

TIBCO ActiveMatrix® Adapter for Files for Unix/Win (TIBCO Business Studio™) Examples

*Software Release 1.3
February 2016*

Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document contains confidential information that is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, Two-Second Advantage, TIBCO ActiveMatrix Adapter for Files for Unix/Win, TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio), TIBCO ActiveMatrix Adapter Framework, TIBCO ActiveMatrix BusinessWorks, TIBCO Adapter SDK, TIBCO Business Studio, TIBCO Designer, TIBCO Enterprise Message Service, TIBCO Hawk, TIBCO Rendezvous, and TIBCO Runtime Agent are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Enterprise Java Beans (EJB), Java Platform Enterprise Edition (Java EE), Java 2 Platform Enterprise Edition (J2EE), and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation in the U.S. and other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

THIS 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 THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

Copyright © 2010-2016 TIBCO Software Inc. All rights reserved.

TIBCO Software Inc. Confidential Information

Contents

TIBCO Documentation and Support Services	5
Examples Overview	6
Prerequisites of Running Examples	7
Required Software	7
Migrating Examples to TIBCO Business Studio	7
Importing Examples to TIBCO Business Studio	8
BaseRecord Project	9
Running the Example	9
Expected Result	9
MultiSchema Project	11
Running the Example	11
Expected Result	11
BusinessDocument Project	13
Running the Example	13
Expected Result	13
DateTime Project	15
Running the Example	15
Expected Result	15
EditableSchema Project	17
Running the Example	17
Expected Result	17
EOL Project	19
Running the Example	19
Expected Result	19
FA2BW Project	21
Running the Example	21
Expected Result	21
JmsQueue Project	23
Running the Example	23
Expected Result	23
JmsTopic Project	25
Running the Example	25
Expected Result	25
ManualSchema Project	26
Running the Example	26
Expected Result	26
MultiDelimiter Project	28
Running the Example	28

Expected Result	28
Positional Project	30
Running the Example	30
Expected Result	30
Promote Project	32
Running the Example	32
Expected Result	32
PublishDelay Project	34
Running the Example	34
Expected Result	34
SFT Project	35
FTP Example	35
Running the Example	35
Expected Result	35
MessageSelector Example	36
Running the Example	36
Expected Result	37
SimpleRecord Project	38
Running the Example	38
Expected Result	39
International Projects	40
Running the Example	40
Expected Result	41

TIBCO Documentation and Support Services

Documentation for this and other TIBCO products is available on the TIBCO Documentation site. This site is updated more frequently than any documentation that might be included with the product. To ensure that you are accessing the latest available help topics, please visit:

<https://docs.tibco.com>

Product-Specific Documentation

Documentation for TIBCO products is not bundled with the software. Instead, it is available on the TIBCO Documentation site. To directly access documentation for this product, double-click the following file:

`TIBCO_HOME/release_notes/TIB_adfilesbs_version_docinfo.html`

where `TIBCO_HOME` is the top-level directory in which TIBCO products are installed. On Windows, the default `TIBCO_HOME` is `C:\Program Files\tibco`. On UNIX systems, the default `TIBCO_HOME` is `/opt/tibco`.

The following documents for this product can be found on the TIBCO Documentation site:

- *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) Installation*
- *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide*
- *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) Examples*
- *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) Release Notes*

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

<http://www.tibco.com/services/support>

- If you already have a valid maintenance or support contract, visit this site:

<https://support.tibco.com>

Entry to this site requires a user name and password. If you do not have a user name, you can request one.

How to Join TIBCOCommunity

TIBCOCommunity is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCOCommunity offers forums, blogs, and access to a variety of resources. To register, go to the following web address:

<https://www.tibcommunity.com>

Examples Overview

TIBCO ActiveMatrix® Adapter for Files for Unix/Win (TIBCO Business Studio™) provides examples on how to configure and run TIBCO ActiveMatrix® Adapter for Files for Unix/Win in TIBCO Business Studio™.

After installing TIBCO ActiveMatrix Adapter for Files for Unix/Win, you can find the preconfigured examples located in the *TIBCO_HOME\adapter\adfiles\version_num\examples* directory. This directory contains the following three folders: **read**, **samples**, and **writer**. The **samples** folder contains the **BusinessWorks5** and **BusinessWorks6** folders.

- The **BusinessWorks5** folder contains examples created in TIBCO Designer™. If you want to run these examples in TIBCO Business Studio, you have to migrate the sample projects into TIBCO Business Studio first. For information about how to migrate sample projects, see [Migrating Examples to TIBCO Business Studio](#).
- The **BusinessWorks6** folder contains examples created in TIBCO Business Studio. Before running the examples, you have to import the compressed sample files into TIBCO Business Studio. For information about how to import sample projects, see [Importing Examples to TIBCO Business Studio](#).
- The **reader** and **writer** folders contain the directories used in the examples where input files exist and output files are placed. The **reader** folder is used by Publication Service. The **writer** folder is used by Subscription Service.

The **sample** folder contains executable files for Windows and Unix platforms. This guide describes how to use Adapter Launcher to run the sample projects in TIBCO Business Studio on a Microsoft Windows platform. Both the **BusinessWorks5** folder and the **BusinessWorks6** folder contain the following sample projects in the separate compressed folders:

- [BaseRecord Project](#)
- [MultiSchema Project](#)
- [BusinessDocument Project](#)
- [DateTime Project](#)
- [EditableSchema Project](#)
- [EOL Project](#)
- [FA2BW Project](#)
- [JmsQueue Project](#)
- [JmsTopic Project](#)
- [ManualSchema Project](#)
- [MultiDelimiter Project](#)
- [Positional Project](#)
- [Promote Project](#)
- [PublishDelay Project](#)
- [SFT Project](#)
- [SimpleRecord Project](#)
- [International Projects](#)


Prerequisites of Running Examples

Before running the examples, you have to install the required software and migrate or import the sample projects.

Required Software

When you use the adapter with TIBCO ActiveMatrix BusinessWorks™, you have to install the required software before running the examples.

The following list contains required software:

- TIBCO Rendezvous®
 - TIBCO Runtime Agent™
 - TIBCO ActiveMatrix BusinessWorks 6.x
 - TIBCO ActiveMatrix Adapter for Files for Unix/Win
 - TIBCO ActiveMatrix® Adapter Framework
 - TIBCO® Enterprise Administrator (if you are using deployment)
 - TIBCO Enterprise Message Service™ (if you are using the JMS transport)
- 
 - The TIBCO Enterprise Message Service server must be running and accessible to the machine on which the adapter is installed.
 - See the readme file for the supported versions of the required software.

Migrating Examples to TIBCO Business Studio

To run sample projects in the **BusinessWorks5** folder, you have to migrate the sample projects to TIBCO Business Studio first.

The example files are in the `TIBCO_HOME\adapter\adfiles\version_num\examples\samples\BusinessWorks5` directory.

Prerequisites

Ensure that you have extracted the compressed example files to a directory.

Procedure

1. Open TIBCO Business Studio.
2. Choose one of the following two ways to open the BusinessWorks Migration Tool dialog:
 - From the menu, click **Project > Migrate BW Projects**.
 - From the menu, click **File > Import**. In the Import wizard, select **Migrate BW Projects > Migrate BW Projects**, and then click **Next**.
3. In the BusinessWorks Migration Tool dialog, click **Browse** next to the **BusinessWorks 5 Projects Folder** field to select the extracted example file containing the example project that you want to migrate.
4. In the BusinessWorks 5 Project panel, click the project you want to migrate.
5. Click **Browse** next to the **Migrated Project Folder** field to select a directory to store the migrated project.
6. Click **Migrate** to start the migration process.

7. Click **Finish** after the migration is finished, and then click **Close**.

Importing Examples to TIBCO Business Studio

To run the sample projects in the **BusinessWorks6** folder, you have to import the sample projects to TIBCO Business Studio first.

The example files are in the *TIBCO_HOME\adapter\adfiles\version_num\examples\samples\BusinessWorks6* directory.

Procedure

1. From the menu, click **File > Import** to open the Import wizard.
2. In the Import wizard, select **General > Existing Studio Projects into Workspace**, and then click **Next**.
3. In the Import Projects dialog, click **Select archive file**.
4. Click **Browse** to select the BusinessWorks6 compressed example file that you want to import.
5. Click **Finish**.

BaseRecord Project

The BaseRecord project contains two adapter configurations: ContainerReader.adfilesmodel and ContainerWriter.adfilesmodel. The sample project shows how the adapter publishes container records.


In ContainerReader.adfilesmodel, Publication Service parses the `container.txt` file, processes the lines into objects and publishes them. The read schema stores the base record Employee. The Department record contains the Employee record as the third field.

In ContainerWriter.adfilesmodel, Subscription Service receives the objects, formats them according to the write schema, and writes the output to the `outcontainer.txt` file. The write schema is configured to write all file records to a separate line. The Department record is no longer a container record and does not have the Employee record as the third field. Instead, the Employee record is now a child record. Therefore, all the records are written to a different line in the output file.

Running the Example

You can use Adapter Launcher to run the example in TIBCO Business Studio.

Procedure

1. Open the sample project in TIBCO Business Studio.
2. From the menu, click **Run > Run Configurations**.
3. In the Run Configurations dialog, click **New launch configuration** .
4. In the Adapter Configuration panel, click **Browse** and select ContainerWriter.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
5. Click **Apply**, and then click **Run**.
6. Repeat [Step 2](#) and [Step 3](#), and in the Adapter Configuration panel, click **Browse** and select ContainerReader.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
7. Click **Apply**, and then click **Run**.

Expected Result

Publication Service parses the input file, processes it into objects, and publishes it. Subscription Service receives the objects, and formats them according to the write schema. The Employee record is a child record of the parent record Department, and all the records are written to a different line in the output file.

You can view the example results in the `TIBCO_HOME\adapter\adfiles\version_num\examples\writer\output` directory. The input file and output results are shown as follows:

Input File	Output File
Department, Engineering, Employee, ID0045, Mary,Employee, ID0056, John, Employee, ID0245, Asha Item, Stapler, 60, \$8 Department, Accounts, Employee, ID0115, Vikram,Employee, ID0144, Sundar Item, NoteBook, 48, \$5	Department;Engineering Employee;ID0045;Mary Employee;ID0056;John Employee;ID0245;Asha Item;Stapler;60;\$8 Department;Accounts Employee;ID0115;Vikram Employee;ID0144;Sundar Item;NoteBook;48;\$5



The results in your output file might be in reverse order. You can correct this using the `adfiles.toggleChildRecordsOrdering` property. For more information about this property, see *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide*.

MultiSchema Project

The MultiSchema project contains two adapter configurations: ContainerReader2.adfilesmodel and ContainerWriter2.adfilesmodel. The sample project shows how the adapter publishes container records.

This sample project has a more complex schema than the BaseRecord project.


In ContainerReader2.adfilesbomodel, Publication Service parses the `container1.txt` file, processes the lines into objects and publishes them. Publication Service has two read schemas: the Order record contains the Item and Payment records; the Process record consists of the BackOrder and Shipped records, which contain the Item1 record.

In ContainerWriter2.adfilesbomodel, Subscription Service receives the objects, formats them according to the write schemas, and writes the output to the `outcontainer2.txt` file. Subscription Service has two write schemas: the Order record is set to write the Order record as it is, with the Item record still contained in it. However, the Process record is set to write all the file records to a different line in the output file. The BackOrder and Shipped records are no longer container records and do not have the Item1 record. Item1 record is a write child record and the Process record is at the parent level.

Running the Example

You can use Adapter Launcher to run the example in TIBCO Business Studio.

Procedure

1. Open the sample project in TIBCO Business Studio.
2. From the menu, click **Run > Run Configurations**.
3. In the Run Configurations dialog, click **New launch configuration** .
4. In the Adapter Configuration panel, click **Browse** and select ContainerWriter2.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
5. Click **Apply**, and then click **Run**.
6. Repeat [Step 2](#) and [Step 3](#), and in the Adapter Configuration panel, click **Browse** and select ContainerReader2.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
7. Click **Apply**, and then click **Run**.

Expected Result

Publication Service parses a `container1.txt` input file, processes the lines into objects and publishes them. Subscription Service receives the objects, formats them according to the write schemas, and writes the output to the `outcontainer2.txt` file.

You can view the example results in the `TIBCO_HOME\adapter\adfiles\version_num\examples\writer\output` directory. The input file and output results are shown as follows:

Input File	Output File
<p>Order, 101, 3/1/2003, Item, Pencil, 6, Item, Eraser, 10, Item, Ruler, 12, Payment# VISA# 1111111111111111# \$20# Payment# Check# 10001# \$25 Customer, C001, Joe Smith Address; 101 California Street; San Francisco, CA Order, 102, 3/2/2003, Payment# Check# 20001# \$75 Customer, C002, Heather Hope Address; 567 University Ave; Stanford, CA Order, 103, 3/3/2003, Item, Marker, 8, Item, Eraser, 5, Item, Staples, 12, Item, Envelope, 24 Customer, C003, Jerry Jones Address; 123 Hillview Court; Palo Alto, CA Process, 101, 3/4/2003 Shipped, Item, Pencil, 6, Item, Eraser, 10 Address; 101 California Street; San Francisco, CA BackOrder, Item, Ruler, 12 Process, 102, 3/5/2003 Shipped, Item, Eraser, 5, Item, Envelop, 24 Address; 123 Hillview Court; Palo Alto, CA BackOrder, Item, Marker, 8, Item, Staples, 12</p>	<p>Order;101;3/1/2003;Item;Pencil;6;Item ;Eraser;10;Item;Ruler;12 Payment;VISA;1111111111111111;\$20 Payment;Check;10001;\$25 Customer;C001;Joe Smith Address;101 California Street;San Francisco, CA Order;102;3/2/2003; Payment;Check;20001;\$75 Customer;C002;Heather Hope Address;567 University Ave;Stanford, CA Order;103;3/3/2003;Item;Marker;8;Item ;Eraser;5;Item;Staples;12;Item;Envelo pe;24 Customer;C003;Jerry Jones Address;123 Hillview Court;Palo Alto, CA Process;101;3/4/2003 Shipped Item;Pencil;6 Item;Eraser;10 BackOrder Item;Ruler;12 Address;101 California Street;San Francisco, CA Process;102;3/5/2003 Shipped Item;Eraser;5 Item;Envelop;24 BackOrder Item;Marker;8 Item;Staples;12 Address;123 Hillview Court;Palo Alto, CA</p>

BusinessDocument Project

The BusinessDocument project contains two adapter configurations: `transferReader.adfilesmodel` and `transferWriter.adfilesmodel`. The sample project shows how the adapter groups objects together and transfers each grouped object as a single message.

Grouping objects helps to speed the transfer process. A business object acts as a container for objects. By default, each object is published separately. When using a business document, objects are placed in a container, and the container is published in a single message. The number of objects in the business document is set in the **Grouping Factor** field. When setting the grouping factor to 2, Publication Service places two objects in the business document, and then publishes the business document.

In `transferReader.adfilesmodel`, Publication Service parses the `oneMegaByteFile.txt` file and publishes each line in the file as a separate message. Publication Service configuration and processing parameters are defined in a similar way as those in other examples. Unlike other examples, in the **Advanced** tab, the **Data Format** field is **Business Document**, and the **Grouping Factor** field is 2. The **Business Document Name** field is set to **transferSample**.

In `transferWriter.adfilesmodel`, Subscription Service receives each message, formats it by changing the delimiter character used between fields and changing the type assigned to some fields, and appends it to the `outtransfer.txt` file. In the **Configuration** tab, the **Wip Creation Mode** field is set to **Append messages to file, close on Business Document Lot End**. The subscriber waits until all objects placed in the business document are received before closing the output file.


The write schema assigned to Subscription Service defines how lines are formatted before being written to the output file. A semicolon is used as the delimiter character. For the `Quantity`, `Price` and `field4` fields, a positive sign (+ character) is prepended. Scientific notation is enabled for the `Price` and `field4` fields.

For example, the line in the input file (`Item, GigaWidget, 60, 75, 00000`) is changed as (`Item;GigaWidget;+60;+7.500000E+001;+0.000000E+000`) in the output file.

Running the Example

You can use Adapter Launcher to run the example in TIBCO Business Studio.

Procedure

1. Open the sample project in TIBCO Business Studio.
2. From the menu, click **Run > Run Configurations**.
3. In the Run Configurations dialog, click **New launch configuration** .
4. In the Adapter Configuration panel, click **Browse** and select `transferWriter.adfilesmodel` for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
5. Click **Apply**, and then click **Run**.
6. Repeat [Step 2](#) and [Step 3](#), and in the Adapter Configuration panel, click **Browse** and select `transferReader.adfilesmodel` for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
7. Click **Apply**, and then click **Run**.

Expected Result

Publication Service parses the `oneMegaByteFile.txt` file and publishes each line in the file as a separate message. Subscription Service receives each message, formats it by changing the delimiter

character used between fields and changing the type assigned to some fields, and appends it to the `outtransfer.txt` file.

You can view the example results in the `TIBCO_HOME\adapter\adfiles\version_num\examples\writer\output` directory. The input file and output results are shown as follows:

Input File	Output File
Item, GigaWidget, 60, 75, 00000	Item;GigaWidget;+60;+7.500000E
Item, MegaBucket, 48, 125, 00001	+001;+0.000000
Item, RtClopper, 40, 100, 00002	E+000
Item, LtClopper, 50, 100, 00003	Item;MegaBucket;+48;+1.250000E
...	+002;+1.000000
	E+000
	Item;RtClopper;+40;+1.000000E
	+002;+2.000000E
	+000
	Item;LtClopper;+50;+1.000000E
	+002;+3.000000E
	+000

DateTime Project

The DateTime project contains three adapter configurations: DateTimeReader.adfilesmodel, DateTimeWriter.adfilesmodel, and DateTimeWriter2.adfilesmodel. The sample project shows how the adapter handles date and time.

In DateTimeReader.adfilesmodel, Publication Service parses the `time.txt` file which has the second field set to the long date and time format. Publication Service is configured to publish a record that has different date formats.


In DateTimeWriter.adfilesmodel, Subscription Service receives the objects, formats and writes them to the `outtime.txt` output file. Subscription Service writes these dates to the file in the long date format, irrespective of the formats they are published in. This is the default behavior of the adapter.

In DateTimeWriter2.adfilesmodel, Subscription Service receives the objects, formats and writes them to the `outtime2.txt` output file. The write schema of Subscription Service is configured to generate date and time in exactly the same formats in which they are published.

Running the Example

You can use Adapter Launcher to run the example in TIBCO Business Studio.

Procedure

1. Open the sample project in TIBCO Business Studio.
2. From the menu, click **Run > Run Configurations**.
3. In the Run Configurations dialog, click **New launch configuration** .
4. In the Adapter Configuration panel, click **Browse** and select DateTimeWriter.adfilesmodel or DateTimeWriter2.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
5. Click **Apply**, and then click **Run**.
6. Repeat [Step 2](#) and [Step 3](#), and in the Adapter Configuration panel, click **Browse** and select DateTimeReader.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
7. Click **Apply**, and then click **Run**.

Expected Result

Publication Service parses the `time.txt` file and publishes the objects. Subscription Service receives the objects, formats and writes them to the `outtime.txt` and `outtime2.txt` files according to their write schemas.

You can view the example results in the `TIBCO_HOME\adapter\adfiles\version_num\examples\writer\output` directory. The input file and output results are shown as follows:

File Name	File Content
time.txt	<pre> time; Apr 22, 2000 8:15:23 AM IST; Apr 22, 2002 8:15:23 AM; 03/03/03 10:14:26 AD;Friday, December 01, 2002 11:15:59 AM PST;09/09/01 12:00 AM time; December 01, 1999 10:15:59 AM PST; December 01, 2000 09:15:59 AM; 02/26/02 8:16:22 AD;Friday, December 01, 2002 11:15:59 AM PST;09/09/01 12:00 AM </pre>
outtime.txt	<pre> time;April 22, 2000 11:45:23 AM GMT+05:30;April 22, 2002 8:15:23 AM GMT+05:30;January 3, 2003 10:14:26 AM GMT+05:30;December 2, 2002 12:45:59 AM GMT+05:30;September 9, 2001 12:00:00 AM GMT+05:30 time;December 1, 1999 11:45:59 PM GMT+05:30;December 1, 2000 9:15:59 AM GMT+05:30;January 26, 2002 8:16:22 AM GMT+05:30;December 2, 2002 12:45:59 AM GMT+05:30;September 9, 2001 12:00:00 AM GMT+05:30 </pre>
outtime2.txt	<pre> time;April 22, 2000 11:45:23 AM GMT+05:30;Apr 22, 2002 8:15:23 AM;14/03/03 10:14:26 AD;Monday, December 2, 2002 12:45:59 AM GMT+05:30;9/9/01 12:00 AM time;December 1, 1999 11:45:59 PM GMT+05:30;Dec 1, 2000 9:15:59 AM;16/26/02 8:16:22 AD;Monday, December 2, 2002 12:45:59 AM GMT+05:30;9/9/01 12:00 AM </pre>

EditableSchema Project

The EditableSchema project contains two adapter configurations:

EditableWSchemaReader.adfilesmodel and EditableWSchemaWriter.adfilesmodel. The sample project shows how the adapter adds fields to a line.


In EditableWSchemaReader.adfilesmodel, Publication Service parses the `editableWSchema.txt` file. Three positional file records (Identity, Item, and Cost) are defined to parse the type of lines found in the `editableWSchema.txt` file. Publication Service creates an object for each purchase order, item and cost, and publishes each object as a single message.

In EditableWSchemaWriter.adfilesmodel, Subscription Service receives the objects, formats them with an additional field that has `DefaultID` assigned to it, and writes them to the `outeditableWSchema.txt` file. The write schema is edited to include an extra field to the Cost record. Therefore, when Subscription Service writes the objects to an `outeditablewrite.txt` file, it adds a field to the Cost record, and sets the default value to `DefaultCost`.

Running the Example

You can use Adapter Launcher to run the example in TIBCO Business Studio.

Procedure

1. Open the sample project in TIBCO Business Studio.
2. From the menu, click **Run > Run Configurations**.
3. In the Run Configurations dialog, click **New launch configuration** .
4. In the Adapter Configuration panel, click **Browse** and select `EditableWSchemaWriter.adfilesmodel` for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
5. Click **Apply**, and then click **Run**.
6. Repeat [Step 2](#) and [Step 3](#), and in the Adapter Configuration panel, click **Browse** and select `EditableWSchemaReader.adfilesmodel` for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
7. Click **Apply**, and then click **Run**.

Expected Result

Publication Service parses the `editableWSchema.txt` file, creates an object for each purchase and publishes each object as a single message. Subscription Service receives the objects, formats them with an additional field and writes them to the `outeditableWSchema.txt` file.

You can view the example results in the `TIBCO_HOME\adapter\adfiles\version_num\examples\writer\output` directory. Fields in the input file are separated by spaces. Fields in the output files are formatted with semicolons. The input file and output results are shown as follows:

Input File	Output File
D001 21/12/02 10:20 AM Purchase order for Stationary \$23 10 STATIONARY PARKER_PEN_MODEL4 231123 COST 8 \$23 D014 16/07/01 12:42 PM Purchase order for Books \$12 4 Books JAVA in 21 days 432211 COST 6 \$12	D001;21/12/02;10:20 AM;DefaultID*****;Purchase order for Stationary;\$23;+1.0000E+001 STATIONARY;PARKER_PEN_MODEL4;231123; (Default) COST;8;\$23; DefaultCost!!!! D014;16/07/01;12:42 PM;DefaultID*****;Purchase order for Books;\$12;+4.0000E+000 Books;JAVA in 21 days;432211; (Default) COST;6;\$12; DefaultCost!!!!

EOL Project

The EOL project contains two adapter configurations: EOLReader.adfilesmodel and EOLWriter.adfilesmodel. The sample project shows how to specify custom end of line characters for file records.


In EOLReader.adfilesmodel, Publication Service parses the `eo1.txt` file and defines a new line when it encounters an at sign (@). The custom end of line character in this example is the at sign (@) as specified in the `eo1.txt` file. The input file uses the at sign (@) to denote the end of line. These lines are processed into objects and published, each as a single message.

In EOLWriter.adfilesmodel, Subscription Service receives the objects, and writes each of them to a new line in the `outeo1.txt` file. Subscription Service is configured with a user-defined end of line character, namely the caret (^).

Running the Example

You can use Adapter Launcher to run the example in TIBCO Business Studio.

Procedure

1. Open the sample project in TIBCO Business Studio.
2. From the menu, click **Run > Run Configurations**.
3. In the Run Configurations dialog, click **New launch configuration** .
4. In the Adapter Configuration panel, click **Browse** and select EOLWriter.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
5. Click **Apply**, and then click **Run**.
6. Repeat [Step 2](#) and [Step 3](#), and in the Adapter Configuration panel, click **Browse** and select EOLReader.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
7. Click **Apply**, and then click **Run**.

Expected Result

Publication Service parses the `eo1.txt` file and defines a new line when it encounters an at sign (@). Subscription Service receives the objects, and writes each of them to a new line with the caret (^) to the `outeo1.txt` file.

You can view the example results in the `TIBCO_HOME\adapter\adfiles\version_num\examples\writer\output` directory. The input file and output results are shown as follows:

Input File	Output File
<pre> Item, SA0002, STATIONARY@ Details, PEN, PARKER Model 2, SAN JOSE@ Cost, \$60, 4@ Item, VA0001, HARDWARE@Details, CD DRIVE, ACER CDR20, SEOUL@Cost, \$300, 4@ Item, SB0007, STATIONARY@ Details, Marker, Blue Color 4, NEWYORK @ Cost, \$20, 3 </pre>	<pre> Item;SA0002;STATIONARY^Details;PEN;PARK ER Model 2;SAN JOSE^Cost;\$60;4^Item;VA0001;HARDWARE^De tails;CD DRIVE;ACER CDR20;SEOUL^Cost;\$300;4^Item;SB0007;STA TIONARY^Details;Marker;Blue Color 4;NEWYORK^Cost;\$20;3^ </pre>

FA2BW Project

The FA2BW project contains two adapter configurations and an ActiveMatrix BusinessWorks process: FAReader.adfilesmodel, FAWriter.adfilesmodel and Process.bwp. The sample project shows how to use the adapter services within TIBCO ActiveMatrix BusinessWorks processes to transfer records.

In FAReader.adfilesmodel, Publication Service reads the `delimited.txt` file and publishes the message.


Process.bwp subscribes to the message, completes a field-to-field mapping, and transfers the message.

In FAWriter.adfilesmodel, Subscription Service subscribes to the republished message and writes it to the `outdelimited.txt` file.

Running the Example

You can use Adapter Launcher to run the example in TIBCO Business Studio.

Procedure

1. Open the sample project in TIBCO Business Studio.
2. From the menu, click **Run > Run Configurations**.
3. In the Run Configurations dialog, click **New launch configuration** .
4. In the Adapter Configuration panel, click **Browse** and select FAWriter.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
5. Click **Apply**, and then click **Run**.
6. In the Run Configurations window, click **BusinessWorks Application > BWApplication**. Select FA2BW.application, click **Apply**, and then click **Run**.
7. Repeat [Step 2](#) and [Step 3](#), and in the Adapter Configuration panel, click **Browse** and select FAReader.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
8. Click **Apply**, and then click **Run**.

Expected Result

Publication Service reads the `delimited.txt` file and publishes the message. Subscription Service subscribes to the republished message and writes it to the `outdelimited.txt` file.

You can view the example results in the `TIBCO_HOME\adapter\adfiles\version_num\examples\writer\output` directory. Records in the input file are separated by commas. Records in the output file are formatted with tabs and semicolons as separators. The input file and output results are shown as follows:

Input File	Output File
Order, ID41678, 20May2000 Item, GigaWidget, 60, \$75 Item, MegaBucket, 48, \$125 Customer, Hopkins Associates, ID26490 Order, ID41680, 20May2000 Item, Rt.Clopper, 40, \$100 Item, Lt.Clopper, 50, \$100 Customer, Jersey WebInovaters, ID46786	Order;ID41678;20May2000 Item;GigaWidget;+6.000000E+001;\$75 Item;MegaBucket;+4.800000E+001;\$125 Customer Hopkins Associates ID26490 Order;ID41680;20May2000 Item;Rt.Clopper;+4.000000E+001;\$100 Item;Lt.Clopper;+5.000000E+001;\$100 Customer Jersey WebInovaters ID46786

JmsQueue Project

The JmsQueue project contains two adapter configurations: JMSQueueReader.adfilesmodel and JMSQueueWriter.adfilesmodel. The sample project shows how the adapter uses the JMS queue transport to send messages.


In JMSQueueReader.adfilesmodel, Publication Service parses each line in the `jmsqueue.txt` file into objects. Each object is published as a message using the JMS queue transport to the `jmsq` destination.

In JMSQueueWriter.adfilesmodel, Subscription Service subscribes to messages using the JMS queue transport, and writes them to the `outjmsqueue.txt` file. It listens on the `jmsq` destination. The **Connection Factory** field is **QueueConnectionFactory**, and the **Delivery Mode** field is **Durable**. Therefore, the messages are held in the queue by the JMS server until Subscription Service consumes them.

Running the Example

You can use Adapter Launcher to run the example in TIBCO Business Studio.

Procedure

1. Start the JMS Server.
2. Open the sample project in TIBCO Business Studio.
3. From the menu, click **Run > Run Configurations**.
4. In the Run Configurations dialog, click **New launch configuration** .
5. In the Adapter Configuration panel, click **Browse** and select JMSQueueWriter.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
6. Click **Apply**, and then click **Run**.
7. Repeat [Step 3](#) and [Step 4](#), and in the Adapter Configuration panel, click **Browse** and select JMSQueueReader.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
8. Click **Apply**, and then click **Run**.

Expected Result

Publication Service parses the `jmsqueue.txt` file and publishes the objects. Subscription Service receives the objects, formats and writes them to the `outjmsqueue.txt` file.

You can view the example results in the `TIBCO_HOME\adapter\adfiles\version_num\examples\writer\output` directory. The input file and output results are shown as follows:

Input File	Output File
Employee, EMP00012 Address, 1035 Aster Avenue, Sunnyvale, CA, USA,94086 Phone, LandLine, 408-557-0420 Personal, 221-23-2342, Male, Married Name, Tiru, Ananth Employee, EMP00001 Address, 1229 Oak Knoll Drive, San Jose, CA, USA,95129 Phone, Cell, 408-910-5519 Personal, 600-23-2342, Female, Married Name, Jones, Catherine Employee, EMP00002 Address, 1688 Yucatan Way, Fallbrook, CA, USA,92028 Phone, LandLine, 760-207-0762 Personal, 221-23-2342, Male, Single Name, Mann, Rod	Employee;EMP00012 Address;1035 Aster Avenue;Sunnyvale;CA;USA;94086 Phone;LandLine;408-557-0420 Personal;221-23-2342;Male;Married Name;Tiru;Ananth Employee;EMP00001 Address;1229 Oak Knoll Drive;San Jose;CA;USA;95129 Phone;Cell;408-910-5519 Personal;600-23-2342;Female;Married Name;Jones;Catherine Employee;EMP00002 Address;1688 Yucatan Way;Fallbrook;CA;USA;92028 Phone;LandLine;760-207-0762 Personal;221-23-2342;Male;Single Name;Mann;Rod

JmsTopic Project

The JmsTopic project contains two adapter configurations: JMSTopicReader.adfilesmodel and JMSTopicWriter.adfilesmodel. The sample project shows how the adapter uses the JMS topic transport to send messages.


In JMSTopicReader.adfilesmodel, Publication Service parses each line in the `jmstopic.txt` file into objects. Each object is published as a message using the JMS topic transport to the `jmstopic` destination. The delivery mode is **Persistent**.

In JMSTopicWriter.adfilesmodel, Subscription Service subscribes to messages using JMS topic transport and writes them to the `outjmstopic.txt` file. Subscription Service listens on the `jmstopic` subject. The delivery mode is **Durable**.

Running the Example

You can use Adapter Launcher to run the example in TIBCO Business Studio.

Procedure

1. Start the JMS Server.
2. Open the sample project in TIBCO Business Studio.
3. From the menu, click **Run > Run Configurations**.
4. In the Run Configurations dialog, click **New launch configuration** .
5. In the Adapter Configuration panel, click **Browse** and select JMSTopicWriter.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
6. Click **Apply**, and then click **Run**.
7. Repeat [Step 3](#) and [Step 4](#), and in the Adapter Configuration panel, click **Browse** and select JMSTopicReader.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
8. Click **Apply**, and then click **Run**.

Expected Result

Publication Service parses the `jmstopic.txt` file and publishes the objects. Subscription Service receives the objects, formats and writes them to the `outjmstopic.txt` file.

You can view the example results in the `TIBCO_HOME\adapter\adfiles\version_num\examples\writer\output` directory. The input file and output results are shown as follows:

Input File	Output File
HEADER, MANOJ, BAJPAI, 650-208-5277, manoj@newyorkmax.com DETAILS, INSURANCE, 3, 5%, 3%, NEWYORKMAX HEADER, KRISHNA, SRIDEVI, 408-992-0234, skrishna@amazon.com DETAILS, BOOKS, 4, 10%, 2.2%, AMAZON	HEADER;MANOJ;BAJPAI; 650-208-5277;manoj@ne wyorkmax.com DETAILS;INSURANCE;+3;5%;3%;NEWYORKMAX HEADER;KRISHNA;SRIDEVI; 408-992-0234;skris hna@amazon.com DETAILS;BOOKS;+4;10%;2.2%;AMAZON

ManualSchema Project

The ManualSchema project contains a business process and an adapter configuration: TestDelimitedWriterManual.bwp and delimitedWriterManual.adfilesmodel. The sample project shows how the adapter transfers records using a manual write schema.

In this example, the TestDelimitedWriterManual.bwp process sends the specified input data in a series of messages on the delimitedWriterManualWithBWProcess subject.

In delimitedWriterManual.adfilesmodel, Subscription Service listens to the subject, receives all the messages, and then formats and writes them to an output file located in a working directory. One minute after opening the file, the service moves the file to its output directory, and names the file as delimitedManual.txt.

The WriteShcema(Manual) write schema in this example is created directly without relying on any read schema. You have to manually create write file records for the write schema.


Running the Example

You can use Adapter Launcher to run the example in TIBCO Business Studio.



If you migrate project in the **BusinessWorks5** folder, and the ActiveMatrix BusinessWorks 5 project has a Start activity, you have to add a Timer or File Poller activity to initiate the migrated process.

Procedure

1. Open the sample project in TIBCO Business Studio.
2. Create a new process, add a Timer or File Poller activity, and then add a CallProcess activity. Select the TestDelimitedWriterManual.bwp process for the CallProcess activity.
 - If you migrate the project from the **BusinessWorks5** folder, you have to complete this step.
 - If you import the project from the **BusinessWorks6** folder, you can skip this step.
3. From the menu, click **Run > Run Configurations**.
4. In the Run Configurations dialog, click **New launch configuration** .
5. In the Adapter Configuration panel, click **Browse** and select delimitedWriterManual.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, D:\temp.
6. Click **Apply**, and then click **Run**.
7. In the Run Configurations window, click **BusinessWorks Application > BWApplication**. Select ManualSchema.application, click **Apply**, and then click **Run**.

Expected Result

After the process starts, the adapter publisher sends the input data in Rendezvous messages. The adapter subscriber receives these messages and writes the data to an output file located in the working directory.

The data in the **Input** tab in the business process is successfully written to an output file. The input data and output results are shown as follows:

Input File	Output File
<pre>Order field_type "Order" Name "ID41678" Date "20May2000" Item_caret_child item field_type "Item" Name "GigaWidget" Quantity 60 Price "\$75" Customer_caret_child item field_type "Customer" Name "Hopkins Associates" ID "ID26490"</pre>	<pre>Order;ID41678;20May2000 Item;GigaWidget;+6.000000E+001;\$75 Customer;Hopkins Associates;ID26490</pre>

MultiDelimiter Project

The MultiDelimiter project contains two adapter configurations: MultiDelimReader.adfilesmodel and MultiDelimWriter.adfilesmodel. The sample project shows how the adapter handles files with multiple delimiters.

In MultiDelimReader.adfilesmodel, Publication Service parses the `multidelim.txt` file and publishes each line with a pair of commas as multiple delimiters. To enable Publication Service to parse the `multidelim.txt` file, the read schema defines three file records: Customer, Business and Contact. The Customer record is configured with a comma as the delimiter between fields. The Business record is configured with two percentage characters (%%) and the Contact record is configured with two hyphens and two asterisks (-*-) as the delimiters between each field.


In MultiDelimWriter.adfilesmodel, Subscription Service receives each message, formats it and writes it to the `outmultidelim.txt` file using a semi colon as a single delimiter. Subscription Service is configured with a single delimiter character between fields.

For example, the line in the input file(`Business%% B0001%% Application Integration%% REUTERS`) is changed as (`Business;B0001;Application Integration;REUTERS`) in the output file.

Running the Example

You can use Adapter Launcher to run the example in TIBCO Business Studio.

Procedure

1. Open the sample project in TIBCO Business Studio.
2. From the menu, click **Run > Run Configurations**.
3. In the Run Configurations dialog, click **New launch configuration** .
4. In the Adapter Configuration panel, click **Browse** and select MultiDelimWriter.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
5. Click **Apply**, and then click **Run**.
6. Repeat [Step 2](#) and [Step 3](#), and in the Adapter Configuration panel, click **Browse** and select MultiDelimReader.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
7. Click **Apply**, and then click **Run**.

Expected Result

Publication Service parses the `multidelim.txt` file and publishes each line with a pair of commas as multiple delimiters. Subscription Service receives each message, formats it and writes it to the `outmultidelim.txt` file using a semi-colon as a single delimiter.

You can view the example results in the `TIBCO_HOME\adapter\adfiles\version_num\examples\writer\output` directory. The input file and output results are shown as follows:

Input File	Output File
Customer, C0002, RDWY, 006998397 Business%% B0001%% Application Integration%% REUTERS Contact-**- 650-799-9234-**- contactus@tibco.com-**- www.tibco.com Customer, C0021, WSWY, 006244123 Business%%B0003%%Courier Service%%ROADWAY EXPRESS Contact-**-732-750-9608-**-service@road wayexpress.com-**-www.roadway.com Customer, C0038, SRAC, 006502837 Business%%B0005%% Car Rental%%Speedy Contact-**-258-123-4567-**-service@spee dy.com-**- www.speedy.com	Customer;C0002;RDWY;006998397 Business;B0001;Application Integration;REUTERS Contact;650-799-9234;contactus@tibco.co m;www.tibco.com Customer;C0021;WSWY;006244123 Business;B0003;Courier Service;ROADWAY EXPRESS Contact;732-750-9608;service@roadwayexp ress.com;www.roadway.com Customer;C0038;SRAC;006502837 Business;B0005;Car Rental;Speedy Contact;258-123-4567;service@speedy.com ;www.speedy.com

Positional Project

The Positional project contains two adapter configurations: `positionalReader.adfilesmodel` and `positionalWriter.adfilesmodel`. The sample project shows how the adapter transfers records to new files.

In `positionalReader.adfilesmodel`, Publication Service parses the `positional.txt` file and creates an object for each order ID, item and customer that it finds in the input file. Publication Service sends each object in a separate message. After processing all lines in the input file, Publication Service exits.


In `positionalWriter.adfilesmodel`, Subscription Service receives the objects, formats and writes each object to a separate file in the output directory, and uses a timestamp to uniquely identify each file.

To enable Subscription Service to receive and format messages, the qualities of service and wire format are set the same as that for Publication Service. A write schema is defined for the service. After the schema is defined, write file records are automatically created, one for each class found by the write schema. The Item file record is modified so that an asterisk (*) is used as padding for the quantity field. The Orders write file record is modified with an ampersand (&) and an underscore (_) for the ID field and the field1 field respectively. The write schema is associated with Subscription Service in the **Schema** tab.

Running the Example

You can use Adapter Launcher to run the example in TIBCO Business Studio.

Procedure

1. Open the sample project in TIBCO Business Studio.
2. From the menu, click **Run > Run Configurations**.
3. In the Run Configurations dialog, click **New launch configuration** .
4. In the Adapter Configuration panel, click **Browse** and select `positionalWriter.adfilesmodel` for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
5. Click **Apply**, and then click **Run**.
6. Repeat [Step 2](#) and [Step 3](#), and in the Adapter Configuration panel, click **Browse** and select `positionalReader.adfilesmodel` for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
7. Click **Apply**, and then click **Run**.

Expected Result

Publication Service parses the `positional.txt` file and publishes the objects. Subscription Service receives the objects, formats and writes them to the `outpositional.txt<timestamp>` file.

You can view the example results in the `TIBCO_HOME\adapter\adfiles\version_num\examples\writer\output` directory. The input file is a positional record. Fields in the output files are formatted with certain characters as padding. The input file and output results are shown as follows:

Input File	Output File
<pre> ID41678 20May2000 GigaWidget 60 \$75 MegaBucket 48 \$125 Hopkins Associates ID26490 ID41680 20May2000 Rt.Clopper 40 \$100 Lt.Clopper 50 \$100 Jersey WebInovators ID46786 </pre>	<pre> The first file: ID41678&&&20May2000_ GigaWidget *****60.000\$75 MegaBucket *****48.000\$125 Hopkins AssociaID26490 The second file: ID41680&&&20May2000_ Rt.Clopper *****40.000\$100 Lt.Clopper *****50.000\$100 Jersey WebInovaID46786 </pre>

Promote Project

The Promote project contains three adapter configurations: PromoteReader.adfilesmodel, PromoteWriter.adfilesmodel and PromoteWriter2.adfilesmodel. The sample project shows how the adapter handles promotion and demotion of records.

In PromoteReader.adfilesmodel, Publication Service reads and parses the `promote.txt` file and publishes the objects. The read schema of Publication Service has three records: Orders, Customer, and Item.


In PromoteWriter.adfilesmodel, Subscription Service receives the objects, formats, and writes them to the `outpromote.txt` file. The write schema of Subscription Service has three corresponding file records: Orders, Item and Customer.

In PromoteWriter2.adfilesmodel, Subscription Service receives the objects, formats, and writes them to the `outpromote2.txt` file. The write schema of Subscription Service is modified so that the Item write file record is at the same level as the Orders writer file record. The Orders record has an additional field, which is the Item record.

Running the Example

You can use Adapter Launcher to run the example in TIBCO Business Studio.

Procedure

1. Open the sample project in TIBCO Business Studio.
2. From the menu, click **Run > Run Configurations**.
3. In the Run Configurations dialog, click **New launch configuration** .
4. In the Adapter Configuration panel, click **Browse** and select PromoteWriter.adfilesmodel or PromoteWriter2.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
5. Click **Apply**, and then click **Run**.
6. Repeat [Step 2](#) and [Step 3](#), and in the Adapter Configuration panel, click **Browse** and select PromoteReader.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
7. Click **Apply**, and then click **Run**.

Expected Result

Publication Service parses the `promote.txt` file and publishes the objects. Subscription Service receives the objects, formats and writes them to the `outpromote.txt` and `outpromote2.txt` files according to their write schemas.

Fields in the input file are separated by commas and Item is a separate line. In PromoteWriter.adfilesmodel, the output files are formatted with semicolons. In PromoteWriter2.adfilesmodel, the Item line is promoted to the level of the Order line.

You can view the example results in the `TIBCO_HOME\adapter\adfiles\version_num\examples\writer\output` directory. The input file and output results are shown as follows:

File Name	File Content
promote.txt	Order, ID41678, 20May2000 Item, GigaWidget, 60, \$75 Item, MegaBucket, 48, \$125 Customer, Hopkins Associates, ID26490 Order, ID41680, 20May2000 Item, Rt.Clopper, 40, \$100 Item, Lt.Clopper, 50, \$100 Customer, Jersey WebInovaters, ID46786
outpromote.txt	Order;ID41678;20May2000 Item;GigaWidget;60;\$75 Item;MegaBucket;48;\$125 Customer;Hopkins Associates;ID26490 Order;ID41680;20May2000 Item;Rt.Clopper;40;\$100 Item;Lt.Clopper;50;\$100 Customer;Jersey WebInovaters;ID46786
outpromote2.txt	Order;ID41678;20May2000;Item;GigaWidget;60;\$75;Item;MegaBucket; 48;\$125 Customer;Hopkins Associates;ID26490 Order;ID41680;20May2000;Item;Rt.Clopper;40;\$100;Item;Lt.Clopper; 50;\$100 Customer;Jersey WebInovaters;ID46786

PublishDelay Project


The PublishDelay project contains the delayedPublisher.adfilemodel adapter configuration. The sample project shows how Publication Service delays sending messages to accommodate subscribers that are slow to process them.

In delayedPublisher.adfilemodel, Publication Service parses the oneMegaByteFile.txt file and publishes each line in the file as a separate message. To enable Publication Service to delay sending messages, the **Document Delay** field (in the **Advanced** tab of Publication Service) is set to 5000, so that a five-second delay is imposed between the sending of each message.

Running the Example

You can use Adapter Launcher to run the example in TIBCO Business Studio.

Procedure

1. Open the sample project in TIBCO Business Studio.
2. From the menu, click **Run > Run Configurations**.
3. In the Run Configurations dialog, click **New launch configuration** .
4. In the Adapter Configuration panel, click **Browse** and select delayedPublisher.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, D:\temp.
5. Click **Apply**, and then click **Run**.

Expected Result

A five-second delay is imposed between the sending of each message.

SFT Project

The SFT project contains two sets of examples: the FTP example and the MessageSelector example. The two examples show how the adapter transfers files in Simple File Transfer (SFT) mode.

FTP Example

The FTP example contains two adapter configurations: FTPReader.adfilemodel and FTPWriter.adfilemodel. The example shows how the adapter transfers files in SFT mode without trigger messages.


In FTPReader.adfilemodel, Publication Service parses the `ftp_ecm.txt` file and publishes the content on the `ftp` subject without any predefined schemas or processing. In the **Configuration** tab of Publication Service, the **Transfer Mode** field is **Simple File Transfer**, the **Life Cycle** field is **Once-Only**. In the **SFT** tab, the **Transmission Buffer Size** field is 32 kilobytes, the **Progress File Name** field is `__FTPReaderPublicationService.prg`, and the **File Transfer Mode** field is **Binary**.

In FTPWriter.adfilemodel, Subscription Service listening on this subject receives the file and writes it to the `outftp.txt` file in the output directory. In Subscription Service, the **Transfer Mode** field is also **Simple File Transfer**. In the **SFT** tab, the **File Transfer Mode** field is **Binary**.

Running the Example

You can use Adapter Launcher to run the example in TIBCO Business Studio.

Procedure

1. Open the sample project in TIBCO Business Studio.
2. From the menu, click **Run > Run Configurations**.
3. In the Run Configurations dialog, click **New launch configuration** .
4. In the Adapter Configuration panel, click **Browse** and select FTPWriter.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
5. Click **Apply**, and then click **Run**.
6. Repeat [Step 2](#) and [Step 3](#), and in the Adapter Configuration panel, click **Browse** and select FTPReader.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
7. Click **Apply**, and then click **Run**.

Expected Result

Publication Service publishes the `ftp_ecm.txt` file. Subscription Service receives it and writes it to the `outftp.txt` file.

You can view the example results in the `TIBCO_HOME\adapter\adfiles\version_num\examples\writer\output` directory. The input and output files are exactly the same because files are transferred without any schema changes. The input file and output results are shown as follows:

Input File	Output File
Item, GigaWidget, 60, 75, 00000	Item, GigaWidget, 60, 75, 00000
Item, MegaBucket, 48, 125, 00001	Item, MegaBucket, 48, 125, 00001
Item, RtClopper, 40, 100, 00002	Item, RtClopper, 40, 100, 00002
Item, LtClopper, 50, 100, 00003	Item, LtClopper, 50, 100, 00003
...	...

MessageSelector Example

The MessageSelector example contains an adapter configuration, a business process and a JMS connection. The example shows how to use JMS trigger messages to transfer files in SFT mode.

The adapter configuration is MessageSelector.adfilesmodel, the business process is TriggerMessageSelector.bwp. After the TriggerMessageSelector.bwp process starts, the JMS topic publisher sends a JMS message to the messagePub topic. Publication Service is triggered and sends the ftp_ecm.txt file to the selected subscribers. The JMS connection is used by the JMS topic publisher.

The MessageSelector.adfilesmodel contains a Publication Service service and two Subscription Service services.

- PublicationService

Publication Service parses the ftp_ecm.txt file. The transfer mode is **Simple File Transfer**, the life cycle is **Repeating**, the polling method is **JMS (topic) Message**, and the polling destination is messagePub. When Publication Service is triggered, it publishes the file content to the selected subscribers.

- SubscriptionService

Subscription Service subscribes to messages and writes them to the outselector.txt file. The transfer mode is **Simple File Transfer**, and the polling method is **JMS (topic) Message**.

The JMS message selector specified on the SubscriptionService endpoint is as follows:

```
(Branch = 'New York' OR Branch = 'ALL') AND ((SalesUpper >= 30
AND SalesUpper <= 70) OR SalesVolume = 'ALL')
```

- SubscriptionService1

Subscription Service subscribes to messages and writes them to the outselector1.txt file. The transfer mode is **Simple File Transfer**, and the polling method is **JMS (topic) Message**.

The JMS message selector specified on the SubscriptionService1 endpoint is as follows:

```
Branch IN ('Boston','ALL') AND ((SalesUpper >= 20 AND SalesUpper
<= 50) OR SalesVolume = 'ALL')
```

Running the Example


You can use Adapter Launcher to run the example in TIBCO Business Studio.



If the ActiveMatrix BusinessWorks 5 project has a Start activity, you have to add a Timer or File Poller activity to initiate the migrated process.

Procedure

1. Open the sample project in TIBCO Business Studio.
2. Create a new process, add a Timer or File Poller activity, and then add a CallProcess activity. Select the TriggerMessageSelector.bwp process for the CallProcess activity.
 - If you migrate the project from the **BusinessWorks5** folder, you have to complete this step.

- If you import the project from the **BusinessWorks6** folder, you can skip this step.
3. From the menu, click **Run > Run Configurations**.
 4. In the Run Configurations dialog, click **New launch configuration** .
 5. In the Adapter Configuration panel, click **Browse** and select MessageSelector.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, D:\temp.
 6. Click **Apply**, and then click **Run**.
 7. In the Run Configurations window, click **BusinessWorks Application > BWApplication**. Select SFT.application, click **Apply**, and then click **Run**.

Expected Result

Publication Service parses the ftp_ecm.txt file and transfers the file using selective routing. Only SubscriptionService1 matches the conditions in the JMS message and receives the message.

After the process starts, the JMS topic publisher sends the JMS message "Branch='ALL',SalesUpper=20" to the messagePub topic. Publication Service is triggered and sends the ftp_ecm.txt file to the selected subscribers. The Branch and SalesUpper properties are included in the outgoing data messages. Only SubscriptionService1 matches the conditions in the JMS message. Publication Service routes the messages to SubscriptionService1.

If you send the JMS message, "Branch='New York',SalesUpper=30", to the messagePub topic, only SubscriptionService can receive the message.

SimpleRecord Project

The SimpleRecord project contains two adapter configurations: delimitedReader.adfilesmodel and delimitedWriter.adfilesmodel. The sample project shows how the adapter transfers records.

- In delimitedReader.adfilesmodel, Publication Service parses the delimited.txt file and sends the content in a series of messages. After processing all lines in the input file, Publication Service exits.

To enable Publication Service to parse the delimited.txt file, the service is configured as follows: the **Life Cycle** field is **Once-only**, so the service reads the file only once, and exits after sending messages. The **Input Directory**, **Recognition Method** and **File Name** fields indicate the directory that contains the input file, how to select the file and the file name. The service uses the Reliable quality of service and publishes the messages using ActiveEnterprise message semantics.

Three delimited file records are defined for parsing each type of line found in the delimited.txt file: Order, Item and Customer. Take the Order file record for example, the **Strip Blanks** and **Repeating Delimiter** check boxes are selected, so blank spaces between fields on the order line are removed, and if any repeated delimiter characters exist between fields, the repeated characters are ignored. The line is identified by a field value. The value is identified by the value of the Constant column. The parent line and child lines to be published are identified in a read schema. The read schema indicates that the Order file record represents the parent line and the Item and Customer file records represent child lines. The read schema is associated with Publication Service in the **Schema** tab.

- In delimitedWriter.adfilesmodel, Subscription Service receives all messages, formats and writes them to an output file located in a working directory. One minute after opening the file, the service moves the file to its output directory and renames the file using a timestamp. Each time you run Publication Service, another output file is created in the output directory and named using a timestamp.

To enable Subscription Service to receive and format messages, the service is configured as follows: the **Wip Creation Mode** field is **Append the messages to file, close on timer**. The quality of service and wire format are configured the same as that for Publication Service. The location of the working and output directories are specified in the **Processing** tab.


A write schema is defined for the service to include the ActiveEnterprise classes created by Publication Service. After the schema is defined, write file records are automatically created, one for each class found by the write schema. The write file records are defined in this way so that the Customer line is written under the Item line.

The Customer write file record is modified so that a tab is used as the delimiter character. The Item and Order write file records are modified with a semicolon as the delimiter character. The write schema is associated with Subscription Service in the **Schema** tab.

Running the Example

You can use Adapter Launcher to run the example in TIBCO Business Studio.

Procedure

1. Open the sample project in TIBCO Business Studio.
2. From the menu, click **Run > Run Configurations**.
3. In the Run Configurations dialog, click **New launch configuration** .
4. In the Adapter Configuration panel, click **Browse** and select delimitedWriter.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, D:\temp.

5. Click **Apply**, and then click **Run**.
6. Repeat [Step 2](#) and [Step 3](#), and in the Adapter Configuration panel, click **Browse** and select delimitedReader.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, D:\temp.
7. Click **Apply**, and then click **Run**.

Expected Result

Publication Service parses the delimited.txt file and sends the content in a series of messages. Subscription Service receives all messages, formats and writes them to the delimited.txt<timestamp> file.

You can view the example results in the `TIBCO_HOME\adapter\adfiles\version_num\examples\writer\output` directory. Records in the input file are separated by commas. Records in the output file are formatted with tabs and semicolons as separators. The input file and output results are shown as follows:

Input File	Output File
Order, ID41678, 20May2000 Item, GigaWidget, 60, \$75 Item, MegaBucket, 48, \$125 Customer, Hopkins Associates, ID26490 Order, ID41680, 20May2000 Item, Rt.Clopper, 40, \$100 Item, Lt.Clopper, 50, \$100 Customer, Jersey WebInovaters, ID46786	Order;ID41678;20May2000 Item;GigaWidget;+6.000000E+001;\$75 Item;MegaBucket;+4.800000E+001;\$125 Customer Hopkins Associates ID26490 Order;ID41680;20May2000 Item;Rt.Clopper;+4.000000E+001;\$100 Item;Lt.Clopper;+5.000000E+001;\$100 Customer Jersey WebInovaters ID46786

International Projects

The **International** folder contains three sample projects: Big5, Japanese, and Korean. The three sample projects show how the adapter works on a cross platform with multiple encoding.

Examples in the three projects apply to files that contain character sets that are not represented in ASCII and ISO-8859-1.



UTF-8 repository encoding is required whenever an adapter handles non ASCII or non ISO-Latin-1 data. For more information about how to change the encoding of your file or server based repository, see *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide*.

The international examples are created to show cross platform support with multiple encodings. The examples contain adapter configuration for the following file content encodings.

Project Name	Language	File Content Encoding	Data Set Description
Big5	Traditional Chinese	Big5	90% of Big5 Character Set
Japanese	Japanese	Shift JIS (CP943)	95% of SJIS Character Set
	Japanese	EUC-JP	Sample EUC_JP Data
	Japanese	Shift JIS (CP943)	Sample SJIS Data
Korean	Korean	EUC-KR	Sample EUC_KR Data

Running the Example


You can use Adapter Launcher to run the example in TIBCO Business Studio.

The following method describes how to run the Big5 project. The procedures also apply to the other two sample projects: the Japanese and Korean projects. When running the other two projects, you can select corresponding adapter configurations in the **Adapter Configuration** field.



On Microsoft Windows, the required environment is set in the properties file of the adapter. On Unix, though most of the environment is set in the properties file of the adapter, you have to set additional environment variables. You can do this by sourcing `adfilesenv.csh` (or `adfilesenv.sh`) before running the adapter.

Procedure

1. Open the sample project in TIBCO Business Studio.
2. From the menu, click **Run > Run Configurations**.
3. In the Run Configurations dialog, click **New launch configuration** .
4. In the Adapter Configuration panel, click **Browse** and select TradChinDelimitedSubscriber.adfilesmodel for the **Adapter Configuration** field. The **Adapter Executable** field is provided automatically. Provide a directory for the **Working Directory** field, for example, `D:\temp`.
5. Click **Apply**, and then click **Run**.
6. Repeat [Step 2](#) and [Step 3](#), and in the Adapter Configuration panel, click **Browse** and select TradChinDelimitedPublisher.adfilesmodel for the **Adapter Configuration** field. The **Adapter**

Executable field is provided automatically. Provide a directory for the **Working Directory** field, for example, D:\temp.

7. Click **Apply**, and then click **Run**.

Expected Result

Publication Service parses the input files and sends the content in a series of messages. Subscription Service receives all messages, formats and writes them to the output files.

Different sample projects use different input files and generate different output files. To view the content of those files, you have to use the corresponding encoding. For details, see [International Projects](#).

You can view the example results in the *TIBCO_HOME\adapter\adfiles\version_num\examples\writer\output* directory. The following table lists the input file and output file names.

Project Name	Input File Name	Output File Name
Big5	TraditionalChinese_Big5.csv	TraditionalChinese_Delimited_Big5_Output.csv
Japanese	Japanese_EUC_JP.txt	Japanese_Delimited_EUC_JP_Output.txt
	Japanese_SJIS.csv	Japanese_Delimited_SJIS_Output.csv
	Japanese_Positional_SJIS.txt	Japanese_Positional_SJIS_Output.txt
Korean	Korean_EUC_KR.txt	Korean_EUC_KR_Output.txt