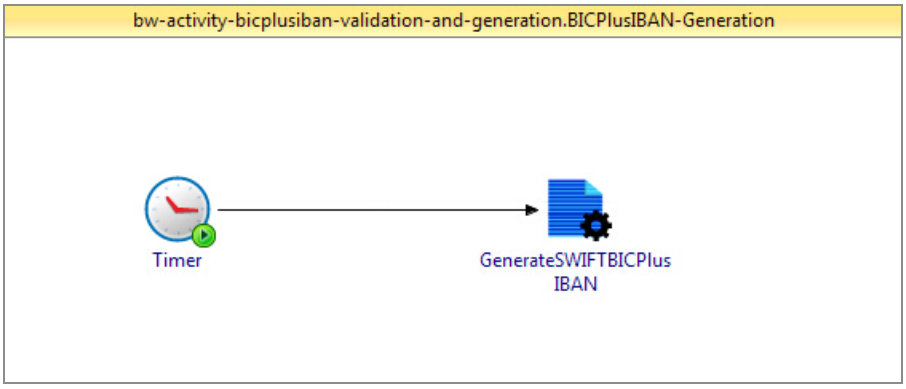
This project contains the following processes:

- BICPlusIBAN-Generation Process
- BICPlusIBAN-Validation Process

BICPlusIBAN-Generation Process

You can use the BICPlusIBAN-Generation process to generate the IBAN with the country code and Basic Bank Account Number (BBAN).

The following figure describes the BICPlusIBAN-Generation process:



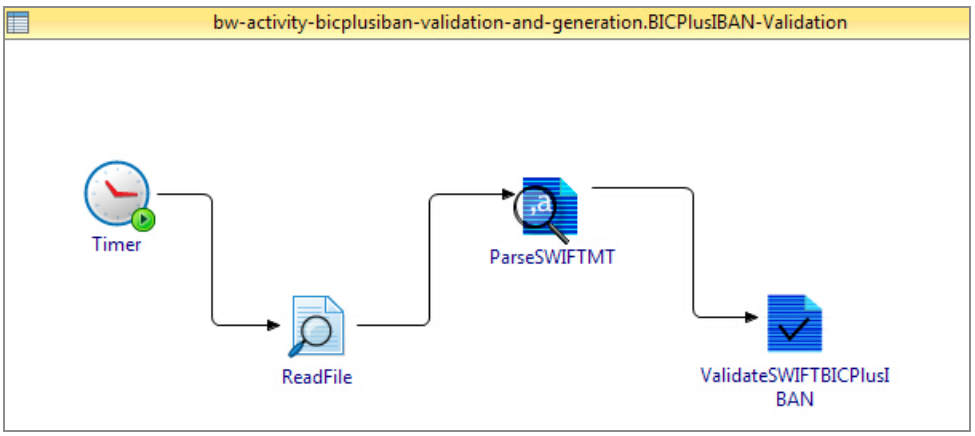
The BICPlusIBAN-Generation process performs the following operations:

1. Validating the BBAN
2. Generating check digits

BICPlusIBAN-Validation Process

You can use the BICPlusIBAN-Validation process to validate the IBAN through the IBAN, BIC (Bank Identifier Code), and Branch Code.

The following figure describes the BICPlusIBAN-Validation process:



The BICPlusIBAN-Validation process performs the following operations:

1. The Read File activity reads the `$PROJ_ROOT/IN/MT102.fin` file, and passes it to the ParseSWIFTMT activity.
2. The ParseSWIFTMT activity validates the incoming MT102 message. If the message is correct, it is passed to the ValidateSWIFTBICPlusIBAN activity.
3. The ValidateSWIFTBICPlusIBAN activity validates the IBAN, BIC, and Branch Code in the message. If the information is correct, the whole record of the information about the financial institutions and their IBAN-related data is generated.

Setting Up the Project

Before running the project, you have to set it up.

Before you begin

You have to import the BW-activity-BICPlusIBAN-validation-and-generation project before setting it up. For more information, see Importing Sample Projects.

Procedure

1. Expand the imported project in the Project Explorer view.
2. Expand the **Module Descriptors** resource, and then double-click **Module Properties**.
3. In the Module Properties panel, expand the **SWIFT** folder, and then complete the following tasks:
    a. Set the value of the `PROJ_ROOT` parameter to the path where the project is located in your workspace. For example, `C:/workspace_SWIFT_example/BW-activity-BICPlusIBAN-validation-and-generation`.
    b. Set the value of the **BICPLUSIBAN_PATH** parameter to the path where the BANK Directory and the IBAN Plus files are located. For example, `D:\swift\BICPLUSIBAN_CORE_FULL`

    > **Note:** TIBCO does not provide the BANK Directory and IBAN Plus files because of copyright restriction. You can download them from https://www.swift.com/.

4. From the menu bar, click **File > Save All** to save the project.

Running the Project

After setting up the project, you can run it.

Procedure

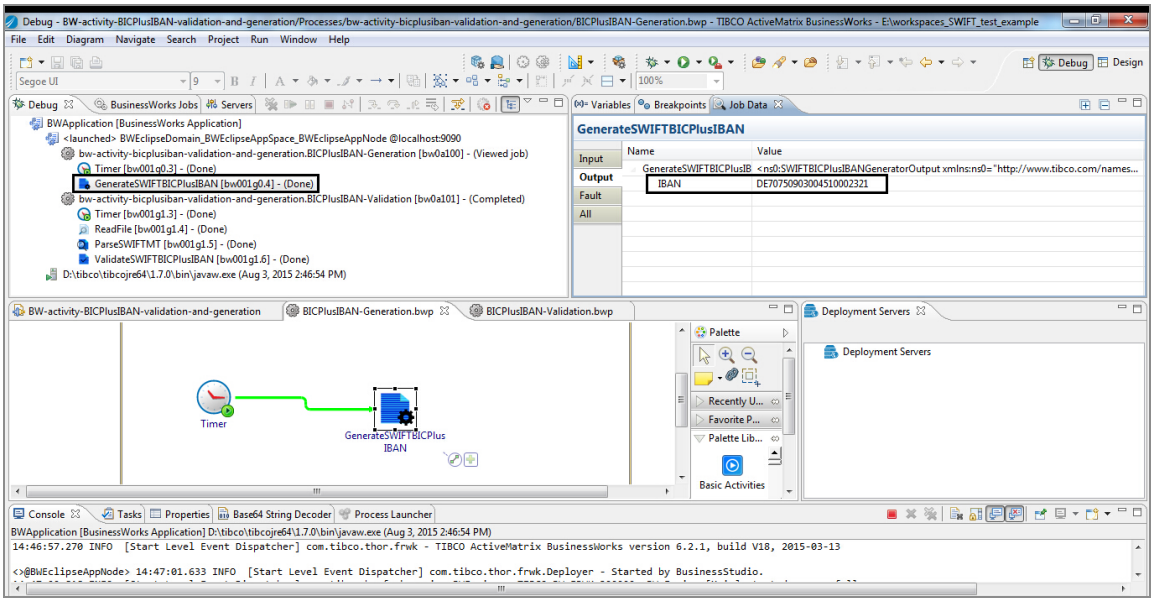
1. In the Project Explorer view, expand the **Module Descriptors** resource, and then double-click **Components**.
2. In the Components editor, click  and select the process that you want to run.
3. On the toolbar, click the  icon to save your changes.
4. From the menu bar, click **Run > Debug Configurations** to run the selected process.
5. In the Debug Configurations dialog, expand **BusinessWorks Application**, and then click **BWApplication**.
6. In the right panel, click the **Applications** tab, and then select the checkbox next to **BW-activity-BICPlusIBAN-validation-and-generation.application**.
7. Click **Debug** to run the process.

8. Click the  icon to stop the process.

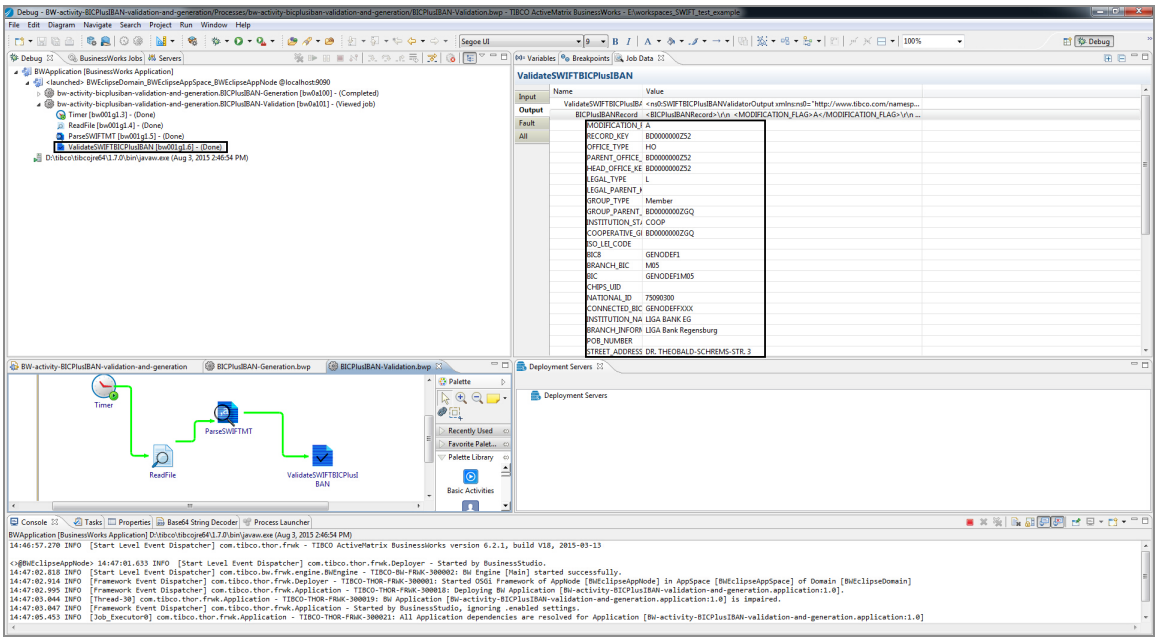
Result

After the processes run successfully, you can check the results.

- Expected results of the BICPlusIBAN-Generation process: an IBAN is generated.



- Expected results of the BICPlusIBAN-Validation process: the validation result is displayed.



Using Parse SWIFT MX and Render SWIFT MX Activities

The MX_Parse_Render project shows how to use the Parse SWIFT MX and the Render SWIFT MX activities to parse and render MX messages.

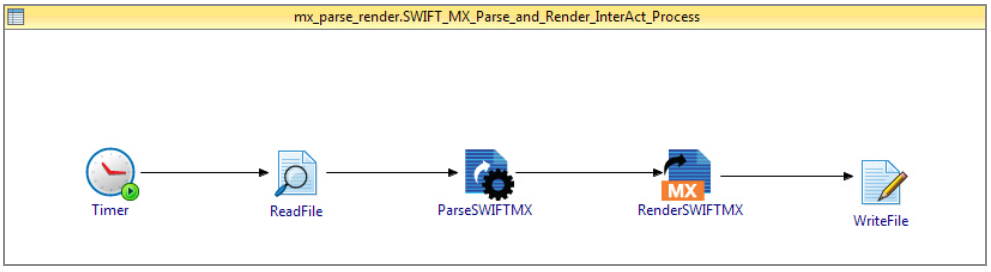
This project contains the following processes:

- SWIFT_MX_Parse_and_Render_InterAct_Process
- SWIFT_MX_Parse_Document_Process
- SWIFT_MX_Parse_Document_with_ValidationFilter_Process
- SWIFT_MX_Parse_InterAct_ISOHeader_Process
- SWIFT_MX_Parse_SAA_Process

SWIFT_MX_Parse_and_Render_InterAct_Process

You can use the SWIFT_MX_Parse_and_Render_InterAct_Process to validate and parse the content in the incoming message, and to generate a new message.

The following figure describes the SWIFT_MX_Parse_and_Render_InterAct_Process:



The SWIFT_MX_Parse_and_Render_InterAct_Process performs the following operations:

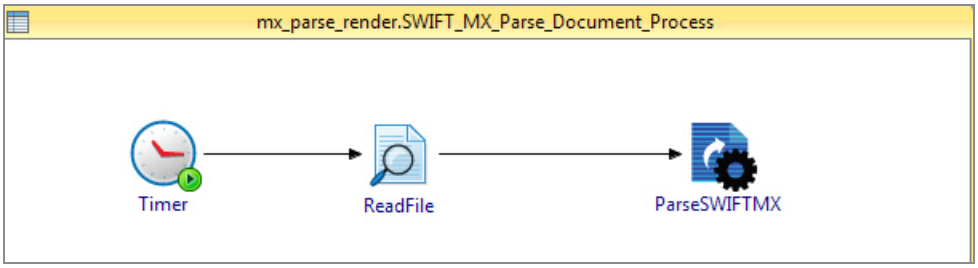
1. The ReadFile activity reads an MX message, and passes it to the ParseSWIFTMX activity.
2. The ParseSWIFTMX activity validates and parses content of the InterAct type message, and passes it to the RenderSWIFTMX activity.

3. The Render SWIFTMX activity validates content of the InterAct type message, generates a specific MX message, and passes it to the WriteFile activity.
4. The Write File activity writes an MX message to the `sese.023.001.010_InterAct_out.xml` file in the `$Example_Root/out` directory.

SWIFT_MX_Parse_Document_Process

You can use the SWIFT_MX_Parse_Document_Process to validate and parse the content of the document type message.

The following figure describes the SWIFT_MX_Parse_Document_Process:



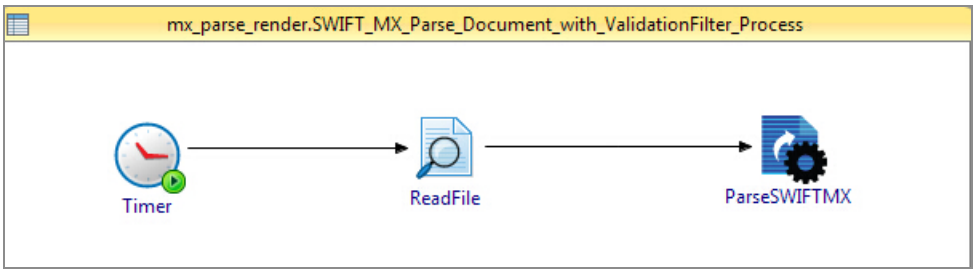
The SWIFT_MX_Parse_Document_Process performs the following operations:

1. The ReadFile activity reads an MX message, and passes it to the ParseSWIFTMX activity.
2. The ParseSWIFTMX activity validates and parses the content of the document type message, and produces an MX message.

SWIFT_MX_Parse_Document_with_ValidationFilter_Process

You can use the SWIFT_MX_Parse_Document_with_ValidationFilter_Process to validate and parse the content of the document type message.

The following figure describes the SWIFT_MX_Parse_Document_with_ValidationFilter_Process:



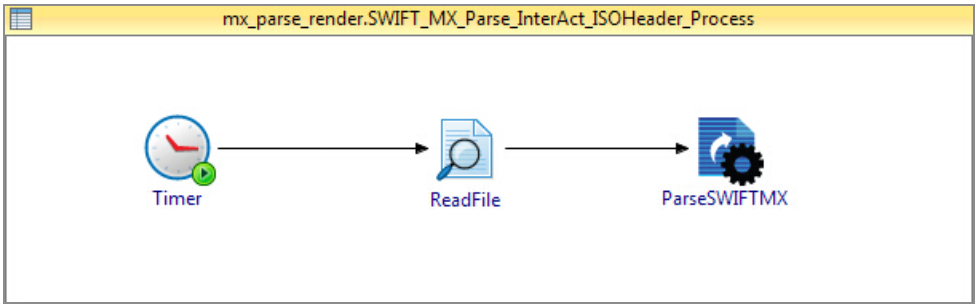
The SWIFT_MX_Parse_Document_with_ValidationFilter_Process performs the following operations:

1. The ReadFile activity reads an MX message, and passes it to the ParseSWIFTMX activity.
2. The ParseSWIFTMX activity validates and parses the content of the document type message with a validation filter file, and produces an MX message.

SWIFT_MX_Parse_InterAct_ISOHeader_Process

You can use the SWIFT_MX_Parse_InterAct_ISOHeader_Process to validate and parse the content of the InterAct ISOHeader type message.

The following figure describes the SWIFT_MX_Parse_InterAct_ISOHeader_Process:



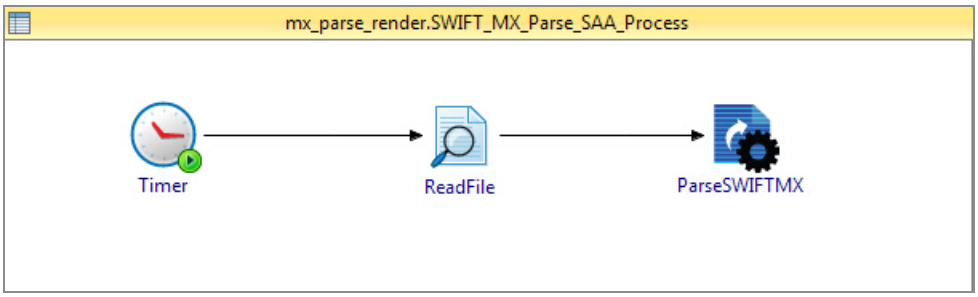
The SWIFT_MX_Parse_InterAct_ISOHeader_Process performs the following operations:

1. The ReadFile activity reads an MX message, and passes it to the ParseSWIFTMX activity.
2. The ParseSWIFTMX activity validates and parses the content of the InterAct type message with ISO AppHeader, and produces an MX message.

SWIFT_MX_Parse_SAA_Process

You can use the SWIFT_MX_Parse_SAA_Process to validate and parse the content of the SWIFT Alliance Access (SAA) type message.

The following figure describes the SWIFT_MX_Parse_SAA_Process:



The SWIFT_MX_Parse_SAA_Process performs the following operations:

1. The ReadFile activity reads an MX message, and passes it to the ParseSWIFTMX activity.
2. The ParseSWIFTMX activity validates and parses the content of the SAA type message, and produces an MX message.

Setting Up the Project

Before running the project, you have to set it up.

Before you begin

You have to import the MX_Parse_Render project before setting it up. For more information, see Importing Sample Projects.

Procedure

1. Expand the imported project in the Project Explorer view.
2. Expand the **Module Descriptors** resource, and then double-click **Module Properties**.
3. In the Module Properties panel, expand the **Example** folder, and then complete the following tasks:
    a. Set the value of the `Example_ROOT` parameter to the path where the project is located in your workspace.

       For example, `C:/workspace_SWIFT_example/MX_Parse_Render`.

    b. Set the value of the `ValidationFilter` parameter to the path where the validation file is located in your workspace.

       For example, `C:/workspace_SWIFT_example/MX_Parse_`

`Render/filter/ValidationFilter.xml`.

4. From the menu bar, click **File > Save** to save the project.
5. Download the required .xsd files that correspond to the MX message files in the `$Example_ROOT/in` directory from the SWIFT website, and then save them in the *`TIBCO_HOME`*`/bw/palettes/swift/`*`version_number`*`/bin/xsd/`*`year`* directory.

Running the Project

After setting up the project, you can run it.

Procedure

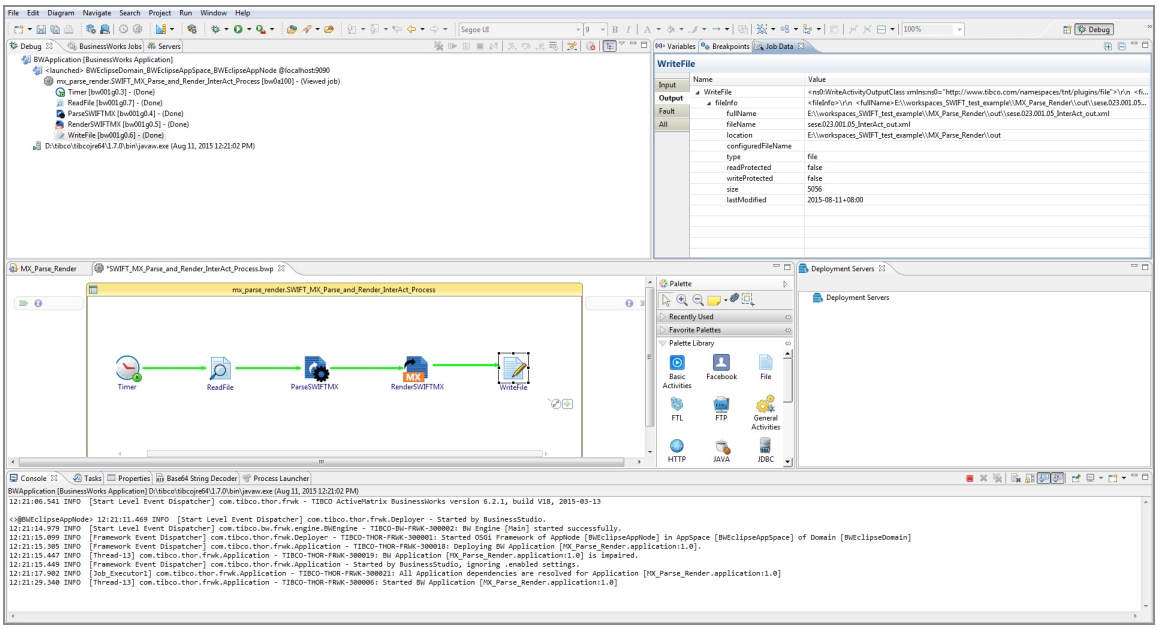
1. In the Project Explorer view, expand the **Module Descriptors** resource, and then double-click **Components**.
2. In the Components editor, click  and select the process that you want to run.
3. On the toolbar, click the  icon to save your changes.
4. From the menu bar, click **Run > Debug Configurations** to run the selected process.
5. In the Debug Configurations dialog, expand **BusinessWorks Application**, and then click **BWApplication**.
6. In the right panel, click the **Applications** tab, and then select the checkbox next to **MX_Parse_Render.application**.
7. Click **Debug** to run the process.
8. Click the  icon to stop the process.

Result

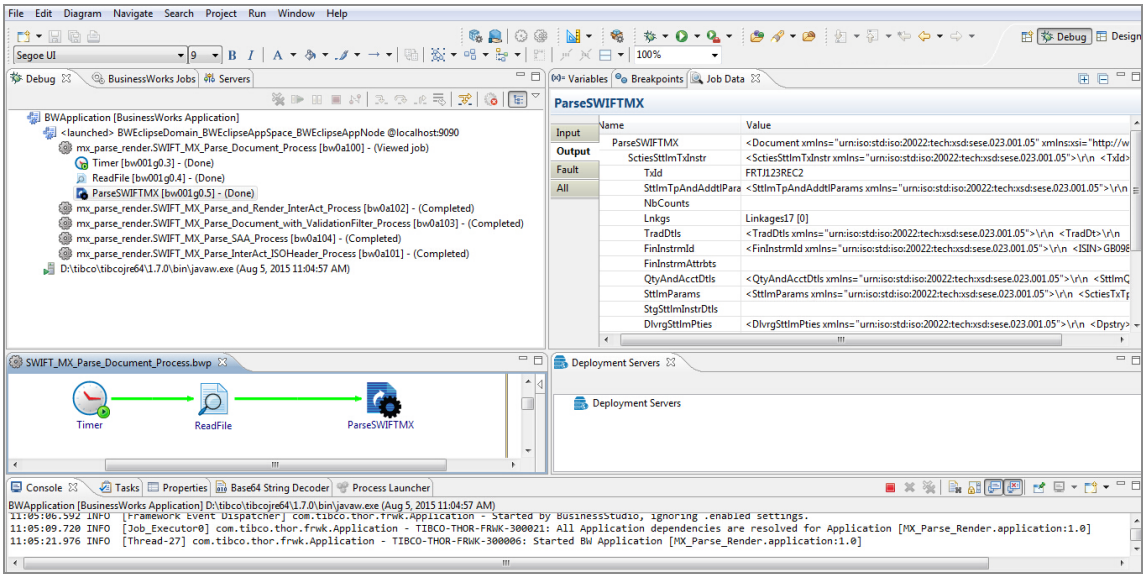
After the processes run successfully, you can check the results.

- Expected Result of the SWIFT_MX_Parse_and_Render_InterAct_Process

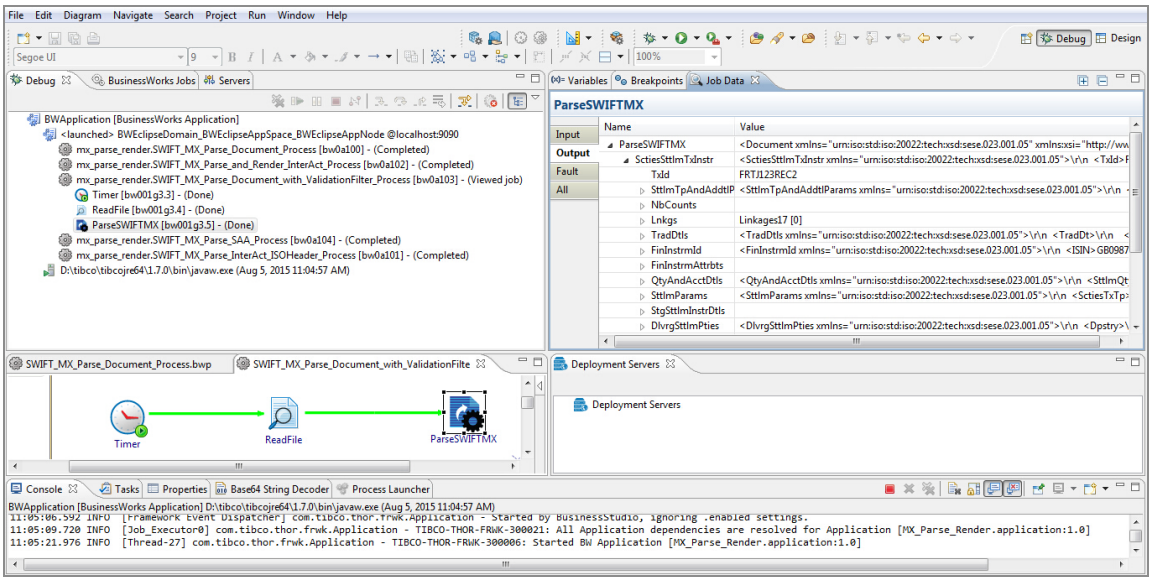
  After this process runs successfully, a `sese.023.001.010_InterAct_out.xml` file is generated in the `$Example_ROOT/out` directory.



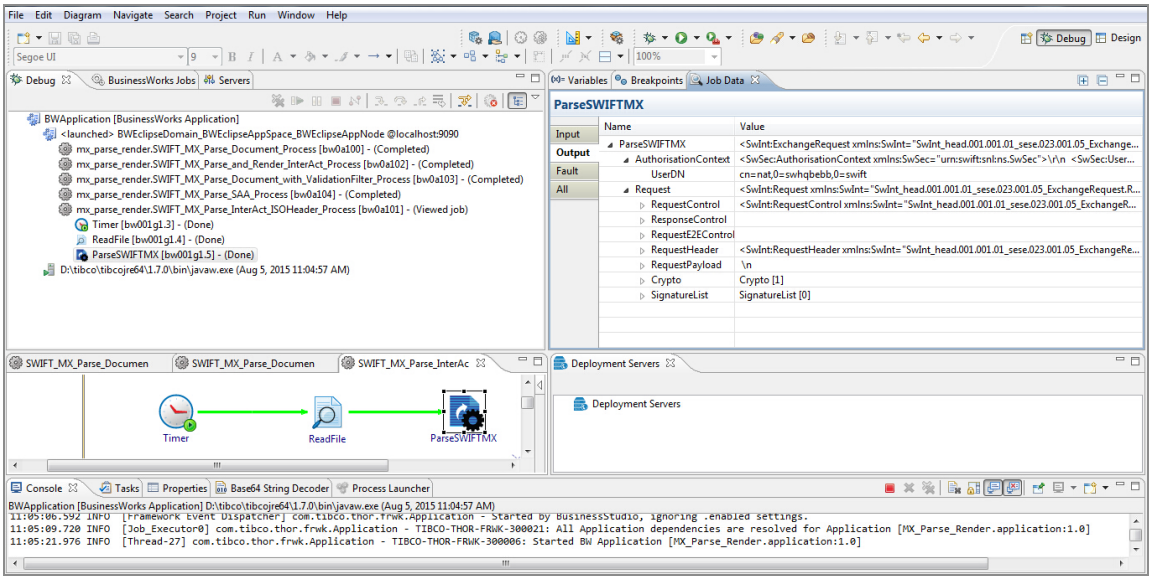
- Expected Result of the SWIFT_MX_Parse_Document_Process



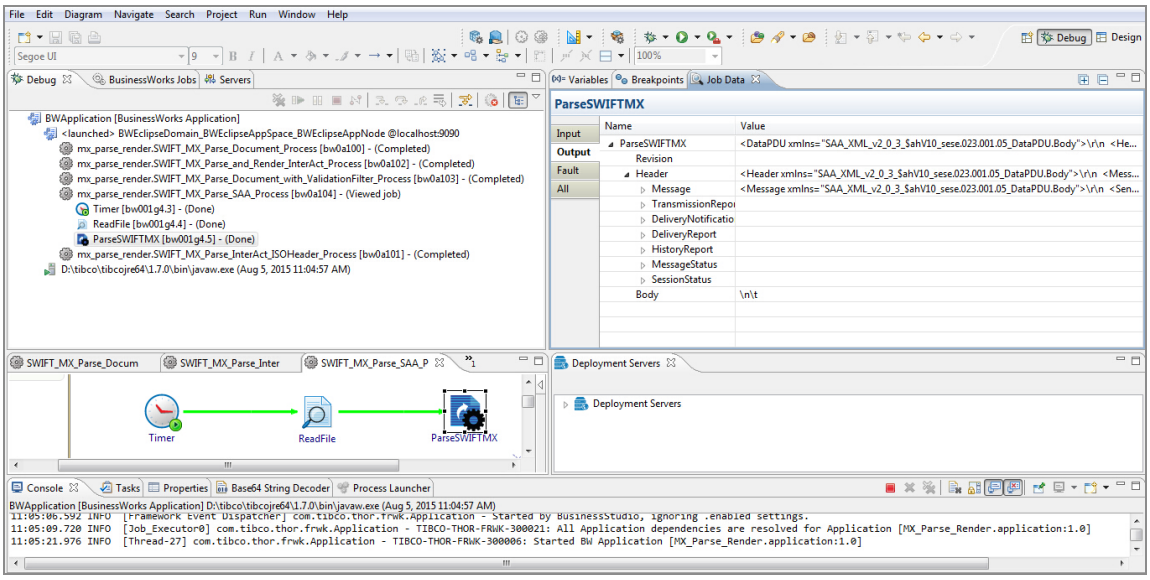
- Expected Result of the SWIFT_MX_Parse_Document_with_ValidationFilter_Process



- Expected Result of SWIFT_MX_Parse_InterAct_ISOHeader_Process



- Expected Result of the SWIFT_MX_Parse_SAA_Process


File Edit Diagram Navigate Search Project Run Window Help
Debug
Design
Debug
BusinessWorks Jobs
Servers
BWApplication [BusinessWorks Application]
<launched> BWEclipseDomain_BWEclipseAppSpace_BWEclipseAppNode @localhost:9090
mx_parse_render.SWIFT_MX_Parse_Document_Process [bw0a100] - (Completed)
mx_parse_render.SWIFT_MX_Parse_and_Render_InterAct_Process [bw0a102] - (Completed)
mx_parse_render.SWIFT_MX_Parse_Document_with_ValidationFilter_Process [bw0a103] - (Completed)
mx_parse_render.SWIFT_MX_Parse_SAA_Process [bw0a104] - (Viewed job)
Timer [bw001g4.3] - (Done)
ReadFile [bw001g4.4] - (Done)
ParseSWIFTMX [bw001g4.5] - (Done)
mx_parse_render.SWIFT_MX_Parse_InterAct_ISOHeader_Process [bw0a101] - (Completed)
D:\tibco\tibcojre64\1.7.0\bin\javaw.exe (Aug 5, 2015 11:04:57 AM)
Variables
Breakpoints
Job Data
ParseSWIFTMX
Input
Output
Fault
All
Name
Value
ParseSWIFTMX
Revision
Header
Message
TransmissionRepor
DeliveryNotificatio
DeliveryReport
HistoryReport
MessageStatus
SessionStatus
Body
SWIFT_MX_Parse_Docum
SWIFT_MX_Parse_Inter
SWIFT_MX_Parse_SAA_P
Timer
ReadFile
ParseSWIFTMX
Deployment Servers
Deployment Servers
Console
Tasks
Properties
Base64 String Decoder
Process Launcher
BWApplication [BusinessWorks Application] D:\tibco\tibcojre64\1.7.0\bin\javaw.exe (Aug 5, 2015 11:04:57 AM)
11:05:09.720 INFO [Job_Executor0] com.tibco.thor.frwk.Application - TIBCO-THOR-FRWK-300021: All Application dependencies are resolved for Application [MX_Parse_Render.application:1.0]
11:05:21.976 INFO [Thread-27] com.tibco.thor.frwk.Application - TIBCO-THOR-FRWK-300006: Started BW Application [MX_Parse_Render.application:1.0]

Mapping SWIFT MT and SWIFT MX Messages

The MT_MX_Mapping project shows how to use MT and MX activities to convert SWIFT MT messages to SWIFT MX messages, and to convert SWIFT MX messages to SWIFT MT messages.

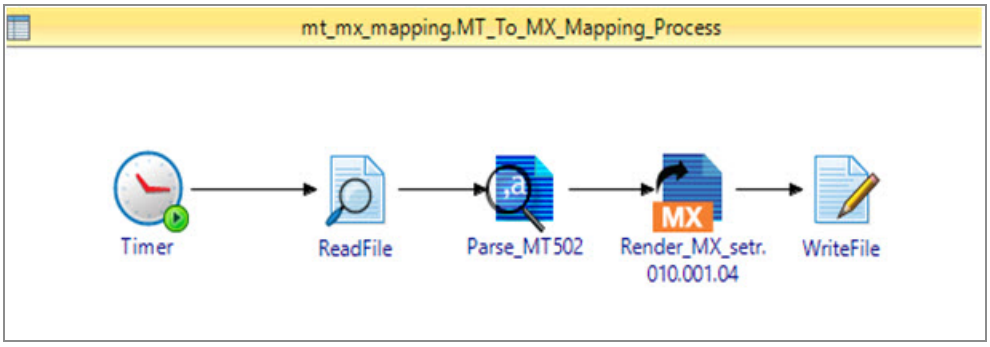
This project contains the following processes:

- MT_To_MX_Mapping_Process
- MX_To_MT_Mapping_Process

MT_To_MX_Mapping_Process

You can use the MT_To_MX_Mapping_Process to convert an MT message to an MX message.

The following figure describes the MT_To_MX_Mapping_Process:



The MT_To_MX_Mapping_Process performs the following operations:

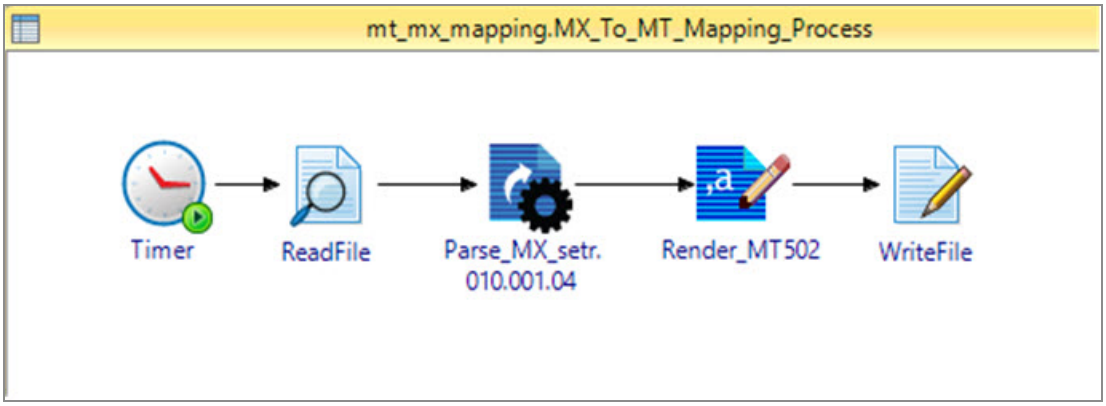
1. The ReadFile activity reads an MT message, and passes it to the Parse_MT502 activity.
2. The Parse_MT502 activity parses and validates the MT message, and passes it to the Render_MX_setr.010.001.04 activity.
3. The Render_MX_setr.010.001.04 activity validates the message, generates an MX message, and passes it to the WriteFile activity.
4. The WriteFile activity writes the MX message to the `setr.010.001.04_output.xml` file

in the `$Root/out` directory.

MX_To_MT_Mapping_Process

You can use the MX_To_MT_Mapping_Process to convert an MX message to an MT message.

The following figure describes the MX_To_MT_Mapping_Process:



The MX_To_MT_Mapping_Process performs the following operations:

1. The ReadFile activity reads an MX message, and passes it to the Parse_MX_ setr.010.001.04 activity.
2. The Parse_MX_setr.010.001.04 activity parses and validates the MX message, and passes it to the Render_MT502 activity.
3. The Render_MT502 activity validates the message, generates an MT message, and passes it to the WriteFile activity.
4. The WriteFile activity writes the MT message to the `MT502_output.fin` file in the `$Root/out` directory.

Setting Up the Project

Before running the project, you have to set it up.

Before you begin

You have to import the MT_MX_Mapping project before setting it up. For more information, see Importing Sample Projects.

Procedure

1. Expand the imported project in the Project Explorer view.
2. Expand the **Module Descriptors** resource, and then double-click **Module Properties**.
3. In the Module Properties panel, expand the **Example** folder, and then complete the following tasks:
    a. Set the value of the `Example_ROOT` parameter to the path where the project is located in your workspace. For example, `C:/workspace_SWIFT_example/MT_MX_Mapping`.
    b. From the menu bar, click **File > Save** to save the project.
    c. Download the required .xsd files that correspond to the MX+ message files in the `$ROOT/in` directory from the SWIFT User Guide. Then save the .xsd files in the *`TIBCO_HOME`*`/bw/palettes/swift/`*`version_number`*`/bin/xsd/`*`year`* directory.

Running the Project

After setting up the project, you can run it.

Procedure

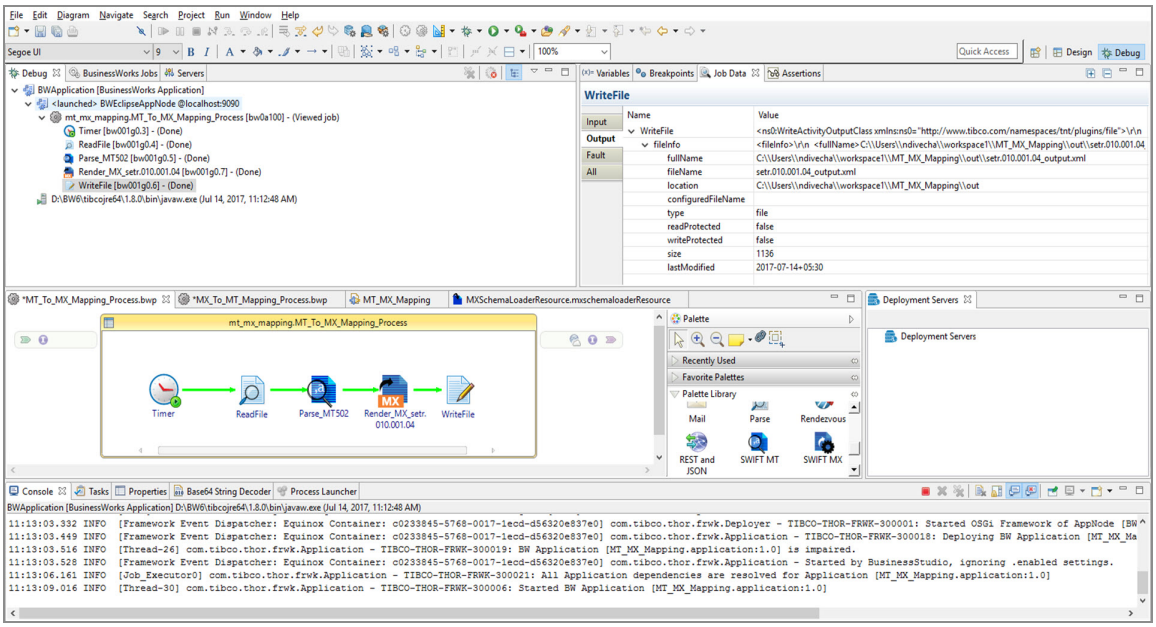
1. In the Project Explorer view, expand the **Module Descriptors** resource, and then double-click **Components**.
2. In the Components editor, click  and select the process that you want to run.
3. On the toolbar, click the  icon to save your changes.
4. From the menu bar, click **Run > Debug Configurations** to run the selected process.
5. In the Debug Configurations dialog, expand **BusinessWorks Application**, and then click **BWApplication**.
6. In the right panel, click the **Applications** tab, and then select the checkbox next to **MT_MX_Mapping.application**.
7. Click **Debug** to run the process.
8. Click the  icon to stop the process.

Result

After this process runs successfully, you can check the results.

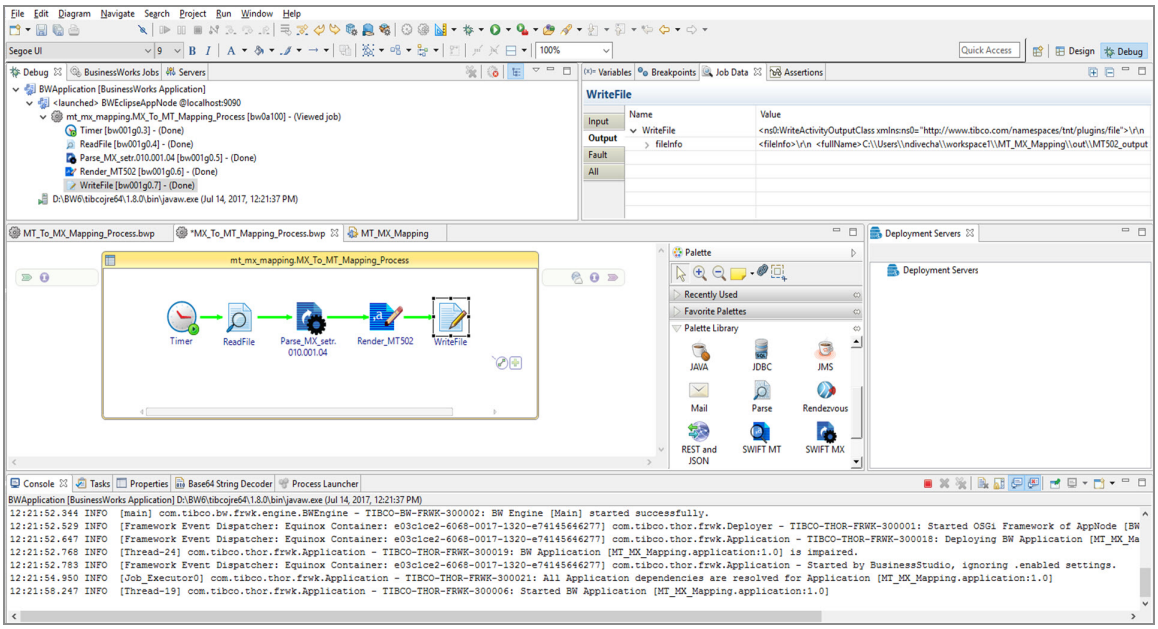
- Expected Result of the MT_To_MX_Mapping_Process

  The message is written to the `setr.010.001.04_output.xml` file in the `$Root/out` directory.



- Expected Result of the MX_To_MT_Mapping_Process

  The message is written to the `MT502_output.fin` file in the `$Root/out` directory.


File Edit Diagram Navigate Search Project Run Window Help
Debug
BusinessWorks Jobs
Servers
BWApplication [BusinessWorks Application]
<launched> BWEclipseAppNode @localhost:9090
mt_mx_mapping.MX_To_MT_Mapping_Process [bw0a100] - (Viewed job)
Timer [bw001g0.3] - (Done)
ReadFile [bw001g0.4] - (Done)
Parse_MX_setr.010.001.04 [bw001g0.5] - (Done)
Render_MT502 [bw001g0.6] - (Done)
WriteFile [bw001g0.7] - (Done)
Variables
Breakpoints
Job Data
Assertions
WriteFile
Input
Output
Fault
All
Name
Value
WriteFile
fileInfo
MT_To_MX_Mapping_Process.bwp
*MX_To_MT_Mapping_Process.bwp
MT_MX_Mapping
mt_mx_mapping.MX_To_MT_Mapping_Process
Timer
ReadFile
Parse_MX_setr. 010.001.04
Render_MT502
WriteFile
Palette
Recently Used
Favorite Palettes
Palette Library
JAVA
JDBC
JMS
Mail
Parse
Rendezvous
REST and JSON
SWIFT MT
SWIFT MX
Deployment Servers
Console
Tasks
Properties
Base64 String Decoder
Process Launcher
12:21:52.344 INFO [main] com.tibco.bw.frwk.engine.BWEngine - TIBCO-BW-FRWK-300002: BW Engine [Main] started successfully.
12:21:52.529 INFO [Framework Event Dispatcher: Equinox Container: e03c1ce2-6068-0017-1320-e74145646277] com.tibco.thor.frwk.Deployer - TIBCO-THOR-FRWK-300001: Started OSGi Framework of AppNode [BW
12:21:52.647 INFO [Framework Event Dispatcher: Equinox Container: e03c1ce2-6068-0017-1320-e74145646277] com.tibco.thor.frwk.Application - TIBCO-THOR-FRWK-300018: Deploying BW Application [MT_MX_Ma
12:21:52.768 INFO [Thread-24] com.tibco.thor.frwk.Application - TIBCO-THOR-FRWK-300019: BW Application [MT_MX_Mapping.application:1.0] is impaired.
12:21:52.783 INFO [Framework Event Dispatcher: Equinox Container: e03c1ce2-6068-0017-1320-e74145646277] com.tibco.thor.frwk.Application - Started by BusinessStudio, ignoring .enabled settings.
12:21:54.950 INFO [Job_Executor0] com.tibco.thor.frwk.Application - TIBCO-THOR-FRWK-300021: All Application dependencies are resolved for Application [MT_MX_Mapping.application:1.0]
12:21:58.247 INFO [Thread-19] com.tibco.thor.frwk.Application - TIBCO-THOR-FRWK-300006: Started BW Application [MT_MX_Mapping.application:1.0]

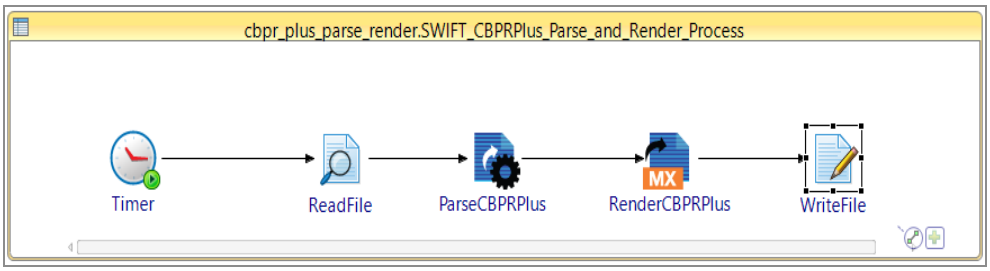
Using Parse SWIFT MX and Render SWIFT MX Activities for CBPR+ Messages

The CBPRPlus_Parse_Render project shows how to use the Parse SWIFT MX and the Render SWIFT MX activities to parse and render CBPR+ messages.

SWIFT_CBPRPlus_Parse_and_Render_Process

You can use the SWIFT_CBPRPlus_Parse_and_Render_Process to validate and parse the content in the incoming message, and to generate a new message.

The following figure describes the SWIFT_CBPRPlus_Parse_and_Render_Process:



The SWIFT_CBPRPlus_Parse_and_Render_Process performs the following operations:

1. The ReadFile activity reads CBPR+ message, and passes it to the ParseCBPRPlus activity.
2. The ParseCBPRPlus activity validates and parses the content of the message, and passes it to the RenderCBPRPlus activity.
3. The RenderCBPRPlus activity validates the content of the message, and generates a specific CBPR+ message, and passes it to the WriteFile activity.
4. The Write File activity writes an MX message to the CBPR_pacs_009_001_08_out.xml file in the $Example_Root/out directory.

Setting Up the Project

Before running the project, you have to set it up.

Before you begin

You have to import the CBPRPlus_Parse_Render project before setting it up. For more information, see Importing Sample Projects.

Procedure

1. Expand the imported project in the Project Explorer view.
2. Expand the **Module Descriptors** resource, and then double-click **Module Properties**.
3. In the Module Properties panel, expand the **Example** folder, and then complete the following tasks:
    a. Set the value of the `Example_ROOT` parameter to the path where the project is located in your workspace.

       For example, `C:/workspace_SWIFT_example/CBPRplus_Parse_Render`.

    b. From the menu bar, click **File > Save** to save the project.

    c. Download the required .xsd files that correspond to the CBPR+ message files in the `$Example_ROOT/in` directory from the SWIFT website, rename them. Now, save .xsd files in the *`TIBCO_HOME`*`/bw/palettes/swift/`*`version_number`*`/bin/xsd/`*`year`* directory. For more information renaming CBPR+ .xsd files, see the T*IBCO ActiveMatrix BusinessWorks™ Plug-in for SWIFT User Guide*.

Running the Project

After setting up the project, you can run it.

Procedure

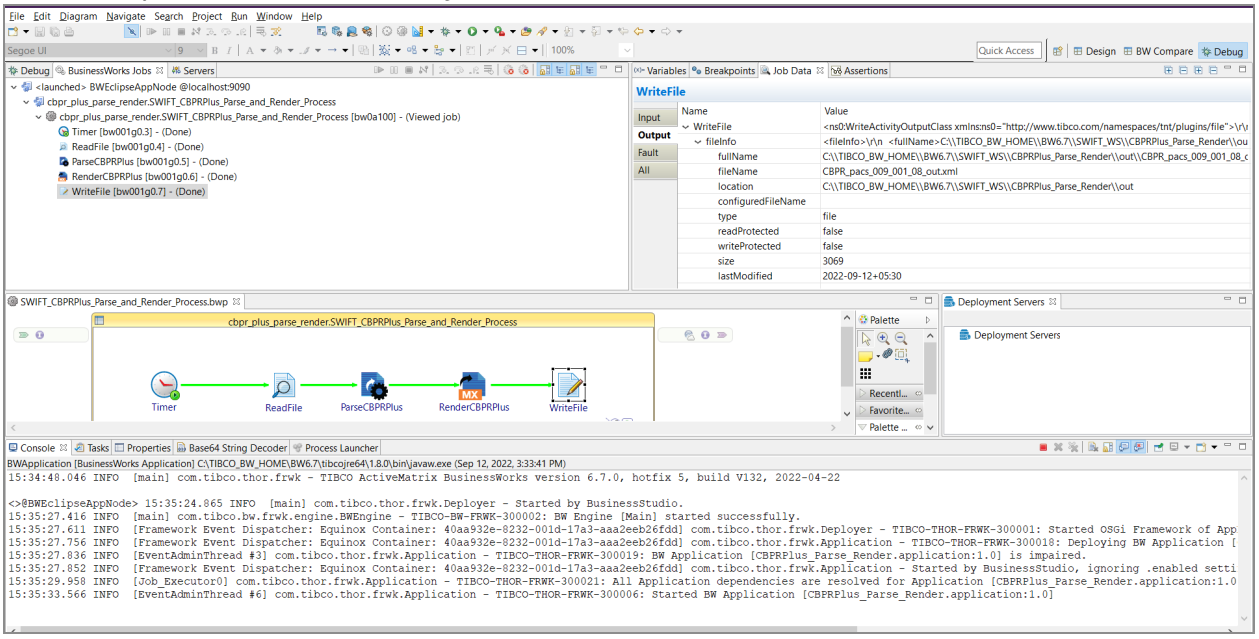
1. In the Project Explorer view, expand the **Module Descriptors** resource, and then double-click **Components**.
2. In the Components editor, click  and select the process that you want to run.
3. To save your changes, on the toolbar, click the  icon.
4. To run the selected process, from the menu bar, click **Run > Debug Configurations**.
5. In the Debug Configurations dialog, expand **BusinessWorks Application**, and then

click **BWApplication**.

6. In the right panel, click the **Applications** tab, and then select the checkbox next to **CBPRPlus_Parse_Render. application.**
7. Click **Debug** to run the process.
8. Click the  icon to stop the process.

Result

After this process runs successfully, a CBPR_pacs_009_001_08_out.xml file is generated in the *$Example_ROOT/out* directory.



TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the Product Documentation website, mainly in HTML and PDF formats.

The Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The documentation for this product is available on the TIBCO ActiveMatrix BusinessWorks™ Plug-in for SWIFT Product Documentation page.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our product Support website.
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the product Support website. If you do not have a username, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature

requests from within the TIBCO Ideas Portal. For a free registration, go to TIBCO Community.

Legal and Third-Party Notices

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file for the availability of a specific version of Cloud SG software on a specific operating system platform.